ExamCollection

Amazon AWS Certified Solutions Architecht - Assoctiate 2018

Study Guide

Table of Contents

Introduction
    AWS Certified Solutions Architect Certification details:
Exam scheduling:
Amazon Study Materials and helpful links:
Regions and availability zones
EC2: Elastic Compute Cloud
    EC2 instance options
    On demand instances:
    Reserved instances:
    Spot instances:
    Instance differences
    Instance types
    HSM Instance Hardware Security Module
    EC2 image login default usernames and how to SSH into them:
EC2 Placement Groups
Launching an EC2 instance
Elastic Container service (ECS)
Docker
Elastic Kubernetes Service (EKS) Kubernetes
Fargate
Lambda
    Serverless Application Model
VPC: Virtual Private Cloud
    VPC Networking
    VPC Enhanced Networking
    VPC Networking Elastic Network Interface (ENI)
    VPC peering
    VPC peering security
Identity Access Manager: IAM
    IAM Roles
    IAM Security Token Services (STS)
Network Address Translation: (NAT)
    VPC NAT Instances

- NAT Gateway
- NAT vs VPC Bastion (jump) server
- VPC Security Network Access Control Lists vs. Security Groups
- Network Access Control Lists (NACL)
- Security Groups
- Internet gateways IGW

Flow logs
Elastic IP addresses: EIP
Route 53

- Domain Name Systems overview
- AWS Route 53 DNS services
- Top Level Domain Name
- Name Servers
- Zone Files
- Hosted Zones
- Domain Registrars
- DNS record types
- A Records Address record
- Alias Records
- CNAME Canonical name
- NS Records Name Server
- SOA records Start of Authority
- TTL record Time to Live
- Route 53 Routing Policies
- Simple Routing Policy
- Weighted Routing Policy
- Latency Based Routing
- Failover Based Routing
- Geolocation based routing
- DNS Health Checks
- Virtual Private Networks VPN
- Hardware AWS VPN
- Direct connect
- Cloud hub VPN
- Software VPN
- Auto scaling groups
- Auto Scaling group Launch Configurations

CloudFront Content delivery network
Elastic Load Balancers: ELB
    Custom VPCs and Elastic Load Balancers
Elastic Beanstalk
Lambda
Lightsail
S3: Simple Storage Services
    S3 Storage tiers:
    S3 Standard:
    S3 standard Infrequent Access (IA):
    S3 standard Infrequent Access Single Availability Zone (IA):
    S3 Reduced Redundancy Storage (RRS):
    S3 version control
    S3 cross region replication
    S3 Lifecycle Management
    S3 security and encryption
    S3 versioning
    S3 Billing
    S3 Transfer Acceleration / Multipart Upload
Glacier
EBS: Elastic Block Store
    EBS Consists of the following offerings:
    Create EBS volumes in the AWS console.
    EBS Snapshots
    SSD GP2
    Provisioned IOPS SSD
    Magnetic (Standard)
EFS: Elastic File Service
AWS Storage Gateway: ASG
Database basics
    Relational databases
    Non-relational databases
    Data warehousing
RDS: Relational data base
RDS Back-ups, Multi-AZ's and Read replicas
    Backups
    Read Replicas

- Snapshots
- Database Encryption
- Database multi-AZ
- RDS option groups

DynamoDB
Neptune
Database Migration Services: DMS
Aurora
Data Migration services
Snowball
- Snowball appliance
- Snowball Edge appliance
- Snowmobile

Server Migration Services: SMS
Analytics
Athena
Redshift
Elastic Map Reduce: EMR
SageMaker
Cloud Search / Elastic Service
Data Pipeline
QuickSight
Security and Identity
- AWS Security and compliance

Identity Access Management: IAM
- Identity Access Management Roles

Inspector
Certificate Manager
Key Management services: KMS
Directory Service
Edge Services
ElastiCache
Web Application Firewall: WAF
AWS Shield
Artifact
AWS Management tools
Cloud Formation

CloudTrail
Opsworks
AWS Config: Config Manager
Trusted Advisor
Step functions
Simple Workflow Service: SWF
    SWF Actors: Workflow starters
    SWF Actors: Workflow deciders
    SWF Actors: Activity Workers
API Gateway
Kinesis
    Kinesis Streams
    Kinesis Firehose
    Kinesis Analytics
Developer tools
    CodeCommit
    CodeBuild
    CodeDeploy
    CodePipeline
Mobile Services
    Mobile Hub
    Cognito
    Device Farm
    Mobile Analytics
    PinPoint
Messaging
    SNS Simple Notification Services
SQS Simple Queueing Services
    Standard SQS queues
    FIFO SQS queues
    Dead letter queues
    SNS/SQS Differences
Business Productivity
    WorkDocs
    Workmail
    SES Simple E-mail services
Internet of Things

Desktop and App Streaming
- Workspaces
- Appstream

Artificial Intelligence
- Alexa
- Polly
- Elastic Transcoder

Machine Learning
- Rekognition

Console services

Import/Export
- VM Import/export

Snowball
- Storage Import/Export Snowball

Storage Gateway
- Volume Gateway (Gateway Stored Volumes)
- Volume gateways
- Volume gateways (Stored Volumes)
- Volume Gateway (Cached Volumes)
- Gateway Virtual Tape libraries (VTL)
- File Gateway

Security groups

Creating an Amazon Machine Image: AMI

AMI types (EBS vs Instance store)

SDK Software development kits

CloudWatch
- Services Used with CloudWatch
- Cloudwatch Custom Metrics
- CloudWatch Logs
- CloudWatch Alarms
- Cloudwatch Integration with IAM
- Cloudwatch Limitations

AWS Command line

AWS Support offerings

Well Architected Framework
- Well architected framework Security
- Well architected framework Reliability

Well architected framework Performance Efficiency
Well architected framework Performance Cost Optimization
Appendix 2 AWS links: updates, pdf's all AWS internals
AWS White papers for AWS-CSA Exam Prep:
AWS Blogs and presentations:
Scripts:
EC2 Instance Metadata
Putty access to an EC2 instance
Make a filesystem on an ECB volume:
Install a web server on EC2:
PuTTY tutorial
Browser troubleshooting utilities
HTML5 sample web page downloads
Bash Scripting
Windows Bash Scripting
Installing the apache webserver
Drawing and documentation applications for AWS

Introduction

AWS Certified Solutions Architect Certification details:

- Multiple-choice and multiple-answer questions including scenario based questions.
- The exam time limit is 130 minutes
- 65 questions
- Available in English, Japanese, Korean, and Chinese
- Exam registration fee is $150.00 U.S. Dollars
- Passing grade is on a curve and not a hard percentage
- You will be notified at the completion of the exam if you passed or not
- AWS will send you an email in several days with notification of your certification being approved and posted in your account

- Exam Objective weighting:

Domain	% of Examination
1.0 Designing highly available, cost-efficient, fault-tolerant, scalable systems	60%
2.0 Implementation/Deployment	10%
3.0 Data Security	20%
4.0 Troubleshooting	10%
TOTAL	**100%**

- Main site for AWS certifications: https://aws.amazon.com/certification/
- AWS Certified Solutions Architect home page: https://aws.amazon.com/certification/certified-solutions-architect-associate/
- AWS Solutions Architect Associate exam blueprint: https://d0.awsstatic.com/training-and-certification/docs-sa-assoc/AWS_certified_solutions_architect_associate_blueprint.pdf

Exam scheduling:

- Create an account on the AWS training and certification portal. This site allows you to schedule your exam and track your certifications. You can also download digital badges and transcripts from this site. They also have a store where you can order shirts and other AWS certified products from.
- Go to: https://aws.amazon.com/certification/ to begin the exam registration process
- https://aws.psiexams.com is the actual site where you schedule the exam
- Give exam proctor the six character authorization code given when booked online
- Exam reschedule within 72 hours incurs a penalty, must contact awscertification@amazon.com
- If a retest is needed you must wait 2 weeks before taking the exam again

Amazon Study Materials and helpful links:

- Follow re:invent conference videos 400 series most useful; 200 series is basic
- http://www.reinventvideos.com
- Slideshare on AWS has thousands of slides: www.slideshare.net/AmazonWebServices
- Architecture center: https://aws.amazon.com/architecture
- Answer center: https://aws.amazon.com/answers
- Case studies: https://aws.amazon.com/case-studies
- All changes and updates to services: https://aws.amazon.com/releasenotes
- What's new in AWS: https://aws.amazon.com/new
- Blogs: https://aws.amazon.com/blogs/aws/
- AWS podcast: https://aws.amazon.com/podcasts/podcast
- AWS forums: https://forums.aws.amazon.com
- ACloud.guru weekly updates: https://acloud.guru/aws-this-week
- Main CSAA certification site: https://aws.amazon.com/certification/certified-solutions-architect-associate/
- AWS YouTube channel: https://www.youtube.com/user/AmazonWebServices
- Git templates: https://github.com/awslabs A massive and critical repository for AWS resources and code
- AWS has a YouTube site that is full of technical presentations: https://www.youtube.com/user/AmazonWebServices/videos
- Reddit has a great site with lots of resources: https://reddit.com/r/amazonwebservices has all of the latest developments on AWS offerings

Regions and availability zones

- An AWS Region is a completely independent entity in a geographical area. There are two more Availability Zones in an AWS Region
- Within a region, Availability Zones are connected through low-latency links
- Any number of components of a workload can be moved into AWS, but it is the customer's responsibility to ensure that the entire workload remains compliant with various certifications and third-party attestations
- Each availability zone consists of multiple discrete data centers with redundant power and networking/connectivity
- Since each AWS Region is isolated from other regions, it provides for high fault tolerance and stability
- For launching an EC2 instance, we have to select an AMI within the same region
- Region code lookup tool: http://docs.aws.amazon.com/general/latest/gr/rande.html
- A Region is a geographical area with two or more availability zones
- An availability zone is simply one or more data centers in a region (A AZ can be more than one data center)
- 2018 there are 18 regions and 55 availability zones worldwide
- 2 or more AZ per region and each availability zone is 2 or more physical datacenters
- Edge location Content delivery network endpoint (Cloudfront) a cache of media in the cloud
- 2018 there are 125 edge locations, 11 regional edge caches in 62 cities across 29 countries. AWS is constantly adding to cloud front edge locations

EC2: Elastic Compute Cloud

- Virtual machines in the cloud
- Four network capacity ratings: Low, moderate, high and 10 Gbps
-
- Instance metadata is information about the EC2 instance that can be defined such as instance ID, instance type, security groups – this data can be obtained by a HTTP call inside the instance
- The default the maximum Amazon EC2 instance limit for all regions is 20 but can be increased by request

- Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduce the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change
- Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use
- EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios
- A very good resource for EC2 information is: http://www.ec2instances.info/
- Default soft limit is 20 EC2 instances per REGION. AWS can increase this if you submit a request varies on the instance type, 20 reserved per AZ
- When you launch and instance it goes into pending state and then moves to running
- When you stop an instance, it can only be done if you are using EBS storage
- If you're using ephemeral storage, you cannot stop it, it can only be terminated which causes it to move from running to shutting-down to terminated at which point it goes away for good and all data is lost in ephemeral storage which is local storage on the server the AMI is running on
- A terminated instance remains visible in the console for a while

before it is deleted. You cannot recover an terminated instance

- A stopped instance does not incur any charges but it does charge for storage in the EBS volumes of stopped instances
- You can modify certain attributes of stopped instances including the instance type
- Starting a stopped instance puts it back into the pending state which moves the instance to a new host machine in the defined regions availability zone and VPC
- When you start and stop an instance, you lose any data on the instance store volumes (Ephemeral) on the previous host computer
- Instances are almost always deployed inside of a VPC
- An instance can be deployed in different availability zones inside of a region
- EC2 instances can use elastic block store or EBS for block storage volumes in each AZ and the EBS volumes can be saved using snapshots
- EC2 uses PKI for security and a public private key pair to encrypt and decrypt the login information, you must have the private key to SSH into the instance which is holding the public key
- Windows uses a key pair and then also a username password to log in using the remote desktop protocol (RDP)

EC2 instance options

- AWS offers several options on reserving and purchases instances and are explained below

On demand instances:

- Fixed rate charges by the hour with no commitment
- Users want the low cost flexibility of Amazon EC2 without any up-front payment or long term commitment
- Applications with short term, spikey, or unpredictable workload that cannot be interrupted.
- Applications being developed and tested on Amazon EC2 for the first time. Test and development environments (use and delete when done)
- Supplement reserved instances, black Friday load increase for example

Reserved instances:

- Reserve for 1 – 3 years with a capacity reservation
- Big discount from the hourly on-demand service
- Applications with steady state or predictable usage
- Applications that require reserved capacity
- Users able to make upfront payments to reduce their total computing costs further
- Cheaper the more you pay up front and the longer the term
- You can change the instance type only within the same instance type family
- You can change the availability zone of a reserved instance
- You cannot move a reserved instance to another region
- You cannot change the operating system nor the instance type family (specific to instance type)
- Limit of 20 reserved instances per region
- Reserved Instances provide you with a significant discount (up to 75%) compared to on-Demand instance pricing
- You have the flexibility to change families, OS types, and tenancies while benefiting from Reserved Instance pricing when you use Convertible Reserved Instances

Spot instances:

- Enables you to bid whatever price you want for instance capacity, lowest cost offering but not guarantee of start and stop times
- If you are outbid and Amazon give two minute notice when you are outbid and they are shutting down your spot instance. Use for High performance, Hadoop etc.
- Look at ec2price.com for pricing
- Applications that have flexible start and stop times and can be interrupted by AWS
- Applications that are only feasible at very low compute prices
- Users with urgent computing needs for large amounts of additional capacity
- In Amazon EC2, you bid for a computing instance. Any instance procured by bidding is a Spot Instance
- Multiple users bid for an EC2 Instance
- A spot instance request includes the bid price and instance type which includes the AMI, instance type and the total number of instances you are requesting
- Once the bid price exceeds the Spot price, the user with the highest bid can launch the instance
- As long as the bid price remains higher than the spot price, the instance is yours to use
- Spot price varies with the supply and demand
- You are actually charged at the spot price rate, not your bid price, however the spot price must be below the bid price
- Once spot price exceeds bid price, the instance will be taken back from the user
- If AWS terminates your spot instance, you will not be charged for the final hour when the instance was terminated
- notification of spot termination / scenarios under which AWS might execute a forced shutdown:
    - AWS sends a notification of termination and you receive it 120 seconds before the intended forced shutdown
    - AWS sends a notification of termination but you do not

receive it within the 120 seconds and the instance is shutdown
    - AWS sends a notification of termination and you receive it 120 seconds before the forced shutdown, but the normal lease expired before the forced shutdown
    - AWS sends a notification of termination and you receive it 120 seconds before the intended forced shutdown, but AWS do not action the shutdown
- When bidding on spot instances, it is a good idea to bid in multiple AZ's as pricing is based on AZs, this allows for you to get the best pricing
- If AWS terminates the spot instance, you are not charged for the partial hour
- If you terminate the spot instance you are charge for the complete hour at the current rate
- You are never charged more than your maximum bid price
- If the spot price exceeds your bid price, you are given a two minute notice that it will be terminated
- http://169.254.169.254/latest/meta-data/spot/termination-time

Instance differences

- Spot Instance and On-demand Instance are very similar in nature. The main difference between these is of commitment
- In Spot Instance there is no commitment. As soon as the Bid price exceeds Spot price, a user gets the Instance
- In an On-demand Instance, a user has to pay the On-demand rate specified by Amazon. Once they have bought the Instance they have to use it by paying that rate
- In Spot Instance, once the Spot price exceeds the Bid price, Amazon will shut the instance. The benefit to user is that they will not be charged for the partial hour in which Instance was taken back from them

Instance types

EC2 families such as

- T2: Cheap web servers small database
- M3, M4: General purpose application servers
- C3, C4: CPU intensive Apps/DB's
- R3: memory intensive apps/DB's
- G2: Graphics intensive video encoding/machine learning/3D apps, application streaming
- I2: High speed storage, NoSQL, DB's, Data Warehousing
- D2: Dense storage fileservers/data warehouse/Hadoop

Family	Speciality	Use case
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop
R4	Memory Optimized	Memory Intensive Apps/DBs
M4	General Purpose	Application Servers
C4	Compute Optimized	CPU Intensive Apps/DBs
G2	Graphics Intensive	Video Encoding/ 3D Application Streaming
I2	High Speed Storage	NoSQL DBs, Data Warehousing etc
F1	Field Programmable Gate Array	Hardware acceleration for your code.
T2	Lowest Cost, General Purpose	Web Servers/Small DBs
P2	Graphics/General Purpose GPU	Machine Learning, Bit Coin Mining etc
X1	Memory Optimized	SAP HANA/Apache Spark etc

HSM Instance Hardware Security Module

- Instance with a hardware encryption card installed
- AWS manages the hardware but does not touch the keys
- Dedicated hardware to manage encryption keys
- If there is a too many failed passwords, the HSM deletes and zeros out all keys and data
- To increase performance place the HSM as close to your EC2 instances as possible
- D- for Density
- R- for RAM
- M- Main choice for general purpose apps
- C- for Compute
- G- for Graphics
- I- for IOPS
- F- Is for FPGA
- T- Cheap general purpose (Think T2 Micro)
- P- Graphics (think Pics)
- X- Extreme memory

EC2 image login default usernames and how to SSH into them:

- For an Amazon Linux AMI, the user name is ec2-user
- For a RHEL AMI, the user name is ec2-user or root
- For an Ubuntu AMI, the user name is ubuntu or root
- For a Centos AMI, the user name is centos
- For a Fedora AMI, the user name is ec2-user
- For SUSE, the user name is ec2-user or root
- Otherwise, if ec2-user and root don't work, check with the AMI provider
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html?icmpid=docs_ec2_console

EC2 Placement Groups

- A placement group is a logical grouping of instances within a single Availability Zone
- AWS provides an option of creating a Placement Group in EC2 to logically group the instances within as single Availability Zone
- We get the benefits of low network latency and high network throughput by using a Placement Group
- Placement Group is a free option as of now
- Using placement groups enables applications to participate in a low-latency, 10Gbps network
- Placement groups are recommended for applications that benefit from low latency, high network throughput, or both
- Used in Hadoop, Casandra, grid computing
- Cannot span multiple availability zones, single point of failure
- The name that you specify must be unique in your account
- Only certain types of EC2 instances can be launched in a placement group (Compute optimized, GPU, Memory optimized, Storage Optimized)
- AWS recommends homogenous instances within a placement group (same size and same family)
- Cannot merge placement groups
- Cannot move instances into placement groups. (Snapshot and copy to group works)
- When we stop an instance, it will run in same Placement Group in restart at a later point of time
- Because of the low latency required for a cluster placement group, each cluster placement group can only exist within 1 availability zone
- The biggest limitation of Placement Group is that we cannot add Instances from multiple availability zones to one Placement Group
- Placement groups cannot be deployed across multiple availability zones
- A cluster placement group is a logical grouping of instances within a single Availability Zone.

- Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both.
- To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking.
- You can move an existing instance to a placement group, move an instance from one placement group to another, or remove an instance from a placement group
- Before you begin, the instance must be in the stopped state

Launching an EC2 instance

- Console and then EC2 dashboard
- Resource section shows what is running
- Shows region and availability zone status
- Create instance
- Choose Amazon Machine Image (AMI)
- Select free tier section
- Best to use Amazon AMI image comes pre-baked with Python and many features like DB's
- Choose instance type (see above) t2 micro is free
- Use on –demand instances
- Create a VPC (important for exam)
- One subnet per AZ
- Auto assign an IP address
- IAM role (create ahead of time in IAM)
- Shutdown, stop or terminate leave as stopped (do not leave running as you will be charged)
- Termination leave on
- Tenancy shared for now
- Advanced (a script that runs on startup)
  - add #!/bin/bash
  - Yum update –y
- Add storage root and mount point, leave as default
- Know that delete on termination is checked, it will delete the storage volume
- Cannot enable encryption on the root volume and you can encrypt additional volumes
- The OS volume cannot be encrypted
- Tag instance is keys/tags
- Security groups big on exam, a virtual firewall create a SG to see what it is all about, endpoint level security ACLs
- Use HTTP SSH RDP in security group rule as a minimum
- Review and Launch gives all the config details
- Bottom right is the launch button

- Create a new key pair (or use existing key pair)
- Give it a name such as Ec2Key.pem
- Download key pair
- Save it off from download directory, this is my private key (DO NOT LOSE IT)
- Launch the instance (takes seconds up to minutes)
- Go to view to see the status of the EC2 instance in the AWS console
- Get the DNS link for PuTTY
    - Ssh ec2-user@x.x.x.x –I Ec2Key.pem
    - Chmod 600 Ec2Key.pem
    - Sudo su (on the instance)
    - Yum update –y
- Look at the console pull down menus, connect and actions
- If you delete the root instance the EBS volume also gets deleted
- Default ec2 username is ec2-user and no password for Linux AMIs

Elastic Container service (ECS)

- Amazon EC2 managed container services (Amazon ECS) for Docker
- ECS is Amazons managed version of Docker
- Amazon ECS eliminates the need for you to operate your own cluster management and configuration management systems, or to worry about scaling your management infrastructure
- Good training video: https://awsdevops.io/p/hitchhikers-video-guide-aws-docker/
- Amazon EC2 Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances
- Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure
- With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your cluster, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes, and IAM roles. You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements
- You can integrate your own scheduler or third-party schedulers to meet business or application specific requirements
- ACS allows you to launch and stop container based applications with simple API calls, allows you to get the state of your cluster from a centralized service, and gives you access to many familiar Amazon EC2 features
- ECS is a regional service that you can use on one or more AZs across a new, or existing, VPC to schedule the placement of containers across your cluster based on your resource needs, isolation policies, and availability requirements
- ECS can also be used to create a consistent deportment and build experience, manage and scale ETL (Extract Transform and Load, which are three different types of database functions, pulls data

from one database and configures it into a form that can be loaded into a second database) workloads, and build sophisticated application architectures on a micro-services model

- Containers are a method of operating system virtualization that allow you to run an application and its dependencies in a resource-isolated process
- Containers have everything the software needs to run – including libraries, system tools, code, runtime
- Containers are created from a read-only template called an image
- A Docker image is a read-only template with instructions for creating a Docker container, It contains:
    - An ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime
    - An Image is created from a DockerFile, a plaintext file that specifies the components that are to be included in the container
    - Images are stored in a registry, such as DockerHub of AWS ECR
    - Similar in function to Cloud Formation
    - Images are based off of other images generally and modified as needed
- ECR is the Amazon Container Registry, it is a managed AWS Docker registry service that is secure, scalable, and reliable
    - Like DockerHum
    - Amazon ECR supports private Docker repositories with research-based permissions using AWS IAM so that specific users or Amazon EC2 instances can access the repositories and images
    - Developers can use the Docker CLI to push, pull, and manage images
- ECS Task definition:
    - A task definition is required to run Docker containers in Amazon ECS
    - Task definitions are text files in JSON format that describe one or more containers that form your application

- Task Definitions include:
  - Which Docker images to use with the containers in the task
  - Who much CPU and memory to allocate to each container
  - If the containers are linked together in a task
  - The Docker networking mode to use for the containers in the task
  - What (if any) port in the container are mapped to the host container instance
  - If the task should continue to run in the container finishes or fails
  - Any commands the container should run at startup
  - What (if any) environmental variables should be passed to the container on startup
  - Any data volumes that should be used with the containers in the task
  - What (if any) IAM role your task should use for permissions
- Parameters you can specify in a task definition include:
  - Which Docker images to use with the containers for your task
  - How much CPU and memory to use with each container
  - Whether containers are linked together in a task
  - Networking node to use for the containers in your task
  - Which (if any) ports from the container are mapped to the host container instance
  - Whether the task should continue to run if the container finishes or fails
  - The command the container should run when started
  - Which (if any) environment variables should be passed to the container when it starts
  - Any data volumes that should be used with the containers in the task
  - What (if any) IAM role your tasks should use for permissions
- ECS can authenticate to private registries, such as DockerHub,

using basic authentication

- When you enable private registry authentication, you can use private Docker images in your task definitions
- Integrated with CloudWatch monitoring service
- Docker provides diagnostics tools for troubleshooting
- ECS is used in continuous integration, continuous deployment, microservices
- Dynamic port mapping is the one to one mapping between ECS and ELBs
- Port mappings are used to send traffic on the host container, to send and receive traffic and are specified as part of the container definition
- The Amazon ECS service allows you to run and maintain a specified number (or, the “desired count”) of instances of a task definition simultaneously in an ECS container
- Services are like Auto-Scaling groups for ECS
- If a task should fail or stop, the Amazon ECS service scheduler launches another instance of your task definition to replace it and maintain the desired count of tasks in the service
- Each ECS cluster is a logical grouping of container instances that you can place tasks on. When you first use the Amazon ECS service, a default cluster is created for you
- You can create multiple clusters in an account to keep your resources separate
- ECS Cluster concepts:
    - Clusters can contain multiple different container instance types
    - Clusters are region specific
    - Container instances can only be part of one cluster at a time
    - You can create IAM policies for your clusters to allow or restrict users access to specific clusters
- ECS Scheduling:
    - Ensures that the specified number of tasks are constantly running and reschedules tasks when a task fails (if the underlying container instance fails for some reason)
    - Can ensure tasks are registered against an Elastic load

Balancer ELB

- Customer Scheduler:
  - You can create your own schedulers that meet your business needs
  - Use third party schedulers, such as Blox
  - ECS schedulers leverage the same cluster state information provided by the Amazon API to make appropriate placement decisions
- ECS Container agent allows container instances to connect to your cluster. The Amazon ECS container agent is included in the ECS-optimized AMI, but you can also install it on any EC2 instance that supports the Amazon ECS specification. The ECS container agent is only supported on EC2 instances
  - Pre-installed on ECS AMIs
  - Linux based with Amazon Linux, Ubuntu Red Hat, Centos etc.
  - Does NOT work with Windows
- ECS IAM roles
  - EC2 instances use an IAM role to access ECS
  - ECS tasks use an IAM role to access services and resources
  - Many to one relationship
  - Roles use temporary credentials, short term only max 36 hours
- Security groups attach at the instance-level (i.e. the host, not the task or container)
- You can access and configure the OS of the EC2 instance in your ECS cluster
- ECS soft limits:
  - Clusters per region default = 1000
  - Instances per cluster default = 1000
  - Services per cluster default = 500
- Hard limits:
  - One load balancer per service
  - 1000 tasks per service (the “desired count”)
  - Max. 10 containers per task definition
  - Max. 10 Tasks per instance (host)

- No additional AWS charge for Amazon EC2 Container Service
- You pay for AWS resources (e.g. EC2 instances or EBS volumes) you create to store and run your application
- https://aws.amazon.com/ecs

Docker

- Docker is a software platform that allows you to build, test, and deploy applications quickly
- Docker is highly reliable: you can quickly deploy and scale applications into any environment and know your code will run
- Dicker is infinitely scalable: Running Docker on AWS is a great way to run distributed applications at any scale
- Docker packages software into standardized units called containers
    - Containers allow you to easily package an applications code, configurations, and dependencies into easy to use building blocks that deliver environmental consistency, operational efficiency, developer productivity, and version control
    - Think of shipping containers except Docker ships software in containers
    - Containers contain application and its dependencies, the bare minimum and does NOT include the operating system
    - Containers include the bare minimum to run the application
- Virtualized systems (traditional VM's) have wasted space since every app basically runs on a virtualized guest operating system that compromises for density
- This requirement for so many Linux and Windows VM's has a lot of resource overhead and reduces the density of applications that can run on a server
- Docker archives much higher density and greater portability by removing the per container guest operating system requirement
- Docker containers start much faster than a virtual machine
- Escape from version dependency issues
- Isolation – performance and stability issues with App A in Container A, wont impact App B in Container B
- Docker makes your code extremely portable
- Docker enables micro-services

- Docker image is like a ISO or AMI but does not contain the operating system
- Docker defines everything that is needed for an application to run
- Each container is created from a Docker image
- A Docker container can be run, started, stopped, moved and deleted
- Each container is a secure and isolated application platform
- Layers/Union File System, changes are made on layers and pushed out so the whole image does not need to be recreated like you would with a virtual machine
- Docker file images built off of based image using steps called instructions, each instruction creates a new layer in the Docker image, stored in the DockerFile are objects such as: add a command, create a directory, get environment variable
- Docker Daemon/Engine: Runs on Linux to create the environment to build ship and run containers
- Docker Client: interface between you and the Docker engine allows the creation, manipulation and deletion of Docker containers and control of the Docker Daemon
- Docker Registries/Docker Hub: public or private stores that hold the images for upload and download/ Hub is a huge collection of images to use and is open to the public
- ECS Amazons managed EC2 container service. Allows you to manage Docker containers on a cluster of EC2 instances
- Containers are a method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes
- Clusters are created from a read-only template called an image
- An image is a read-only template with instructions for creating a Docker container
- Images are stored in a registry, such as DockerHub or AWS ECR
- Amazon EC2 container registry is a managed AWS Docker registry service
- A tasks definition is required to run Docker containers in Amazon ECS
- Task definitions are text files in JSON format that describe one or more containers that form your application
- A task definition is like a cloud formation template but for Docker

that configures resources such as the amount of CPU, RAM etc.

- The Amazon ECS service allows you to run and maintain a specified number (or, the “desired count”) of instances of a task definition simultaneously in a ECS cluster
- Think of services like Auto-Scaling groups for ECS
- A ECS cluster is a logical grouping of container instances that you can place tasks on
- Clusters can contain multiple different container instance types
- Clusters are region specific
- Container instances can only be part of one cluster at a time
- You can create IAM policies for you clusters to allow or restrict users access to specific clusters
- You can schedule ECS in two ways, a service scheduler or a customer scheduler
- ECS agent to connect EC2 instances to your ECS cluster, LINUX ONLY
- Use IAM to restrict ECS access
- Security groups operate at the instances level, not the task or container level
- Acloud.guru course on application load balancers has a live lab showing how to configure containers

Elastic Kubernetes Service (EKS)
Kubernetes

- Introduced re:Invent 2017
- Not in AWS Architect exam but important to know
- ECS is Amazons managed version of Kubernetes
- Kubernetes containers
- Across multiple AZ's
- Hybrid cloud compatible
- High availability
- Automated upgrades and patches
- Integrated into AWS CloudTrail, CloudWatch, ELB, IAM, VPC and private link and more to be added

Fargate

- Introduced re:Invent 2017
- Not in AWS Architect exam
- Run containers without managing servers or clusters
- On ECS today
- In EKS 2018
- No clusters to manage
- AWS manages the underlying infrastructure
- Highly scalable, no servers, no clusters no provisioning, sets up all surrounding infrastructure
- Runs containers at the task level instead of the serer level

Lambda

- Lambda is a compute service that runs your code in response to events and AWS automatically manages the underlying compute resources for you
- Serverless computing
- An AWS managed service
- Know this for the exam
- AWS handles the server automation and you supply the code
- Lambda is an abstraction layer, stateless computing
- Lambda runs your code on high-availability compute infrastructure and performs all the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging
- All you need to do is supply the code
- Lambda is a service to run code basically in the PaaS service model
- Lambda functions can run between 100 milliseconds and five minutes in duration
- Many different resource allocations are available to pick from to size the compute requirements for your workload
- Events that trigger Lambda, you can use Lambda to respond to table updates in DynamoDB, modifications to objects in S3 buckets, messages arriving in Kinesis stream, AWS API call logs created by CloudTrail, and custom events from mobile operations, web applications, or other web services
- Starts Lambda code within milliseconds of an trigger event
- You do not have to worry about high availability, scaling, deployment, or management
- Supported programming language is Javascript
- Availability is 99.99%
- First 1 million requests are free and $0.20 per 1 million requests there after
- Duration is calculated from the time your code begins executing until it returns or terminates

- Rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. You are charged $0.00001667 for every GB-second used
- https://aws.amazon.com/lambda/pricing/
- https://aws.amazon.com/lambda/faqs/
- AWS creates applications based on AWS Lambda
- Lambda applications are composed of functions that are triggered by an event
- Lambda functions are executed by AWS in their cloud. You do not have to specify or buy any instances or server for running these functions
- An application created on AWS Lambda is called a serverless application in AWS
- AWS Lambda is a service from Amazon to run a specific piece of code in Amazon cloud, without provisioning any server. So there is no effort involved in administration of servers
- In AWS Lambda, we are not charged until our code starts running. Therefore, it is very cost effective solution to run code
- AWS Lambda can automatically scale our application when the number of requests to run the code increases. So we do not have to worry about scalability of application to use AWS Lambda.
- Some of the main use cases in which AWS Lambda can be used are as follows:
- Web Application: We can integrate AWS Lambda with other AWS Services to create a web application that can scale up or down with zero administrative effort for server management, backup or scalability
- Internet of Things (IoT) applications, can use AWS Lambda to execute a piece of code on the basis of an event that is triggered by a device
- Mobile Backend: create Backend applications for Mobile apps by using AWS Lambda.
- Real-time stream Processing can use AWS Lambda with Amazon Kinesis for processing real-time streaming data
- ETL: use Lambda for Extract, Transform, and Load (ETL) operations in data warehousing applications. AWS Lambda can execute the code that can validate data, filter information, sort data

or transform data from one form to another form

- Real-time File processing: AWS Lambda can also be used for handling any updates to a file in Amazon S3. When we upload a file to S3, AWS Lambda can create thumbnails, index files, new formats etc. in real-time
- In AWS Lambda we can run a function in synchronous or asynchronous mode
    - In synchronous mode, if AWS Lambda function fails, then it will just give an exception to the calling application
    - In asynchronous mode, if AWS Lambda function fails then it will retry the same function at least 3 times
- If AWS Lambda is running in response to an event in Amazon DynamoDB or Amazon Kinesis, then the event will be retried till the Lambda function succeeds or the data expires. In DynamoDB or Kinesis, AWS maintains data for at least 24 hours
- Default ephemeral disk capacity "/temp/space" is 512MB

Serverless Application Model

- Use AWS Serverless Application Model (AWS SAM) to deploy and run a serverless application
- AWS SAM is not a server or software. It's just a specification that has to be followed for creating a serverless application
- Once serverless application is created, use CodePipeline to release and deploy it
- CodePipeline is built on Continuous Integration Continuous Deployment (CI/ CD) concept

VPC: Virtual Private Cloud

- A virtual or logical cloud data center in the public cloud where all of your assets and services are deployed
- VPC's are absolutely critical to know and understand to pass the CSAA exam
- A logically isolated section of the AWS cloud
- Launch AWS resources in a virtual network that is exclusive to you
- VPC configuration is in the networking section of the AWS console
- An Amazon VPC is associated with exactly one region that is specified when the VPC is created
- You have complete control over the virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways
- The network configuration is completely customizable for your Amazon Virtual Private Cloud.
- You can create public-facing subnet for your webservers that has access to the internet, and place your backend systems such as databases or application servers in a private-facing subnet with no internet access
- VPCs support multiple layers of security, including security groups and network access control lists, to control access to Amazon EC2 instances in each subnet
- A great re:Invent presentation on VPC: https://youtu.be/St3SE4LWhKo
- 5 VPCs per region (default), more can be requested
- 5 internet gateways per region (this is equal to the VPC limit because you can only have one internet gateway attached to a VPC at a time)

- 50 customer gateways per region, request more if needed
- 50 VPN connections per region
- 200 route tables per region / 50 entries per route table

- 5 elastic IP addresses per VPC
- 500 security groups per VPC
- 50 rules per security group
- 10 VPN connections per VPC
- 200 subnets per VPC (more upon request)
- 5 security groups per network interface (security groups although generally referred to as being on the instance level are technically on the VPC level and not EC2 even though they are applied to EC2 instances)
- VPCs can span availability zones
- VPCs cannot span across regions
- VPCs can be connected together
- VPCs are a big part of all three associate level exams
- Logical datacenter in the AWS public cloud
- Minimum IPv4subnet size in a VPC is a /28 16-network subnet that supports 14 hosts each (16 block)
- Maximum IPv4 subnet IP address range in a VPC is a /16 subnet
- The main VPC route table is created by default when the VPC is created
- All VPC subnets can communicate with each other by default (when the subnets are provisioned, routes are automatically added to the routing table)
- Peer VPCs in the same account or a different accounts is allowed in the same region
- No transitive peers, all peering must be direct between VPCs
- Peering is a networking connection between two Amazon VPCs that enables instances in either Amazon VPC to communicate with each other as if they were within the same network
- Peering is available only between Amazon VPCs in the same region
- Access via the internet through a router and then NACL's into the different subnets via security groups in each subnets to the instances
- Subnets cannot span availability zones, each subnet is exclusive to an availability zone
- Launch instances into a public or private subnet of your choosing
- Assign custom IP address ranges in each subnet

- Configure route tables between subnets
- Create an internet gateway and attach it to a VPC (only one gateway per VPC)
- VPCs offer very good security over your cloud resources
- Instance security groups are stateful. If HTTP is allowed into the VPC by default it is allowed out too
- Subnet network access control lists (ACLs) stateless, must create a rule in and out (matching in/out ACL rules)
- Default vs custom VPCs:
- The default VPC creates a private and public IP address for all EC2 instances at launch
- Default VPC is created when you set up your account
- All subnets in the default VPC have a route out to the internet by default
- Each region has a single default VPC for your account (never delete it)
- Default VPC CIDR block is 172.31.0.0/16 (RFC 1918 private address space)
- If you create a resource such as a EC2, EBS, S3 etc. and don't specify a VPC, it will be placed in the default VPC
- Best practice is to use non-default VPCs at all times and use the default only for testing
- Default VPC includes a default subnet, Internet Gateway, main route table, default security groups and a default network ACL
- Each EC2 instance can have both a public and private IP address when created
- 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 to another via a direct route using private IP addresses. Use a separate VPC for separate functions and interconnect them
- Peered VPCs behave as if they are 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 done in a star configuration, 1 central VPC peers with 4 other but NO TRANSITIVE PEERING you cannot transit through a VPC, you must peer the direct links between the VPCs you want

to talk to each other

- n=nodes, Point to Point connections = n x (n-1)/2 so for a 5 node network 5×4=20 divided by 2 = 10 connections
- A VPC endpoint creates a private connection between your VPC and another AWS service without requiring access over the Internet, through a NAT, VPN, or direct connect
- Endpoints are virtual devices. You use endpoint policies to control access to resources in other services
- VPC endpoints are used because most AWS services are NOT IN A VPC and so they must be connected to externally
- Usually they connect over the Amazon public network
- VPC endpoints connect without using the public network for better performance
- VPC endpoints are supported within the same region only
- VPC endpoints for additional services beyond S3 will be added in the future
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_N
- Tenancy shared or dedicated (dedicated hardware is expensive avoid it if at all possible)
- Once a VPC is set to dedicated hosting, it is not possible to change the VPC or the instances to Default hosting
- You must re-create the VPC to revert to a shared tenancy Create AMIs of all your instances. Create a new VPC with Default as the hosting tenancy attribute, and use them to create new instances using Default tenancy
- Must create your own subnets, in the name tag use the subnet and AZ name. “172.16.33.0 us-west-1a” and then “create subnet”, select the VPC where the subnet is to be created
- Security groups and network ACL’s can span multiple availability zones but subnets MUST be in only one AZ and you are not allows to have the same subnet span across AZ’s
- Create as many VPC’s as needed (there are capacity limits though). For example: Shared, Development, Test, Production
- Account patterns that drive the multiple VPC model, Shared services (DNS, email) Development, Test Production
- A second VPC approach is where you set up smaller AWS accounts instead of the multiple VPC approach of using one

account. Maybe individual accounts is better due to isolation. Maybe it's one of these use case cases that will drive the decision. Is there more than one IT team or many? Billing argues for multiple accounts is a better solution than multiple VPC under one account. Are there compliance issues that require accounts to be separate?

- Size of organizations structure and may drive the VPC / Multiple account decision
- If you have big data applications, put it in its own VPC for example
- Configure the default VPC as other VPC's e.g. add more subnets
- Default VPC security group permissions defaults to wide open
- IPsec is the security protocol supported by Amazon VPC
- The CIDR block is specified upon VPC creation and cannot be changed later

Good VPC Analogy and basic setup:

- VPC=The City
- Subnets=Postal codes
- Route tables=roads
- Network ACLs=Security gates
- Servers and services=buildings
- Security groups=security guards
- Benefits of using VPCs in an AWS account:
- Assign static IPv4 addresses to our instances in VPC
- Static IP addresses will persist even after restarting an instance
- Use IPv6 addresses with supported instances in a VPC
- VPCs allows instances to run on single tenant hardware
- Can define Access Control Lists (ACL) to add another layer of security to our instances in VPCs
- VPC allows changes to the security group membership of instances while they are running and take effect immediately

VPC Networking

- When creating a VPC, a route table is created by default. You must manually create subnets and an IGW however
- Minimum IPv4subnet size in a VPC is a /28 16-network subnets with 14 hosts each (16 block)
- Maximum IPv4 subnet IP address range in a VPC is a /16
- All VPC subnets can communicate with each other by default (when the subnets are provisioned, routes are automatically added to the routing table)
- You may only have one IGW (Internet gateway) for each Amazon VPC
- Security groups are stateful, if traffic is allowed in one direction, it is automatically allows back in the other direction
- Use the 10.0.0.0 /16 CIDR block as a recommendation
- Cannot use anything larger than a /16 CIDR block format is the base IP with a /x
- AWS reserves three IP addresses per subnet by default. .1 is the gateway, .2 is the DNS server (Route 53)and .3 is reserved by AWS for future use, there is also the base network and broadcast IPs
- Each subnet that you create is automatically added to the main VPC route table for internal routing
- If you want the subnet to be routed outside of the VPC, it must be manually added to the routing table use the “Create Internet Gateway” on the VPC dashboard and attach the default gateway to the VPC
- It is a common practice to create one public subnet multiple internal private subnets
- Route out to internet, VPC dashboard the route table, there are the local routes for internal communications, they refer it to as private
- Create a new routing table called PublicRoute select the VPC, then edit, add a new route, 0.0.0.0 and target is automatically defined which is the Internet gateway router
- There is no route to the internet on the main routing table for security reasons, use the internet routing table you created

- Associate the subnet you want to access the internet
- Public subnets and allow assign auto a public IP
- It is allowed to add a subnet to an Amazon VPC any time after it has been created, as long as the address range falls within the VPC CIDR block and does not overlap with the address range of any existing CIDR block that has already been assigned
- You can set up peering relationships between VPCs after they have been created
- In a custom VPC, if you create a EC2 instance in a public subnet, it will not be accessible via the internet until you apply an elastic IP or an Elastic load balancer instance
- The majority of resources will be on VPC private subnets and then use public subnets to control remote access
- Plan for a large number of private IPs to meet your requirements
- If you run out of available IPs you can't add more to that subnet
- Only one route table per subnet is allowed
- Main and custom route tables
- Best security practice is to use a custom route table for each subnet
- When a VPC is created, it automatically has a main route table that allows full access between all subnets
- When you create a new subnet, if you do not explicitly assign it to a custom routing table, it gets associated in the main routing table by default
- Put all subnets in a custom route table this allows you to control routing and what gets routed outside of the VPC
- A default subnet is created in each availability zone for each default VPC
- Never delete this default subnet
- Public subnet with a CIDR block of /20 (4096 IPs)
- To change a default subnet into a private subnet remove the route to the IGW
- Adding a new AZ is to a region the your default VPC in that region gets a subnet placed in the new AZ
- It is best practice to never use the default subnet (or VPC) for production, create a new VPC and subnets to get full control of them

- By creating a route out to the Internet using an IGW, you have made the subnet public
- Network ACLs are stateless, you must create a rule both directions
- You must disable source/destination checks on the NAT for it to work
- DHCP to get a host to resolve DNS names outside of AWS use the DHCP option set for the EC2 instance
- In the EC2-Classic network (no longer used), when stopping and starting the EC2, the elastic IP will be disassociated with the instance; in the EC2-VPC network, the EIP remains associated with the instance
- Regardless of the underlying network, a stop/start of an Amazon EBS-backed Amazon EC2 instance always changes the host computer

VPC Enhanced Networking

- All instances are on the same physical hardware for maximum performance in a placement group
- More packets per second
- Lower latency
- Less Jitter
- High-Performance Computing (HPC) cluster needs very low latency and high bandwidth between instances: use an instance type with 10Gbps network performance, put the instances in a placement group, enable enhanced networking on the instances

VPC Networking Elastic Network Interface (ENI)

- VPCs allow the creation of a dual-homed instance is by attaching an ENI (Elastic Network Interface) with different subnets to an instance can make the instance dual homed
- A virtual network interface that attaches to an instance in a VPC
- ENI have the following characteristics:
  - Primary private IP address
  - One or more secondary private IPs
  - One public IP address that can be auto-assigned to the elastic network interface for eth0 when you launch an instance
  - One or more security groups
  - MAC address
  - Source/destination check flag
  - description
- Attaching a ENI to a instance that is stopped, the term is a warm attach
- Attaching a ENI to a instance that is running, the term is a hot attach
- Attaching a ENI to a instance that is launching, the term is a cold attach
- Configured in the EC2 section of the console under configure instance details
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#attach_eni_launch

VPC peering

- VPC peering allows you to connect one VPC to another via a direct route using private IP addresses
- Use a separate VPC for separate functions and interconnect them
- Cannot peer across different regions, inside of a region only but can span availability zones inside of a region
- No transitive peering, for traffic to flow VPCs must be directly peered, you cannot pass through an intermediate VPC peer to reach another
- Cannot have more than one peer connection between the same two VPCs at the same time
- MTU is 1500 bytes, this is the standard ethernet frame size but opens the possibility for jumbo frame issues
- A placement group can span across peered VPC but there may be bandwidth limitations
- You cannot reference a security group from a peer VPC as a source or destination for ingress or egress rules in your security group. Instead, reference CIDR blocks of the peer VPC as the source or destination of your security groups ingress or egress rules
- VPC connections have redundancy using two parallel IPsec tunnels
- It is acceptable to have an internet gateway and a VPC peering connection on the same VPC (IGW and VPG in each VPC but only one of each)
- A Customer gateway (CGW) is a physical or software application that is located at your on-premise data center. It is the VPN connector on the data center side of the connection and must be configured with a static public IP address
- The Virtual Private Gateway (VPG) and customer gateway are the two connectors on both sides of the VPN connection and both are required
- A Public Virtual Interface allows you to interface with AWS resources that have a public endpoint (like S3 storage or Elastic Load Balancers)
- Private DNS values cannot be resolved between instances in peered

VPCs

- A VPC peer count of 50 or more is normal
- Full mesh formula is x(x-1)/2
- To peer VPC between regions, use AWS Direct connect or a VPN connection since VPC peering is not supported between regions
- Configuring and implementing VPC peering is a very straightforward process
- Remember to add static routes on both ends pointing to the CIDR block at the remote VPC with the next hop being the VPC peering connection
- http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/peer configurations.html

VPC peering security

- Two-way handshake to establish a peering connection
- Routing controls: Routing tables control the local subnets that can route to remote subnets
- Security groups control what traffic a subnet can send and receive
- No edge-to-edge routing or transitive trusts: Reduces inadvertently creating unexpected network connections

Identity Access Manager: IAM

- AWS management application for users and device/service level of access in the AWS console
- Granular permissions for users and devices are configured in IAM
- Shared access to users in the account
- Centralized control of account users for authentication and authorization as well as service roles
- 2 factor / multifactor authentication is optional and configured in IAM
- Can provide temporary access for users and devices to AWS services
- Offers a password rotation policy
- Groups: a collections of users under one set of permissions, you define a group, put users into the group and assign permissions to the group. This simplifies administration since each user does not have to be assigned roles and permissions individually
- Roles: Create roles and assign them to resources such as EC2, S3 and others to define what they can access and what operations can be performed
- Policies: A document that defines one or more permissions, attach polices to users, groups or roles
- Not region specific, IAM it is universal for the AWS account
- Secure root account and use multifactor on this account use google authenticator for Android to secure access to IAM is a best practice. Lock it down because if an intruder gains access to AIM they have control over your account and the resources inside the account
- When you create a new user they have no permission to do anything, all permissions must be explicitly granted
- Go to policies to assign the permissions
- Policy documents are written using the JSON format
- Create the policy options and then attach, the policy to a user or group
- Create an admin group in IAM and assign users and attach a policy

for users that need IAM access. This can be very granular, such as S3, or DynamoDB administrators, and then what they're allowed to do with those services, the list of options is endless and you can define your own if the preconfigured options do not meet your requirements

- Roles can be used for object to object access EC2->S3
- By default a service will not be allowed to access another, such as above with an EC2 instance needing to access a specific S3 bucket, AIM is where you create the policies to enable access
- Root account has full permissions, use with caution and limit access to the root of your AWS account

- When you get the key values and password, save them because you will never see them again. AWS does not keep them and they are presented only when an account or service is created. Store them in a secure place.
- IAM allows you to manage users and their level of access to the AWS console
- AIM offers a centralized administrative control point of your AWS account
- Global to AWS, not region based, there is one AIM control point per AWS account
- Identity Federation (Active directory, Facebook, LinkedIn can be used for account credentials
- Provide temporary access for users/devices and services where necessary

- Roles: You create roles and can then assign them to AWS resources Role for EC2 instance to access S3 for example
- Policies are statement that defines one (or more) permissions. Polices are attached to users, groups of roles
- I is a best practice to customize your AWS account login URL to make it more readable, must be globally unique
- Generally never use the global root account for normal administrative activates
- Policies = JSON key/value pairs
- New Users are assigned an access key ID & secret access key when

first created, only viewable once so download it & store it in a secure place

- AIM is also integrated with the AWS marketplace
- Run applications on an Amazon EC2 instance with an assigned IAM role to access needed resources and services
- IAM roles provide a temporary security token to an application using an SDK
- IAM polices require a service name and an action
- IAM controls access to AWS resources only. Installing ASP.NET will require Windows operating system authorization, and querying an Oracle database will require Oracle authorization. However, launching and EC2 instance or adding a message into a SQS queue are controlled by IAM
- IAM security features include Multi-factor authentication and password policies
- The benefits of using Amazon EC2 roles include no key rotation is necessary and credentials do not need to be stored on the EC2 instance, EC2 roles must be assigned a policy
- Temporary security tokens are used by EC2 roles and Federations
- To lock down administrator user accounts add Multi-factor authentication to the accounts, Implement a password policy and apply a source IP address condition to the policy that only grants permissions when the user is on the corporate network
- IAM permits users to have no more than two active access keys at one time

IAM Roles

- IAM roles re configured to grant access to AWS services
- Roles for EC2
  - enable apps running on EC2 to make AWS API calls
  - AWS manages all security credentials
  - No need to put keys on the instance which is a security risk
  - Easy to attach and detach a role to a new or existing instance
  - Add/update permissions without logging into the instance
- Service roles:
  - Grant AWS series access to perform actions on your behalf
  - Control permissions that service can run on
  - Track actions AWS service perform on your behalf using CloudTrail
  - Example's: AWS config, AWS OPSworks, and AWS Directory Service
- Service-linked roles:
  - Grant AWS services access to perform actions on your behalf
  - Pre-defined permissions that the linked service requires
  - Protection from inadvertently deleting a role
  - Tracks actions of AWS services person on your behalf using CloudTrail
- Short term access, roles are temporary (granted for a maximum of 36 hours and then need to be refreshed)
- Federates identities into AWS such as active directory
- Use roles to enable cross account access either internally or with other AWS accounts
- Federated ID
  - AD connection or SAML, IAM roles grant permissions to your federated users enable federated single sign
- Open ID connector

    - Enables your app users to sign in using an independent directory service
- Cross account access
    - Assume a role in another account
- Switch role in the console
    - Sign into the console and switch between different roles
- To access an IAM role
    - Secure Token Service outputs temp credentials
    - API assume role, AssumeRoleWithSAML, AssummeRoleWithWebIdentity, GetSessionToken
- Intended to have multiple users assume a role
- https://aws.amazon.com/blogs/security

IAM Security Token Services (STS)

- Create and provide users temporary security credentials that controls access to AWS resources
- STS works like long-term access key credentials that IAM users use
- Default valid time for a session token is 1 hour
- Temporary security credential are valid for the time specified in the AssumeRole or the time configured in the SAML authentication resonses SessionNotOnOrAfter value, whichever is shortest.
- Token active time is a minimum of 900 seconds (15 min) to a maximum of 3600 seconds (1 hour) 1 hour is the default
- With multiple accounts you can configure cross account access using tokens
- Cross account access allows IAM users access to AWS resources they don't already have access to, such as resources in another AWS account
- Cross account access is useful for existing IAM users to get temporally elevated privileges in another AWS account
- Can authenticate off of a web service like facebook, google, Amazon r other services using OpenID connect compatible provider with AssumeRoleWithWebIdentity that returns a set of temporary security credentials
- Use GetSessionToken for temporary security credentials lasting more than one hour valid from 15 minutes to 36 hours with a default of 12 hours

Network Address Translation: (NAT)

VPC NAT Instances

- Private NAT not managed by AWS
- You manage the NAT instance
- Outdated, replaced by NAT gateways
- Must go to the Marketplace and select a NAT AMI Gateway or NAT AMI Instance
- NAT gateways came out in 2016 and are a better solution than NAT instances
- Community AMIs and search “NAT” There are a lot of NAT instances in the Amazon marketplace to choose from. Choose the Amazon ones at the top, amzn-ami-vpc-hvm-2015.09.a.x86_64_ebs for example
- Really, do not go with the instance since they are outdated, go with NAT Gateways since they are auto patched, auto scale and is fully managed service from AWS
- NAT Instances must be behind a security group (gateways do not)
- NAT must connect to a public subnet
- Must use a public IP address for the outside NAT function, it must be internet accessible
- Must always be behind a security group
- Disable the source/destination check on your NAT instances to get it to work
- Instance disable source destination check (traffic goes through the NAT instance and does not terminate). This is very important as NAT will fail if source/destination checking is enabled
- Create a route from the private subnets to the NAT instance, Main/default routing table 0.0.0.0/0 -> target is the NAT instance
- The NAT instance is a single point of failure but you can use an auto scaling group. Multiple public subnets with multiple NAT instances, no real good choices. Use NAT gateways
- Slow EC2 instance can affect NAT performance, generic Linux AMI configured for NAT

NAT Gateway

- NAT gateway were introduced in 2016 much better than NAT instances
- NAT gateways are a fully managed AWS service
- AWS does all the NAT gateway maintenance, it is a managed service
- NAT gateway AWS does the patches and it scales, it is an AWS managed service
- NAT gateway is an option on the left hand side of the main VPC console screen
- Instances are EC2 images, in community marketplace AMI's and select AWS branded AMI, use the public facing security group. Really do not use this approach, use the gateway
- Create a route out to the internet from the NAT gateway. Disable source/destination check
- Route from private subnet to the NAT box. Watch the cloud guru VPC lecture on this
- NAT gateway, no security groups need to be defined as AWS does it all for you
- Deploy in public subnet and on the private side, the EC2 instances point to the NAT gateway inside interface
- Crates an elastic IP automatically
- Automatically assigned a public IP address
- Add a route table to make 0.0.0.0/0 with next hop to the NAT gateway. This is in the public routing table
- Does not need a source/destination check or behind any security groups
- User guide VPC/Networking/NAT
- 10 GBPS burst throughput (this is what they scale up to)
- No need to disable source/destination checks
- Cannot associate with a gateway
- Network ACL's work with gateways

NAT vs VPC Bastion (jump) server

- Jump servers, SSH or RDP to the bastion host from the outside internet and then from the jump server, initiate a connection in the private VPC/Cloud IP subnet to the devices
- The bastion host is hardened and locked down
- NAT outgoing is common but not for incoming traffic, use the bastion host for incoming sessions
- Put a bastion host in each public subnet
- Can use auto scaling group on the bastion hosts and then Route 53 / DNS will handle the new addressing
- NAT is used to provide internet traffic to EC2 instances in a private subnet for patching and updates usually
- Bastion hosts are used for administration of EC2 instances (using SSH or RDP) in private subnets
- SSH uses the private half of the instance's key pair locally and the public key is on the bastion host

VPC Security Network Access Control Lists vs. Security Groups

- Search VPC Security to AWS paper "Comparison of Security Groups and Network ACLs"
- Configured in the VPC dashboard

Network Access Control Lists (NACL)

- Subnet level (network level not host level)
- Supports allow and deny statements IP address and port number
- Stateless, need a rule for both inbound and outbound
- Top to bottom rule evaluation, if there is a match traffic is allowed and checking stops
- Applies to all instances in the subnet
- Custom ACLs deny everything by default
- Standard default ACL allows all be default
- One subnet can only be associated with one ACL
- Defined in the console under VPC > Security > Network ACL
- Defines both inbound and outbound rules
- It is best practice to space out rule numbers in increments of 100
- Rules are evaluated sequentially with a first match like any other router ACL, when there is a match the evaluation process stops
- Create the ACL after giving it a name
- A new ACL is not associated with any subnets and is deny any/any, , you must configure the rules and apply the NACL to a subnet
- Use “subnet associates” to apply the rule to the subnet
- Internet facing ACL permits for http, https, SSL, RDP are common 80, 443, 22, 3389, (deny all)
- Source is common at 0.0.0.0/0 (everything / any)
- Ephemeral ports may be needed for internet facing servers, open 1024-65535 as a custom TCP rule do for outbound and inbound, open outbound ephemeral for SSH to work since SSH return traffic uses the higher ephemeral ports
- Each subnet in a VPC must be associated with a ACL, if you do not explicitly associate a subnet with an ACL it is automatically associated with the default ACL
- When you associate a subnet with a different ACL, the subnet is removed from the ACL it was in which is usually the default ACL and the default is permit any/any
- If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL

Security Groups

- Instance level, resource level security. More granular than the subnet level NACLs
- Allow lists only anything that is not explicitly allowed is denied
- Stateful: traffic allowed out is automatically allowed back in and vice versa
- Open a port and it is allowed both directions (stateful)
- Evaluates all rules before deciding to allow traffic
- All instances must belong to a security group
- Allows you to permit or deny TCP and UDP ports at the device level
- All traffic in a security group is denied by default
- Defined using traffic direction, port, protocol and source and/or destination address
- If an instance belongs to two security groups, each security group is aggregated to create one set of permissive rules, so the result is a combination of all traffic allowed by the rules in both security groups
- AWS provides the security group functionality as a service, but you are responsible for configuring their own security groups
- Security group updates (changes) are applied immediately
- Security groups are defined at the instance virtual network interface at the hypervisor level, they are host based and not subnet based
- It is a best practice to create security groups with inbound rules for each functional tier (web/app/data/etc.) within an application with inbound rules defined to allow traffic in from the source tier directly above it
- The default security groups default settings are:
    - All inbound is denied
    - Allow all outbound traffic
    - All subnets in the security group can talk to each other in the default security group
- CLI example: revoke-security-group-ingress will remove rules from inside a security group

- Instances associated with a security group can't talk to each other unless you add rules allowing it (exception: the default security group has these rules by default)
- When you specify a security group as the source for a rule, this allows instances associated with the source security group to access instances in the security group
- Using Security Groups, traffic can be restricted by any IP protocol, by service port, as well as source/destination IP address (individual IP or Classless Inter-Domain Routing (CIDR) block)

Internet gateways IGW

- Allow communications between instances in a VPC and the internet
- Are horizontally scaled, redundant, and highly available by default
- AWS managed service
- Provide a target in your VPC route tables for internet routable traffic
- To enable access to/from the internet into your VPC subnet, do the following:
  - Attached an internet gateway (IGW) to your VPC
  - Ensure that the subnet's route table points to the IGW
  - Ensure that instances in the subnet have public IP addresses or Elastic IP addresses
  - Ensure your NACLs and security groups all the relevant traffic to flow to/from your instance
- Use NAT for private RFC 1918 address spaces to access the internet
- For Internet connectivity you need a public IP address, an IGW and a route to the IGW

Flow logs

- Console > VPC > actions > create flow log
- Packet flow utility and reports to CloudWatch, captures IP traffic flow information of your resources
- Can create a flow log for a VPC, subnet or network interface
- Create new IAM role for flow logs with a destination of CloudWatch into a log group
- Each interface has a unique log stream record identifier
- In management tools, in CloudWatch and click select on logs and create a new log group
- In the CloudWatch console, select logs and create a log stream
- Data is similar to a standard a syslog format with a timestamp, endpoint IP address etc.
- Use flow logs for troubleshooting connectivity, security issues and testing network access rules

Elastic IP addresses: EIP

- Amazon provides an Elastic IP Address with an AWS account
- 5 elastic IPs per VPC by default
- An Elastic IP address is a public and static IP address based on IPv4 protocol
- It is designed for dynamic cloud computing, EIPS move between instances, preserving DNS
- This IP address is reachable from the Internet
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface such as a load balancer
- If there is no specific IP address for a EC2 instance, then you can associate the instance to the Elastic IP address included with your AWS account
- If you associate additional EIPs with that instance, you will be charged for each additional EIP associated with that instance per hour on a pro rata basis
- Additional EIPs are only available in Amazon VPCs
- The discourage the use of EIPs that are idle, AWS will impose a minor hourly charge when these IP addresses are not associated with a running instance or when they are associated with a stopped instance or unattached network interface
- The EIP remains associated with the instance in a VPC when it is stopped
- You are billed for an elastic IP hourly when it is NOT being used
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html
- When an Elastic IP address gets associated with an instance or the primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses
- You cannot reuse a public IPv4 address. For more information, see Public IPv4 Addresses and External DNS Hostnames
- You are allowed to disassociate an Elastic IP address from one

resource, and re-associate it with a different resource

- EIPs are portable inside of a VPC
- A disassociated Elastic IP address remains allocated to your account until you explicitly release it
- An Elastic IP address is for use in a specific region only
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS hostname of the instance changes to match the Elastic IP address
- Aa public DNS hostname is resolved to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance
- If you stop an instance that is using an EIP, its Elastic IP address remains associated (need verification)
- Instances support multiple IPv4 addresses, and each one can have a corresponding Elastic IP address
- Under VPC console select Elastic IPs
- If you associate additional EIPs with that instance, you will be charged for each additional EIP associated with that instance per hour on a pro rata basis
- Used to mask a failure of an instance or software by allowing your users and clients to use the same IP address with replacement resources
- If one instance crashes, clients can use the same IP address to reach the replacement instance

Route 53

Domain Name Systems overview

- resolves domain names to IP addresses
- Uses port 53 TCP/UDP (primarily UDP based)
- DNS is used to convert friendly domain names such as https://mycompany.com into an Internet Protocol IP address such as https://8.8.8.8. There are two types if IP Addressing
- DNS is a very reliable and cost effective way to route end users to Internet applications by translating names like **www.example.com** into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other
- Limited IPV6 inside of AWS, no ipv6 on EC2 but limited Route53 support
- IPv6 is now supported in VPC's as of December 2016
- Top level domain names, .com, .edu .co.uk, .com.au etc.. also ".cloud"

AWS Route 53 DNS services

- Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) service
- Amazon Route 53 supports both IPv4 and IPv6

- IPv4 has a 32 bit address field
- IPv6 is the new version of IP Address available, and the address space is 128 bit field
- IPv6 is supported in Route 53, VPCs and EC2
- Route 53 charges for CNAME requests but not for alias record requests
- Always use an Alias Record over a CNAME if you can
- If you create a new subdomain for your website and you need to point it to an ELB, use a CNAME
- Route53 has a security feature that prevents internal DNS from being read by external sources. The work around is to create a EC2 hosted DNS instance that does zone transfers from the internal DNS, and allows itself to be queried by external servers
- Main Route 53 features are domain registration, DNS services and health checking (not load balancing)
- DNS uses port 53 and is UDP based and also supports TCP
- If the DNS response is greater than 512 bytes then TCP is used this would be seen in zone transfers
- Route 53 supports public and private hosted zones
- Route53 costs around $1.50 per month

- ELB's do not have pre-defined IPV4 addresses, you can only resolve then using a DNS name. (this is changing in 2018)
- Use Route53 internal to a VPC in AWS for domain resolution that does not face the outside world as an option
- Understand the differences between an Alias Record and a CNAME. CNAMEs are billable, use alias on naked domain name mappings if possible
- Understand Route 53 routing policies of Simple, Weighted,

Latency, Failover and Geolocation

- Excellent AWS links for Route 53:
- How should I create record sets?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resourc record-sets-creating.html
- Wondering how to choose the correct routing policy?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing policy.html
- What is an alias record? Should I use it?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resourc record-sets-choosing-alias-non-alias.html
- What are the assigned name servers for my hosted zone?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/GetInf
- What will be the best practice for migration process?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Migrat
- Is there a way to have additional availability from DNS layer?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html
- Can I use Route 53 as my private DNS service?
- http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted zones-private.html
- Want some hands on experience?
- https://aws.amazon.com/training/intro_series/#networking-1
- How do I transfer a domain to Route 53?
- https://aws.amazon.com/premiumsupport/knowledge-center/transfer-domain-to-aws/
- How do I verify that resource record sets are accessible from the Internet?
- https://aws.amazon.com/premiumsupport/knowledge-center/route-53-reachable-resource-record-sets/
- What can I check when I'm unable to access my website when using the Route 53 DNS Service? https://aws.amazon.com/premiumsupport/knowledge-center/route-53-dns-website-unreachable/

Top Level Domain Name

- The top level domain names are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database which is essentially a database of all available top level domains
- These include things like .com or .net
- Big database of top level domain names
- It can be viewed at http://iana.org/domains/root/db
- Domain registrars, all of the names in a given domain name have to be unique, there has to be a way to organize this all so that domain names aren't duplicated
- Domain registrars accomplish this function
- A registrar is an authority that can assign domain names directly under one or more top level domains
- The domains are registered with InterNIC, a service of ICANN which enforces uniqueness of domain names across the internet
- Each domain name becomes registered in a central database known as the WhoIS database

SOA (Start of Authority)

- The Start of Authority stores basic properties of the domain name and the zone that the domain is in
- It contains the following information:

- The primary name server for the domain, which is ns1.dnsprovider.com or the first name server in the vanity name server list for vanity name servers.
- The responsible party for the domain, which is admin.dnsprovider.com
- A time-stamp that changes whenever your domain is updated
- The number of seconds before the zone should be refreshed
- The number of seconds before a failed refresh should be retried
- The upper limit in seconds before a zone is considered no longer authoritative

- The negative result TTL for example, how long a resolver should consider a negative result for a sub-domain to be valid before retrying
- Each domain MUST have a SOA record

Name Servers

- Name server records that are used by Top Level Domain Names servers to direct traffic to other DNS servers which contain authoritative records
- For example, you can create a route 53 zone, you will be provided 4 NS records
- Configure your domain name to point to these NS Records and then you use Route 53 to manage all your DNS records for the Domain Name

Zone Files

- A zone file is a text file that contains the mapping between domain names and IP Addresses
- Zone files reside on name servers and define the resources available under a specific domain

Hosted Zones

- Hosted Zones are groups of resource records sets hosted by Amazon Route 53
- Similar to standard DNS zone files
- Hosted zones are used to manage records under a single domain name
- Hosted zones will have metadata and configuration information
- There are two types of hosted zones:
- Private – These are used to provide configuration information on how to route traffic for a domain and its sub-domains within one or more Amazon Virtual Private Clouds (VPCs) Additionally:
- Amazon VPC Settings – To use private hosted zones, you must set the following Amazon VPC settings to true:

- *enableDnsHostnames*
- *enableDnsSupport*

- Amazon Route 53 Health Checks – In a private hosted zone, you can associate Amazon Route 53 health checks only with failover resource record sets
- Split-View DNS – To maintain internal and external versions of the same website or application (for example, for testing purposes), you can configure public and private hosted zones to return different internal and external IP addresses for the same domain name
- It is permissible to associate a VPC with more than one private hosted zone, but the namespaces must not overlap
- You cannot create NS records in a private hosted zone to delegate responsibility for a sub-domain
- Custom DNS Servers – If you have configured custom DNS servers on Amazon EC2 instances in your VPC, you must configure those DNS servers to route your private DNS queries to the IP address of the Amazon-provided DNS servers for your VPC
    - Public – These are used to provide configuration

information on how to route traffic for a domain and its sub-domains on the Internet

- It is critical to understand that CNAME records are not allowed for hosted zones in Amazon Route 53. You need to use Alias Record

Domain Registrars

- Domain names need to be unique across the Internet
- Domain registrars are responsible for ensuring this and management of all domain names
- Registrars can associate domain names directly under one or more top-level domains
- Route 53 offers Domain Registration services and management
- This is a recent change and so you no longer need to host your domain name with third party registrars and
- You can manage all DNS registrar activity with the Route 53 service
- Route 53 support domain name registrations for both generic and geographical top level domains

DNS record types

- A Record: The most basic record which enables point a domain name to an IP Address
- TTL: Time to Live which is the length of time in seconds that you want the DNS resolver to cache values for a resource record before submitting another request to Route 53 to get current values for a record
- During the TTL period, the DNS resolver will respond to requests from its cache
- Amazon Route 53 charges based on the number of DNS queries made and so a longer TTL can help reduce your charges
- Be careful that you don't have stale records out on the Internet
- When performing DNS migrations, reduce the TTL. By default most TTL are valid for 2 days
- CNAMES: These are Canonical names that can be used to resolve one domain name to another
- Instead of having multiple A records all pointing to an IP address, you can setup CNAME records to point one domain name to another
- CNAME records maps a name to another name it should be used only when there are no other records of that name
- Alias Records are similar to CNAME records that can map one DNS name www.example.com to another. However, Alias records are special in that unlike CNAME records, you can map an Alias record to a Zone Apex, e.g. company.com, i.e. without the host name like 'www'. You cannot do the same using a CNAME record
- Alias resource record set contains a pointer to a CloudFront distribution, an Elastic Beanstalk environment, an ELB Classic or Application Load Balancer, an Amazon S3 bucket that is configured as a static website, or another Amazon Route 53 resource record set in the same hosted zone
- Elastic Load Balancers come with a DNS name and you are not provided with an IP Address (this is changing in 2018). You can therefore use a CNAME or Alias Record (if you are using Route

53) to host your DNS zone and then configure your company domain name to point to the Elastic Load Balancer

- Alias records allow you to route DNS queries to your load balancer for the zone apex of your domain (for example, mydomain.com)
- Amazon Route 53 doesn't charge for DNS queries for alias records
- Amazon Route 53 will automatically recognizes changes in the records sets that the alias records refers to
- If an Alias Record points to an ELB and if the IP Address of that ELB changes, Amazon Route 53 will reflect those changes in the replies for the zone apex name
- Suppose an alias resource record set for mydomain.com points to an ELB load balancer at lb1-8835.us-west-2.elb.amazonaws.com. If the IP address of the load balancer changes, Amazon Route 53 will automatically reflect those changes in DNS replies for mydomain.com without any changes to the hosted zone that contains resource record sets for mydomain.com
- An alias resource record set only works inside of Route 53. This implies that both the alias resource record set and its target must exist in Amazon Route 53
- Mail Exchanger (MX): MX records define mail servers for a domain and routes email messages to your mail server
- MX Records point to an 'A' record which then points to the IP Address of the Mail Server or Load Balancer in front of your mail servers
- AAAA records are used to route traffic to a IPv6 address (A records are IPv4)
- A PTR record resolves an IP address to a domain name is called reverse DNS
- SPF records are used to verify authorized senders of mail from your domain and are used to prevent spoofing and spam
- All zones must have a SOA record by default (start of authority)
- TXT record can be used to store human readable information about a server, network, and other accounting data with a host. TXT records are arbitrary and unformatted

A Records Address record

- Fundamental type of DNS record and that “A” stands for address
- Most basic DNS record type
- The A record is used by a computer to translate the name of the domain to the IP address
- www.tipofthehat.com -> 10.11.12.123
- Used to point a domain or subdomain to an IP address

Alias Records

- Route53 specific and not standards based
- Alias records map resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets that are configured as websites
- Alias records work like a CNAME record in that you can map one DNS name (www.mydomain.com) to another “target” DNS name (elb_abc.elb.amazonaws.com
- A CNAME can’t be used for naked domain names (base domain with no www or anything) (zone apex). Cannot have a CNAME for http://mydomain.com, it has to be either an A record or use an alias that allows you to map naked domain names naked domain names are ones that are have no prefix such as “www” only the domain name itself
- Maps a naked domain name (apex) to Elastic Load Balancer for example
- Route53 will automatically update the IP address if the load balancers IP changes
- ELB’s do not have a predetermined IPv4 address, you resolve to them using a DNS name
- It is a best practice to use Alias records over CNAMES, there no AWS charge and mapping naked domain name offer a lot of flexibility

CNAME Canonical name

- Resolves one domain name to another
- Example: A mobile website with the domain name http://m.mydomain.com that is used for when users browse to your domain name on their mobile devices. You may also want the name http://mobile.mydomain.com to resolve to the same address, in this case you would use a CNAME record to point one domain to another
- Points from one domain to another domain
- Eliminates the need for multiple A records pointing to the same IP address
- CNAMES are pointers to another domain name mobile.tipofthehat.com cname points to the www.tipofthehat.com A record for example
- Can resolve across completely different domains

NS Records Name Server

- NS stands for Name Server records and are used by Top Level Domain servers to direct traffic to the correct DNS server which contains the authoritative DNS records
- The domain is delegated to other DNS servers
- For example, all ".com" sub-names such as "mydomain" are delegated from the ".com" zone contain a NS name for all .com sub-names
- Top level domain authorities place NS records for your domain in the TLD name servers pointing to "mydomain" DNS servers

SOA records Start of Authority

- Contains information about:
- The name of the server that supplied the data for the zone
- The administrator for the zone
- Current version of the data file
- The number of seconds a secondary name server should wait before retrying a failed zone transfer
- The maximum number of seconds that a secondary name server can use data before it must either be refreshed or expire
- Default number of seconds for the time-to-live file on resource records

TTL record Time to Live

- The amount of time that a DNS record is cached on either the resolving server or the users local PC is “Time to live” (TTL) measured in seconds
- The lower the time to live, the faster changes in DNS records take to propagate throughout the internet
- For DNS migrations, reduce the TTL value to 300 seconds two days ahead of time (since the default TTL is two days). The 300 second TTL propagates across the internet DNS system as 300 seconds, or 5 minute cache time and the new IP address assigned to the domain name will take over with a 5 minute delay

Route 53 Routing Policies

- Route 53 provides you with 5 different routing policies. These are:

- Simple
- Weighted
- Latency
- Failover
- Geolocation

- Simple is the default routing policy when you create a new record set
- Simple is most commonly used when you have a single resource that performs a given function for your domain like a lone web server that serves content for the http://tipofthehat.com website
- There is no intelligence built into simple routing
- Weighted routing policies let you split your traffic based on different weights assigned. For example you can set 10% of your traffic to go to US-EAST-1 and 90% to go to US-WEST-1. Route53 splits the traffic
- Or split between load balancers inside of a single region, blue/green site testing
- Latency routes traffic based on the lowest network latency for the end user (i.e. which region will gives the user the best response time)
- Use latency-based routing you create a latency resource record set for the EC2 or ELB resource in each region that hosts the website. When Amazon Route53 receives a query it selects the latency resource record set for the region that gives the user the lowest latency. Route53 then responds with the value associated with the record set
- Failover is an active/passive Route 53 option. If host your primary site at US-WEST-1 and a Disaster recovery site in US-EAST-1. Route53 will monitor the health of your primary site using a health check. A health check monitors the health of your end points, if

they are not reachable, then Route 53 will service IPs for the backup site at US-EAST-1

- Geolocation routing directs traffic will be sent based to the nearest geographic location of the end user (i.e. the location from which DNS queries originate). All requests from Japan will be routed to be routed to the EC2 or ELB services that are specifically configured or Japanese users. These servers may be configured for the Japanese language with pricing in Yen. Granularity is Continent, Country, State

Simple Routing Policy

- Default routing policy when you create a new resource
- Use the simple routing policy when you have a single server that performs a given function for your domain
- For example, Route 53 will respond to DNS queries based only on the values in the resource record set such as responding with the IP address of an A record
- There is no redundancy or intelligence; you would use it to connect to a single web server for your domain name
- Route 53 with simple routing will respond to DNS queries based on the values in resource record set

Weighted Routing Policy

- With the weighted routing policy, a weight is defined according to which multiple resources will handle the load
- If there are two webservers, you can divide load in 45/55 ration between these servers for example
- Weighted Routing enables you to associate multiple resources with a single DNS name.
- If you have multiple resources performing the same function, you can have Route 53 direct a percentage of traffic to one resource and remaining traffic to another resource
- Typical uses cases include:

- You host you web sites in both the Japan and Singapore regions and want to direct 50% of your traffic to web servers in the Japan

region, and 25% of the traffic to servers in the Singapore region

- A/B testing and want to direct a small percentage of web requests traffic to a second load balancer that sends traffic to servers that have a new version of the website
- In order to configure weighted routing you will need to:

- Create two or more resource record sets that have the same DNS name and type
- Assign each resource record set to a unique identifier and a relative weight

Route 53 will search for a resource record set or groups of recourse records sets and select one record from the group using the following formula:

$$\frac{\text{Weight for a given resource record set}}{\text{Sum of the weights of the resource record sets in the group}}$$

Latency Based Routing

- Route 53 will respond to DNS queries with the servers that have the best latency
- Routes traffic based on the lowest network latency for the end users so they have the fastest responses
- Use latency based routing when resources that serve the same functions are located in different availability zones or regions
- Ensures that users around the world have access to your resources as fast as possible regardless of location

Failover Based Routing

- Route 53 is configured for active-passive failover
- Default servers accepts all traffic and if they fail, traffic gets rerouted to backup servers
- Configure active/passive by using failover based routing in Route 53
- One resource will get all the traffic when it is up, if it fails, all the traffic gets routed to second resource that is now active during failover

- Configure Route 53 to monitor the health of your primary endpoints using health checks
- Health Checks instruct Route 53 to send requests to endpoints to verify that they are operational
- Specify:
  - protocol to use
  - IP address and port
  - domain name and path to check
- Route 53 monitors the health of the active resource and if it should fail the failover routing policy will be applied and DNS will serve IP addresses associated with that domain name to the passive resource
- The following failover options are available:
- Active-active failover: Route 53 can detect that it's unhealthy and stop including it when responding to DNS queries
- Active-passive failover: A primary group of resources are available and a secondary group of resources is on standby in case all of the primary resources fail. When responding to queries, Amazon Route 53 includes only the healthy primary resources. If all of the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources responding to DNS queries
- Active-active-passive and other mixed configurations: You can combine alias and non-alias resource record sets to produce a variety of Route 53 operations
- Route 53 does not support failover resource record sets for private hosted zones

Geolocation based routing

- Geolocation DNS routing enables traffic routing on the geographical location of the users
- You configure users in Canada to be directed to the Toronto region and users in the Midwest to connect to the Ohio region
- Geolocation tracks the location of end users from where requests originate and directs them to the nearest AWS region where you have resources deployed
- Geolocation is used to direct traffic that belongs to a specific

geographical area to resources that have been configured specifically for them such as language and currency

- To ensure that only traffic from certain geographical regions can access your content for example where you have restrictions on distribution rights
- GeoLocation can be specified by:
  - Continent
  - Country
  - State within the US
- Geolocation works by mapping source IP addresses to the locations where the IP blocks have been allocated, this can sometimes create problems if some IP Address ranges have not been mapped to a specific location
- Create a default resource record set to field DNS queries from locations that cannot be identified, or where you do not specify geolocation records sets for
- If there is no default resource record set then Route 53 will return a 'no – answer' response for queries from those locations

DNS Health Checks

- Route 53 health checks monitors the health and performance of your AWS resources such as load balancers or EC2 instances
- You can specify intervals when Route 53 submits automated requests over the Internet to verify your endpoints are responding
- Configure a CloudWatch alarm for each health check is a supported feature
- Deploy web servers for example across multiple regions and multiple availability zones and enable health checking in Route 53
- If a health check determines that the underlying resource is unhealthy, Amazon Route 53 routes traffic away to other healthy resources

Virtual Private Networks VPN

- Connects your VPCs Virtual Gateway and your datacenter over the internet
- Available in hardware or software VPNs
- There is a range of third party VPN products on the Amazon marketplace
- A VPC VPN gateway can connect multipole remote customer networks into that VPC
- VPC Virtual Private Gateways use BPG peering to exchange routes
- One VPG (Virtual Private Gateway) per VPC is supported
- VPG is the VPCs connection, or interface, to the VPN from the VPC
- VPN's are created in the VPC console
- Remember to add the private data centers routes into the VPC route table to route back to the remote end
- Use BPG routing to configure routing priorities, policies, and weights (metrics) in their BGP advertisements to influence the network path between remote networks and the AWS VPC
- When using BGP, both the IPSec tunnel and the BGP peer connections must be terminated on the same customer VPN device, the customers VPN termination box must be capable of supporting both the IPSec and BGP protocols

Hardware AWS VPN

- IPSec hardware VPN between the VPC and your network
- Create hardware VPN connections between your corporate datacenter and your VPC and leverage AWS cloud as an extension of your corporate datacenter (Hybrid cloud)
- A CGW (Customer Gateway) is the customer side of a VPN connection, and an IGW (Internet Gateway) connects a network to the Internet. A VPG is the Amazon side of a VPN connection
- You can create a VPN connection from your network using an EC2 instance in your VPC that's running a VPN appliance application
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_V
- Set up an IPsec, hardware VPN connection between your AWS VPC and your remote network
- On the AWS side of the VPN connection, a virtual private gateway provides two VPN endpoints for automatic failover
- Configure the customer gateway, which is the physical device or software application on the remote side of the VPN connection (your end)
- For details, see Adding a Hardware Virtual Private Gateway to Your VPC document, and also reference the Amazon VPC Network Administrator Guide

Direct connect

- Dedicated private connection from a remote network to your VPC
- Direct Connect is a dedicated private connection from your network to an AWS VPC
- It is possible to combine a direct connection with an AWS hardware VPN connection to create an IPsec-encrypted connection.
- Alternate to access AWS thought the public internet
- Direct connect uses private dedicated circuits and not the internet
- Reduced network transfer costs
- Since it is dedicated bandwidth, there can be better application performance with predictable metrics
- Good for transferring large amounts of data between your facility and the AWS cloud
- Helps meet security and compliance requirements since the data does not go over the public internet
- Often used in Hybrid cloud architectures
- Private data center extension to the cloud
- Alternative to internet based IPSec VPNs
- Best ractice when using a direct connect design to use the IPSec VPN as a backup for failover
- Equinix, coresite, Eircom, Telecity Group, Terramark, Verizon and Level 3 are AWS interconnect providers
- The direct connect is to regions, this means that if you have a direct connect to US-EAST, it does not mean you do not have a direct connect to US-WEST, you would need to add a second direct connect to a accomplish that
- Not redundant by default, you would need to provision a second circuit as a backup
- Private Virtual interface in the VPC connects to VPC internal resources is supported
- Public Virtual Interface, used to connect AWS services that use public IP addresses, primarily used to connect to AWS services that do not live inside of a VPC and have public IPs such as S3 and DynamoDB

- Direct connect is a regional access service which means that you cannot connect though AWS to access the Internet
- The connection from on-premise to the Direct connect authorized provider is call the Cross-Network connection
- Uses 802.1Q VLAN tagging
- 1-10 GBS provisioned connections
- The cross-network connection is the connection between your on-premise data center and the Direct Connect Authorized Provider
- For cross region VPC-to-VPC peering can use direct connect
- http://aws.amazon.com/directconnect/details/
- http://docs.aws.amazon.com/directconnect/latest/UserGuide/getting

Cloud hub VPN

- Multiple AWS hardware VPN connections via your VPC to enable communications to various remote sites
- If you have more than one remote network (for example, multiple branch offices), you can create multiple AWS hardware VPN connections via your VPC which enables communications between these networks
- For more information, see Providing Secure Communication Between Sites Using VPN CloudHub in the AWS documentation

Software VPN

- AWS EC2 instance in a VPC that is running a software VPN appliance
- You can find images in the AWS marketplace

Auto scaling groups

- Dynamically reacts to changing load conditions by adding or terminating Amazon Elastic Compute Cloud instances
- Auto scaling defines a group with launch configurations and auto scaling policies that allow elasticity to add and remove EC2 instances based on load
- Auto Scaling is designed to scale out based on an event like increased traffic while being cost effective by not being charged for instance usage when it is not required
- Configure Auto Scaling to scale out as traffic increases and scale in when traffic decreases
- Configure the launch configuration to start new instances from a preconfigured Amazon Machine Image (AMI)
- Four different types of Auto Scaling plans:
    - Manual scaling,
    - Maintain current instance level
    - Scheduled scaling
    - Dynamic scaling
- Only the launch configuration name, AMI, and instance type are needed to create an auto scaling launch configuration
- Specifying a key pair, security group, and block storage device mapping are optional for the Auto Scaling launch configuration
- Launches instances from a specified Amazon Machine Image (AMI) created by you or from the standard offering. This allows you to launch custom configured instances
- Enforces a minimum number of running Amazon EC2 instances, if one fails, auto scaling will automatically launch a replacement to maintain a minimum instance count
- Elastic load balancers and EC2 instances feed metrics into CloudWatch that then triggers auto scaling events based on the threshold values you configure
- Amazon CloudWatch alarms execute auto scaling policies
- All of these services work standalone, when combined together they become more powerful and increase the control and flexibility

of demand resources in AWS

- Launch configurations specify:
  - Launch configuration name
  - AMI ID
  - Instance type
  - Key pair
  - Security groups
  - Block storage device mapping
  - User data
- Auto scaling groups specify:
  - Minimum number of EC2 instances
  - Desired capacity of running instances
  - Scaling policies to launch and terminate instances as demand increases and decreases
  - Maximum number of EC2 instances
- Requirements when attaching EC2 instances to an existing auto scaling group:
  - Instance must be in a running state
  - AMI used to launch the instance must exist in that region
  - The instance cannot be a member of another auto scaling group
  - Instance must be in the same availability zone as the auto scaling group
  - If the auto scaling group is using a load balancer, the instance and the load balancer must both be in EC2-classic or the same VPC
  - If the auto scaling group has an attached target group, the instance and the application load balancer must both be in the same VPC
- You can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it
- If you need to change the launch configuration for the Auto Scaling group, you need to create a new launch configuration and then update the Auto Scaling group with the new launch configuration
- When changing the launch configuration for the auto scaling group, any new instances are launched using the new configuration

parameters, but existing instances are not affected or modified in any way

- http://docs.aws.amazon.com/autoscaling/latest/userguide/LaunchCo
- Auto scaling works as a combination of three AWS services, ELB, Cloudwatch and Auto scaling working together
    - ELB and EC2 feed metrics to CloudWatch
    - Cloudwatch alarms execute the auto scaling policies to effect the size of your fleet
    - Auto scaling defines a group with launch configurations and auto scaling policies
- Auto scaling is available at no additional charge, you pay for the instance usage but not the auto scaling service
- To add to an existing autoscaling group:
    - The AMI must exist in the region
    - The instance must not be a member of another autoscaling group
    - The instance is required to be in the same availability zone as the autoscaling group
- By default, an Auto Scaling group determines the health state of each instance by periodically checking the results of the EC2 instance status checks
- If an instance fails the EC2 instance status checks, Auto Scaling considers the instance unhealthy and replaces it
- However, if you have attached one or more load balancers to your Auto Scaling group and an instance fails the load balancer health checks, Auto Scaling does not replace the instance by default
- If you have attached an ELB to your Auto Scaling group and an instance fails the load balancer health checks, Auto Scaling does not replace the instance by default

- Auto scaling is where you add and remove EC2 instances based on metrics defined on demand
- Grow and shrink groups
- Create an auto scaling group
- Scaling group allows you to define the number of instances and availability zones
- Choose all availability groups for redundancy, puts an instance in

each AZ or balances them out

- Scaling policies is where you define the additions and deletions of instances
- Launch groups are the “What”, the AMI used, the Instance type, you define a launch group and attach it to an autoscaling policy
- The Auto Scaling cool-down period is a configurable setting for your Auto Scaling group that helps ensure that Auto Scaling doesn’t launch or terminate additional instances before the previous scaling activity takes effect
- After the Auto Scaling group dynamically scales using a simple scaling policy, Auto Scaling waits for the cool-down period time to complete before resuming scaling activities
- Cyclic scaling occurs at a fixed interval (daily, weekly, monthly) scales up for anticipated events and then scales back down

Auto Scaling group Launch Configurations

- Know how you would go about making changes to an Auto Scaling group, fully understanding what you can and can't change
- You specify one launch configuration for an auto scaling group at a time
- You are not allowed to modify a launch configuration after you've created it
- When you want to change the launch configuration for your auto scaling group, you need to create a new launch configuration and then update the auto scaling group with the new launch configuration name
- When changing the launch configuration for the auto scaling group, new instances are launched using the new configuration parameters, but existing instances are not affected
- The AMI ID used in the autoscaling policy is specified in the launch configuration
- The launch configuration is referenced by the Auto Scaling group instead of being part of the Auto Scaling group because:
    - It allows you to change the Amazon Elastic Compute Cloud (Amazon EC2) instance type and Amazon Machine Image (AMI) without disrupting the Auto Scaling group
    - It facilitates rolling out a patch to an existing set of instances managed by an Auto Scaling group
    - It allows you to change of the security groups associated with the EC2 instances launched without having to make changes to the Auto Scaling group
- An Auto Scaling group may use either on-demand or spot instances
- Minimum requirements to create an auto scaling launch configuration:
    - Launch configuration name
    - AMI

    - Instance type
- Launch configurations specify:
    - AMI ID
    - Instance type
    - Key pair
    - Security groups
    - Block storage device mapping
    - User data
- When a launch configuration scales down the instance with the oldest launch config gets shot first
- The default termination policy is designed to help ensure that your network architecture spans across availability zones evenly
- Using the default termination policy, Auto Scaling selects an instance to terminate as follows:
    - Auto Scaling determines whether there are instances in multiple Availability Zones. If so, it selects the availability zone with the most instances and at least one instance that is not protected from scale in
    - If there is more than one Availability Zone with this number of instances, auto scaling selects the Availability Zone with the instances that use the oldest launch configuration
    - Auto scaling determines which unprotected instances in the selected availability zone use the oldest launch configuration. If there is one such instance, it terminates it
    - If there are multiple instances that use the oldest launch configuration, Auto Scaling determines which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances while minimizing the number of hours you are billed for Amazon EC2 usage.) If there is one such instance, Auto Scaling terminates it
    - If there is more than one unprotected instance closest to the next billing hour, Auto Scaling selects one of these instances at random.

CloudFront Content delivery network

- Aws managed content delivery network
- Requests for web content are automatically routed to the nearest CloudFront edge location, so content is delivered with the best possible performance
- If your application is content rich and accessed across multiple locations, use CloudFront to increase performance
- Edge / Regional Locations: CloudFront is a worldwide network of Edge and Regional Edge locations that caches the data. Edge locations serve all of the geographical areas across the globe
- Supported in free tier with traffic restrictions
- Content delivery based on geographic location
- Remote cache and AWS network access with points of presence worldwide
- Huge capacity at each edge location
- Not the same as a region or AZ, this is an access point that caches frequently requested data at many locations around the world for faster response times and reduced network traffic inside of AWS
- Built in security
- Feature rich that gives you complete control of configurations, API's and console
- Real-time metrics alarms for monitoring and management
- Supports static and dynamic content
- Accelerates dynamic content delivery by storing content at the edge locations worldwide with frequent updates for the origination content server to keep the data fresh
- Reduces traffic back to the origin servers or storage services
- Provides scalability, security and increased performance of applications
- Edge location: where the content is cached, this is separate from an AWS region or availability zone
- Edge locations are located in cities throughout the world
- Origin: The origin of all the files that the CDN will distribute. An origin can be a S3 bucket, EC2 instance, elastic load balancer or

Route 53 DNS

- You can upload data to S3 by writing directly to an Edge location
- Distribution: is the name given to the Content Distribution Network (CDN) which is a collection of edge locations
- Create a CloudFront distribution; DNS needs to be changed to point to the distribution IP at the edge location. You configure the distribution to point to your source content, this is referred to as the origin. If there are more than one origins and get routed based to the correct origin by looking at the URL patterns
- Two distribution types:
    - Web: access static or dynamic web content in any combination of up to 10 S3 buckets and custom origins
    - RTMP distribution the origin for RTMP (real time media streaming) data resides in an S3 bucket
- When a user requests content, the first web request goes to an edge location (offered by DNS), if the object is cached it is returned to the requester, if the data is not stored locally CloudFront will retrieve the data from the source and cache the content locally for future requests
- Cached for the TTL of the object. Set the time to live on the objects in CloudFront console (value is in seconds). Default is 24 hours or 86400 seconds. Maximum is 31536000 seconds (365 days)
- A 0 second TTL refreshes every time and revalidates every request
- If there is no cache header control configured, the default is 24 hours, each edge location checks for an updated version of your file whenever it receives a request more than 24 hours after the previous time it checked the origin for changes to the file
- Changing the file at the origin and if the TTL is still valid at the edge location, then the file does not get updated to the CloudFront servers since CloudFront thinks it already has current content. To get around this, use a different filename which forces CloudFront to pull down the updated file
- When you PUT an object at the origin it does not PUSH to CloudFront, CloudFront can only refresh when it pulls from the origin
- AWS will charge you to clear the cached objects over a certain number

- Cloudfront can be used to deliver an entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations.
- Supports S3, Elastic load balancing, Route 53, and Elastic compute
- Works with non-AWS origin traffic
- Web distribution supports HTTP and HTTPS only
- RTMP media Adobe Flash media is supported
- CloudFront is not only a read cache, writes are also support to /CloudFront, you can put objects on them (S3 uploads for example)
- When you first create a distribution, it will takes 5 to 10 minutes to create and 15 minutes to disable in the in the CloudFront console
- Whitelist or blacklist countries using geo-restrictions (whitelist of blacklist but not both)
- Technologies used by Amazon CloudFront:
- Caching: CloudFront caches the copies of content at locations closer to users accessing the data
- Using caching the content is delivered to users with lower latency. Loading on the main server is lowered
- Pricing:
    - On demand,
    - reserved capacity
    - price classes (turn off and on)
    - user controlled
- There are no transfer charges from origin to cloud front. You pay transfer from cloud front to end user location
- Regional edge cache is free of charge
- Receive data into the cloud, integrated with AWS WAF and Shield for security, edge location count now at 80 and growing
- Regional edge cache servers are edge locations that site between the regions and edge locations
- Persistent Connection support, CloudFront keeps persistent connections with the main server to download content quickly
- Optimization: CloudFront uses optimization techniques such as TCP initial congestion window etc. to deliver high network performance data transfer
- The Regional Edge Cache locations lie between your host webserver and the global edge locations

- When the popularity of an object content decreases, the global edge location may remove it from the cache. However, the Regional Edge location maintains a larger cache. So the object or content can remain longer at Regional Edge location. Using this technique, CloudFront does not have to go back to main webserver as often. When it does not find the data in the Global Edge location it queries the nearest Regional Edge location
- CloudFront supports the Lambda@ Edge utility to push processing to the edge and helps improve network latency for users. In Lambda@ Edge can remove requirements to provision or manage servers
- You can upload Node.js code to Lambda and create functions that will be triggered by CloudFront requests at the edge
- When a request for content is received at a CloudFront edge location, the Lambda code is executed locally at the Cloudfront edge location
- You can scale up processing operations by using Lambda in CloudFront without having to manage EC2 or ECS instances
- Events triggered by CloudFront:
    - Viewer Requests, HTTP/ HTTPS, to CloudFront, this event is triggered at the Edge Location closer to the end user then the data in the remote region
    - Viewer Response, when a CloudFront server is ready to respond to a request, this event is triggered
    - Origin Request, when the CloudFront server does not have the requested object in its cache, the request is forwarded to Origin server. This triggers an event
    - Origin Response, The CloudFront server at an Edge location receives the response from the Origin server, an event is triggered
    - Geo Targeting, CloudFront detects the country from where users request content. This information can be passed to the Origin server by CloudFront. The request is sent in a new HTTP header. Based on different countries you can generate different content for different versions of the same content. These different versions of localized content can be cached at different Edge Locations that are

closer to the end users of that country. This enable you to target users based on geographical locations

- Features of Amazon CloudFront:
  - Device Detection Protocol Detection
  - Geo Targeting
  - Cache Behavior
  - Cross Origin Resource Sharing
  - Multiple Origin Servers
  - HTTP Cookies
  - Query String Parameters
  - Custom SSL certificate
- CloudFront can source data from a S3 bucket or any HTTP server running as a EC2 instance in AWS or remotely in a private data center
- Enabling multiple origins and configuring multiple cache behaviors allows for the serving of static and dynamic content from the same distribution
- Origin Access Identifiers and signed URLs support serving private content from Amazon CloudFront
- Amazon CloudFront OAI (Origin Access Identifiers) is a special identity that can be used to restrict access to an Amazon S3 bucket only to an Amazon CloudFront distribution. Signed URLs, signed cookies, and IAM bucket policies can help to protect content served through Amazon CloudFront, but OAIs are the simplest way to ensure that only Amazon CloudFront has access to a bucket
- Supports multiple origins and cache behaviors
- The CloudFront Origin Access Identifier (OAI) is a special identity that can be used to restrict access to an S3 bucket only to an Amazon CloudFront distribution this forces everyone through CloudFront and prevents them from accessing the content directly from S3
- To prevent the S3 bucket From direct web access remove the public read attribute and use signed URLs with expiration dates on the S3 objects
- Re:Invent2016 CloudFront presentation https://www.youtube.com/watch?v=h2uN9VoAnz8
- Re:Invent2016 presentation CloudFront best practices presentation

https://www.youtube.com/watch?v=fgbJJ412qRE

- AWS training on CloudFront: https://www.slideshare.net/AmazonWebServices/edge-services-as-a-critical-aws-infrastructure-component-august-2017-aws-online-tech-talks
- Cloudfront can helpful when there is a planned or unplanned traffic surge:
    - Live broadcasts
    - Product launches or advertising promotions
    - Load testing
    - Viral internet events such as a twitter storm, a major web sites links to yours, social media campaigns
    - Web attacks such as DDoS, Bots and Scrapers

- Common errors in getting CloudFront to work are incorrectly configured caching policies or and unscaled origin
- Latency based routing continuously learns latency distance from billions of real user measurements around the world
- Security is enabled by default
    - HTTPS SSL/TLS termination close to viewers
    - High security ciphers
    - TCP fast open
    - Perfect Forward Security
    - Caches session tickets
    - Online Certificate Status Protocol (OCSP) Stapling

- See Aws.amazon.com/caching
- If the cache expiration time is not set or set to zero, CloudFront does not cache the objects at the edge location. The Edge location will then request the object from the origin for reach request

- It is recommended to set high TTLs (Time to Live) for intermediary caches
    - Max-age=3600. S-maxage=86400
    - Don’t forward the headers, query strings or any cookies
    - It is best to keep the default values in CloudFront

- Optimizing the cache settings:
  - Cache all the data you can, even dynamic content because there can still be a high volume of requests for dynamic content in a short period of time, then configure a low TTL value, like 5 seconds for rapidly changing content
  - A TTL value of 0 seconds means no-cache or do not store on CloudFront
  - Support for “if modified since” and “if none match” logic when the object in the cache has expired
  - You can configure more than one cache behaviors as required to meet your needs
  - Try not to use user-agent header, use as-mobile-view, is-tablet-view or is-smart-tv-view values instead, avoid forwarding all cookies, forward only the select cookies that you use to vary content

- Optimizing the end to end network path:
  - Use https2 between CloudFront and clients
  - Set the keep-alive timeout to the max value to keep TCP session open
- Scale at the origin:
  - Use auto scaling groups for web applications

- Lambda support
  - Lambda can be run at the CloudFront edge, reduces the processing load at the origin
  - Cloudfront triggers for Lambda@edge functions
  - User propertiew
  - Delete modify headers
  - A/B testing
  - Use to rewrite URLS
  - Convert to shorter URLS
  - Use Lambda to detect search engine bots and filter that traffic
  - Confirmation of valid sessions
  - Push EC2 processing functions out to the CloudFront

edge
    - Configure custom error pages at the edge
    - Deliver errors pages from S3
- Aws.amazon.com/CloudFront/events
- If you are publishing content over the Internet and need to restrict access to the documents, business data, streaming data, or content that is intended for authorized users, such as users who subscribe to your site. To securely serve this private content, then require that users access this private content by using special CloudFront-signed URLs or signed cookies
- Sample code to load into S3 bucket, set the correct name and image (add image also)

```
<html>
<head>CloudFront Test</head>
<body>
<p>Add text output in this section.</p>
<p>                                                    <img
src="http://w1d3hm53t3qove.cloudfront.net/mycoolpicture.jpg"     alt="
CloudFront_S3 Test image" />
</body>
</html>
```

Elastic Load Balancers: ELB

- Load balancers distribute connections between multiple servers in a pool and have one public facing IP address
- AWS provides two types of load balancers:
- Classic load balancer (CLB), uses application or network load information to route traffic. It is a simple way of load balancing to divide load among multiple EC2 instances
- Classic Load Balancing supports IPv4, IPv6, and dual stack (both IPv4 and IPv6), and -DNS names.
- Application Load Balancer (ALB), uses advanced application level information to route the traffic among multiple EC2 instances. ALB generally uses the URL name to make load balancing decisions, this is sometimes referred to as content switching
- Classic Load Balancer (CLB) features include:
    - Health checking of the servers verifies that they are active and can receive traffic. Based on the result of health check, the CLB will send traffic to the EC2 instance. If any instance does not respond to a health check, the CLB will stop sending user traffic to that server
    - Security groups are created for the load balancers in the Virtual Private Cloud (VPC). This is a security overlay for the load balancing application
    - High Availability (HA), distributes user traffic among EC2 instances in single or multiple Availability Zones for scalability and resiliency in the event of an availability zone going offline, the other AZ can scale to assume the load
    - Sticky Session support uses session cookies which enables traffic from a user to always be routed to the same EC2 instance so that user gets seamless experience
- Status monitoring for CLB collects statistics on request count, latency and other object metrics These metrics can be monitored in AWS CloudWatch

- Connection draining allows the load balancer to complete in-flight requests to instances that are de-registered or unhealthy
- Application Load Balancer (ALB) features:
    - Content-Based switching, the content in the request header is used to decide the routing of a request to a specific service
    - HTTP/ 2 support for the new version of the HTTP protocol. Allows for the sending of multiple requests over one established connection
    - Support for TLS and header compression
    - WebSockets support in EC2. Allows the server to exchange real-time messages with the end-users
    - Layer-7 Load Balancing for HTTP/ HTTPS application
    - Delete protection option in the AWS console to prevent it from getting deleted by mistake
    - Containerized Application Support load balances multiple containers across multiple ports on the same EC2 instance
- Both internet and internal facing load balancers are supported
- Instances monitored by ELB are reported as either InService, or OutofService based on the results of their health checks
- ELBs have their own DNS name, you are not given an actual IP address (AWS handles all resolution), this will change in the future and real IPs for the VIP will be supported
- You are not given an IP address for the public VIP, only the DNS name is published by AWS (currently)
- Configure the load balancer to accept incoming traffic by specifying one or more listeners, or as they are often referred to, VIP or Virtual IPs.
- Configure the Secure Sockets Layer (SSL) certificate so that clients connecting to the load balancer is not presented with a warning that the domain names does not match
- Create one SSL certificate with the Server Name Indication (SNI) value checked
- The SSL certificate must specify the name of the website in either the subject name or listed as a value in the SAN extension area of the certificate in order for connecting clients to not receive a

warning

- Create one SSL certificate with a Subject Alternative Name (SAN) value for each website name to avoid the user getting a warning that there is a discrepancy in the domain names when they connect
- Health checks can be a ping, a connection attempt, or a page request with the page request being the best option to validate the application layer is operational
- When an Elastic Compute Cloud (EC2) instance that is registered with an AWS load balancer that is configured to use connection draining is deregistered or unhealthy, the following will happen:
    - Keeps the connections open to that instance, and attempts to complete in-flight requests
    - Forcibly closes all existing connections to that instance after the timeout period of 300 seconds expires
    - Connection draining enables the load balancer to stop sending requests to a deregistered or unhealthy instance and attempt to complete in-flight requests until a connection draining timeout period is reached, which is 300 seconds by default
- ELB supports the Server Order Preference option for negotiating connections between a client and a load balancer
- This means that during the SSL connection negotiation process, the client and the load balancer present a list of ciphers and protocols that they each support and in the order of preference by each end
- By default, the first cipher on the client's list that matches any one of the load balancer's ciphers is selected for the SSL connection
- When the load balancer is configured to support Server Order Preference, then the load

balancer selects the first cipher in its list that is in the client's list of ciphers. This feature ensures the load balancer is the device that determines which cipher is used for SSL connection

- If Server Order Preference is not enabled then the order of ciphers presented by the client is used to negotiate connections between the client and the load balancer
- ELB does not and cannot load balance between regions

- ELB log files are delivered every 5 minutes to S3 storage
- The S3 bucket used for ELB logging must be in the same region as the load balancer.
- Access log files are compressed
- ELB logs are stored in a year/month/day directory structure in S3
- To load balance across availability zones, check the box for cross zone load balancing to enable this in the ELB configuration
- ELB access logs show detailed information on requests, each log contains timestamp, source (client) IP, latencies, request path, and the server responses
- Use access logs to analyze traffic patterns
- Access logs are enabled on the load balancer as a configuration item under load balancer attributes where you can enable the feature and specify the S3 bucket to output to
- Instances automatically launched by an ELB will not be terminated when the ELB is deleted.
- http://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/h elastic-load-balancing-works.html

Custom VPCs and Elastic Load Balancers

- Console, EC2 area
- Go to load balancers
- High availability requires two public subnets each in a different availability zone

Elastic Beanstalk

- Elastic Beanstalk is an automated deployment and scaling service
- For basic and simple deployments, this is the AWS idiot app.
- AWS service that looks at the code you want to deploy and provisions AWS network to accommodate this
- This service is intended to for ease of you for developers to deploy AWS resources
- An AWS service offering for deploying and managing applications without being concerned about the infrastructure that runs those applications
- Elastic Beanstalk is a fast and simple service to get an application up and running on AWS
- Developers upload their application code, and Elastic Beanstalk deploys and configures al of the needed services such as load balancing, Auto Scaling, and monitoring
- Enables the developers to focus on writing code instead of having to manage and configure servers, databases, load balancers, firewalls, and networks
- JVM settings (such as min, max, heap size) can be modified using elastic beanstalk
- Accepts Java, PHP, Node JS, Python, Ruby, Go, or Docker code
- Deploys Apache, Nginx, passenger and IIS servers
- Allows updates to the environment and platform
- Supports time based scaling
- JVM settings can be modified
- Can automatically handle Load balancing, health monitoring, auto scaling, application platform management and code deployment
- All application files and log files are stored on S3
- You have access to the underlying infrastructure and can make modification to EC2 instances

Lambda

- Lambda is an AWS compute service where you can upload your code and create a lambda function
- Lambda takes care of provisioning and managing the servers that you use to run the code
- You don't have to worry about operating systems, patching, scaling, etc.
- Serverless computing
- No operating system for you to manage, this is a container micro service
- Introduced by AWS in 2016
- Alexa uses Lambda as its compute engine
- AWS handles all underlying server technology
- User uploads the code and configures a trigger event to fire off the lambda instance that runs the code
- Use case: A file gets uploaded to an S3 bucket that event triggers a lambda function to act on the uploaded file
- Pure code delivered in a container
- Lambda = data centers, hardware, assembly code/protocols, high level languages, application layer APIs in a service based on containers
- Here are some examples of using Lambda:
    - An event driven compute service where AWS lambda runs your code in response to events.
    - These events could be changes to data in an S3 bucket or a Dynamo DB table with many other services being added over time
    - As a compute service to run your code in response to HTTP requests using amazon API gateway or API calls made using AWS SDKs, no servers, all lambda micro services
- No worries about peak traffic as lambda is on-demand and gets called as needed, has almost unlimited scalability
- Lambda pricing is very cheap; the first 1 million requests are free

and then $0.20 per 1 million requests thereafter

- You are charged for the duration that the instance is run. Calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. You are charged $00.00001667 for every GB-Second used
- Lambda supports the following languages: Node.js, Java, Python, C# (more are being added)
- No severs
- Continuously scales
- Lambda is a very cost effect compute service
- Default execution time is 3 seconds
- Minimum execution time is 1 second
- Maximum execution time is 300 seconds (5 minutes)
- Take the Alexa course on acloud.guru to learn Lambda, no programming skills

Lightsail

- Complete cloud service
- Easily creates Virtual Private Servers
- Out of the box cloud
- Preconfigured templates
- Auto deployment that then you can customize your VM
- For people that do not know how to use AWS
- 10-2017 added window server instances
- Includes
    - Virtual machine
    - SSD-based storage
    - data transfer
    - DNS management
    - Static IP
    - Low predictable price.

S3: Simple Storage Services

- Simple Storage Services = S3
- S3 is a very large part of the CSAA exam, you must know this material in depth before sitting for the exam
- Think of S3 as a virtual disk in the Cloud
- Stores data as objects mainly files
- File are stored in buckets and called objects
- Web, CLI, SDK and API based access
- Amazon S3 is Object Based storage, it is not block or file based
- Objects are stored in buckets, and objects contain both data and metadata
- Amazon S3 cannot be mounted to an Amazon EC2 instance like a file system
- S3 is object storage service, which differs from block and file cloud storage. Each object is stored as a file with its metadata included and given an ID number. Applications use this ID number to access an object
- Unlike file and block cloud storage, a developer can access an object via a REST API
- S3 should not serve as primary database storage
- Amazon S3 can store unlimited amounts of data

- S3 is the AWS primary data storage offering that features secure, durable, highly-scalable object storage
- S3 features an easy to use browser interface to store and retrieve any amount of data from anywhere on the internet

- This is Object based storage and not block so it is not used for Operating system volumes (use EBS instead)
- Files size 1 byte to 5 terabytes
- Unlimited storage
- Flat file system
- S3 has a universal name space inside of AWS so each bucket name must be globally unique

- S3 storage consistency depends on the operation performed. It takes time to propagate the objects between the S3 backup locations
  - Read after write consistency for PUTS of new objects.
  - Immediate consistency for new files.
  - Eventual consistency for overwrite PUTS and DELETE
- S3 is a key value store, object based
  - Key = name of the object
  - Value = this is the data and is made up of a sequence of bytes
  - Version ID = AWS assigned value used for versioning
  - Meta data = Data about data you are storing create date for example
- Tired storage is called lifecycle management
- Successful S3 writes return a HTTP 200 code
- Lifecycle management moves between storage tiers
- There are multiple encryption options
- Access Control Lists and Bucket policies are used to secure the S3 data
- Cross region replication of S3 buckets requires versioning be enabled
- Default maximum of 100 buckets per account (That is the default number of S3 buckets) more buckets can be obtained per account on request
- Bucket naming requirements:
  - Must be globally unique
  - Universal name space
  - From 3 to 30 characters long
  - Bucket name labels are separated with a “.” Period
  - Names cannot be formatted as an IP address
  - Can contain lower case letters, numbers, and hyphens but no spaces
  - The label must start and end with a lower case letter or a number (no upper case allowed)
  - Your bucket name *always* comes first, "s3-website" followed by the Region *always* comes next. Example: tipofthehat1.s3-website-us-west-2.amazonaws.com

  - Bucket naming convention: <bucektname>.s3.amazonaws.com
  - Cannot start or end a bucket name with a hyphen don't follow or precede a period with a hyphen
  - http://<bucket>.s3-aws-region.amazonaws.com
  - http://s3-aws-region.amazonaws.com/<bucket>
- S3 can host static web content and supports website redirects (no need for a EC2 instance running a web server)
- By default, all S3 policies are private, only the resource owner, the one when created it, can access the resource
- The owner grants access to others by creating a policy
- Two types of polices
  - Resource policies applied directly to a resource (object of bucket)
  - User policies that are applied to IAM users in your account
- Use resource or user policies individually or together
- All policies are evaluated at the same time
- If access is granted by one policy and denied by another, denied will always win, deny always trumps an allow
- Buckets are partitioned by the prefix, so if they all start with the same text, they get grouped together (this may effect performance as they all get written to the same storage array), random numbers at the beginning of the filename offers better performance but can make them harder to manage since the objects are not grouped by name

- S3 = Object based. Objects consist of the following:
  - Key = name of the object
  - Value = the data
  - Version ID (for versioning)
  - Metadata (tags)
  - Sub resources
  - Access Control Lists (ACLs)
- Can be configured to include confidentiality, integrity, availability and accountability of the data
- Not for application or database storage or OS volumes, this is not a

file systems, it is object based

- Uses the REST API
- Used for backup, archive, content storage, file distributions, static website hosting, disaster recovery and more
- Security includes pre-signed URLs, ACLs and bucket policies
- Bucket policies can grant access from other accounts outside of your own or outside users, access rights are very flexible and feature rich
- Protection against accidental deletion, use the versioning and multifactor authentication options
- To encrypt the data before is it sent (data in flight) use client-side encryption with customer-managed keys
- Before cross region replication can be configured, you must enable versioning and create an AWS Identity and IAM policy to allow S3 to replicate objects on your behalf
- Bucket access can be restricted by IP address, AWS account and objects with a specific prefix
- Increase S3 performance for read heavy operations use some randomness in the key space by including a hash prefix to key names, this allows for random writes and less write latency
- S3 is an eventual consistency storage system, you can get stale data with a GET or LIST after a DELETE or a GET after overwrite PUT(PUT to an existing key)
- A successful PUT returns HTTP 200 result code and a D5 checksum to reply that the operation was a success
- http://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOS
- S3 supports static web hosting. Configure the bucket for static hosting and specify an index and error HTML documents, create a bucket with the same name as the website and make the objects world readable
- S3 static website:
    - The two general forms of an Amazon S3 website endpoint are:
    - bucket-name.s3-website-region.amazonaws.com
    - bucket-name.s3-website.region.amazonaws.com
- S3 data is automatically replicated inside of the region that the bucket is in

- Server access logs stores records created of who accesses the data including the IP address
- This is Object based storage. This is **not** block based storage!
- Some of the important features of Amazon S3 are as follows:
  - Amazon S3 provides unlimited storage for files
  - File size in Amazon S3 can vary from 0 Bytes to 5 Terabytes
  - 100 buckets allowed per account by default
- In Amazon S3, names of buckets have to be unique globally
- S3 objects are a key: value store. Key=name of the object and the value= the data being stored (0-50TB) and a Version ID= string of data assigned to an object if versioning is enabled
- Object key names use UTF-8 encoding and must not be longer than 1024 bytes
- Objects stored in S3 can be tagged to categorize objects using a key/value pair
  - Project: Architect _Associate (Key)
  - Classification=confidential (value)
- Object tags enable very detailed access control, lifecycle management, filtering for CloudWatch metrics and CloudTrail logs
- Tagging allows you to group your objects
- Object tagging features:
  - Keys can be 128 Unicode characters in length
  - Keys and Values are case sensitive
  - Each tag must have a unique key
  - Up to 10 tags per object
  - Values can me 256 Unicode characters in length
- Key point: Bucket+Key+VersionID uniquely identify an object in S3
- Metadata is also attached to an object as a Name-Value pair which are used to store information about the object
- Sub-resources can also be assigned to an object such as an Access Control List
- S3 has a flat file structure, there are no directories or subdirectories
- S3 can give the appearance of directories by using prefixes /images/thumbnails/public can act as a prefix and me.jpg the file but in reality it is a file name on not a directory structure

- Both buckets and objects are classed as resources which is an AWSD entity you can manage
- Upload as many objects as you like into buckets, there is no real limit
- The total volume of data and number of objects you can store are unlimited.
- Individual Amazon S3 objects can range in size from a minimum of 0 bytes to a maximum of 5 terabytes
- The largest object that can be uploaded in a single PUT is 5 gigabytes
- For objects larger than 100 megabytes, customers should consider using the Multipart Upload capability
- By default, you can create up to 100 buckets per account, this is a soft limit that you can request to be increased
- Buckets are created in regions
- Objects stored in a region will never leave that region unless you explicitly transfer them out
- Can be accessed using standards-based REST and SOAP API interfaces
- S3 universal namespace, it is unique regardless of the region, global like a domain name, no two buckets can have the same name anywhere in AWS
- S3 objects are referred to with an Amazon Resource name (ARN)
- Buckets can be accessed via Virtual or Path style URL"
    - Virtual: (region is optional)
    - http://bucket.s3.amazonaws.com
    - http://bucket.s3-aws-region.amazonaws.com
    - Path: (region must be specified and the bucket name is at the end)
    - http://s3-aws-region.amazonaws.com/bucket
    - Format link to bucket: https:s3-eu-west-1.amazonaws.com/tipofthehat
    - All S3 objects have a URL https://s3-us-west2.amazonaws.com/tipofthehat1234 for example
    - In a URL, the bucket name precedes the string "s3.amazonaws.com/," and the object key is everything after that. There is no folder structure in Amazon S3

    - https://bucket1.tipofthehat.com.s3.amazonaws.com/folder = The object “folderx/myfile.doc” is stored in the bucket “bucket1.tipofthehat.com.”
    - Remember that there is no directory structure even though /folder/myfile.doc looks like it, it is actually the complete name of the object
- Amazon S3 uses a REST (Representational State Transfer) Application Program Interface (API)
- Amazon S3 supports durability at the scale of 99.999999999% of time. This is 9 nines after decimal. Often called “The eleven nines” Memory aide: Durability = Damm!
- Availability is 4 9’s or 99.99%
- S3 supports Read after Write consistency when creating a new object by issuing a PUT. It means as soon as you write the new object, you can access it anywhere in the region
- Read after write consistency means you can only read the data after its been successfully written to all facilities and returned a success response, this is another S3 option in addition to eventual consistency
- S3 has eventual consistency when you overwrite an existing object with a PUT or perform a DELETE operation
- Eventual Consistency means that the effect of overwrite (puts) will not be immediate and will occur after a short period of time. For deletion of an object, S3 supports Eventual Consistency after DELETE
- Eventual consistency allows access to the object before it has been replicated to all of the facilities which means old data may be returned, deleted keys may still show up in the bucket
- Eventual consistency has much higher throughput and lower latency
- If two requests are made at roughly the same time, the one with the latest timestamp (most recent) wins
- The data is consistent after a GET is performed on a new PUT (new object stored)
- It is not immediately consistent when a GET or LIST after a DELETE
- It is not immediately consistent when a DELETE is done after a

PUT of a new object; Amazon S3 provides read-after-write consistency for PUTs to new objects (new key), but eventual consistency for GETs and DELETEs of existing objects (existing key)

- S3 uses key-value pairs.
- Can send out notification triggers based on PUT, POST, Copy, Multipart upload, or Delete
- Trigger workflows with Amazon SNS, SQS, and AWS Lambda functions
- Video intro to S3: https://www.youtube.com/watch?v=77lMCiiMilo
- Getting Started with Amazon Simple Storage Service: http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS
- Frequently Asked Questions: https://aws.amazon.com/s3/faqs/
- Manage Access to Your S3 Resources: http://docs.aws.amazon.com/AmazonS3/latest/dev/intro-managing-access-s3-resources.html
- Log Requests to Your Bucket: http://docs.aws.amazon.com/AmazonS3/latest/UG/ManagingBucket
- Version Control Your Objects: http://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html
- Cloudberry labs offers a windows explorer frontend to S3 which features a directory/subdirectory view https://www.cloudberrylab.com/explorer/amazon-s3.aspx
- CLI document: http://docs.aws.amazon.com/cli/latest/reference/s3/
- Features that can be used to restrict access to Amazon Simple Storage Service data:
  - S3 Access control lists on a bucket or object
  - pre-signed URL for an object
  - S3 bucket policy
- To protect from inadvertent S3 bucket deletion
  - Enable MFA (multi-factor authentication)
  - Enable versioning on the bucket
- Versioning protects data against inadvertent or intentional deletion by storing all versions of the object, and MFA Delete requires a one-time code from a Multi-Factor Authentication (MFA) device to delete objects

- S3 data is automatically replicated within a region and can be configured for cross region replication
- Replication to other regions and versioning is optional and must be explicitly configured
- Amazon S3 server access logs store a record of what requestor accessed the objects in your bucket, including the requesting IP address
- Server access logs provide a record of any access to an object in Amazon S3
- Server access logs must be enabled on the bucket and are not by default
- Cross-region replication can help lower latency at remote locations since the data is closer to the requesters
- Cross-region replication can satisfy compliance requirements on distance between data stores (separation)
- S3 is designed for eleven nines durability for objects in a single region, this means a second region does not increase durability
- Cross-region replication does not protect against accidental deletion
- It is a requirement to enable versioning before you can enable cross-region replication
- S3 must have sufficient IAM permissions to perform the replication
- S3 bucket policies can specify request IP range, AWS account, and a prefix for objects that can be accessed
- S3 does not support FTP transfers
- Pre-signed URLs allow you to grant time-limited permission to download objects from a bucket
- S3 storage service gives a subscriber access to the same systems that Amazon uses to run its own websites
- S3 enables a customer to upload, store and download practically any file or object that is up to five Terabytes (5 TB) in size
- S3 is a RESTful web service
    - Interact with it over web based protocols such as HTTP and HTTPS
    - Requests are made using the REST API
    - GET=download/read
    - PUT=upload/write
    - DELETE=Delete

- S3 charges for:
  - Storage
  - Requests
  - Data Transfer
  - Transfer acceleration
  - Management functions
  - metrics beyond free tier
  - Storage class analyses
  - S3 inventory
  - Object tagging
- Data transfer into and out of S3 can use SSL encrypted endpoints (HTTPS)
- S3 server side encryption (SSE) allows S3 to manage the encryption keys for you
- S3 SSE encrypts data on upload using an additional request header when writing an object and decryption happens automatically when the data is retrieved
- SSE uses 256 bit EAS encryption
- SSE encrypts the encryption key
- Bucket policies can be defined to state that only encrypted data can be stored
- The SSE keys are stored remotely and not in the bucket

S3 Storage tiers:

- Lifecycle transitions can be set by lifecycle policies that will migrate objects to different tiers
- Lifecycle policies restriction to consider when applying polices to migrate objects form one storage class to another are minimum duration and minimum object size
- The tiers are S3 standard, S3 infrequent access, reduced redundancy storage and glacier

S3 Standard:

- The standard tier supports durable storage of files that become immediately available.
- This is used for frequently used files
- SSL support
- Availability = 99.99%
- Durability: 99.999999999% (11 nine’s)
- Designed to sustain the loss of two AZ datacenters in a region
- stored redundantly across multiple devices in multiple facilities
- There is no retrieval fee for the data
- No minimum object size
- No minimum storage duration

S3 standard Infrequent Access (IA):

- The RRS-IA tier provides durable storage that is immediately available
- Used for files that are infrequently accessed, backup and archive, disaster recovery, file sync and sharing and data that needs to be retained for a long time but not frequently accessed
- Lower cost the S3 standard
- Eleven nines durability, high throughput with low latency
- Low per GB storage pricing with a per GB retrieval fee
- Can lose two facilities (AZs) in a Region and survive
- Availability = 99.9%
- Durability: 99.999999999%
- Availability SLA: 99%
- Minimum storage duration: 30 days
- Min object size: 128Kb (you can store smaller but with be charged at the minimum of 128K)
- Retrieval fee per GB retrieved
- Used in Lifecycle management or when you want to save money over S3 standard and the data is not frequently accessed
- The minimum object size is 128KB. Objects smaller than 128KB are charged at the minimum 128K rate
- Minimum storage duration 30 days (soft limit) will be charged for 30 days if you use less
- Migrating the data to Amazon S3 Standard-IA after 30 days using a lifecycle policy is common
- Retrieval fee pricing structured for longer term storage with less access
- SSL support

S3 standard Infrequent Access Single Availability Zone (IA):

- Similar to standard Infrequent Access S3 but the data in stored in only one AZ (as compared to three)
- Costs 20% less
- Retains 11 9’s durability
- Availability is reduced to 99.5% over a year.
- Used for non-critical data such as a secondary backup

S3 Reduced Redundancy Storage (RRS):

- RRS provides you the option to store data in S3 at lower levels of redundancy.
- Data is copied to multiple locations (AZs) but not on as many locations as standard S3
- Use for non-critical data
- Can lose one facility (AZ) in a Region and survive
- RRS trades durability for lower cost
- Amazon S3 RRS should only be used for easily replicated data or data that you can afford to lose
- It is not a good idea to use RRS to store critical data
- Single facility fault tolerance
- 99.99% durability
- 99.99% availability

S3 version control

- When enabled all existing objects are retained and a new version is created with a new version ID
- Must be enabled for cross region replication
- Protection from unintended user deletes or application failures
- New version of the object with every upload
- Retrieval of deleted objects is easy and you can roll back to previous versions of the object
- S3 provides DELETE API to delete an object. If the bucket in which the object exists is version controlled, then you are able to specify the version of the object that you want to delete.
- The other versions of the Object still exist within the bucket
- If you do not specify the version, and just pass the key name, Amazon S3 will delete the object and return the version ID
- Then the object will not appear on the bucket
- In the case where the bucket is Multi-factor authentication (MFA) enabled, then the DELETE request will fail if we do not specify the current MFA token
- Youi cannot disable versioning on a version-enabled bucket in Amazon S3 once it has been configured, however, you can suspend it which has the same effect
- Suspending Versioning:
    - You can suspend the versioning on a bucket in S3. Once suspending versioning, S3 will stop creating new versions of the object. It stores the object with null version ID
    - On overwriting an existing object, it replaces the object with null version ID
    - Any existing versions of the object still remain in the bucket. But there will be no more new versions of the same object except for the null version ID object

S3 cross region replication

- Use Cross Region Replication of S3 to make copies of an object across buckets in different AWS Regions
- Copying occurs automatically and is asynchronous
- Configure cross region replication in the source S3 bucket to enable
- CRR creates exact replicas of the objects from the source to destination buckets in different regions
- Use cases of Cross Region Replication are:
  - Geopolitical separation compliance: When laws/ regulatory requirements that data be stored at geographical separated locations. This can be accomplished using AWS Regions that are spread across the world
  - Failover: To reduce the probability of system failure due to complete blackout in a region. Use Cross-Region Replication in such a scenario to maintain operations by having data available in other regions.
  - Reduce network latency: When the data is being accessed from multiple geographies, you can replicate objects in the geographical Regions that are closer to end customer.
  - Security: remote replicas managed by separate AWS accounts
- Deletes and lifecycle actions are not replicated
- 1:1 replication between any 2 regions
- Enable versioning on the source bucket to perform Cross Region Replication
- Versioning must be enabled on both the source and destination buckets
- Create an IAM role to grant S3 permission to copy objects on users behalf
- Cross-region replication is a bucket-level feature that enables automatic, asynchronous copying of objects across buckets in different AWS regions.
- To activate add a replication configuration to the source bucket.

- In the configuration, you provide information such as the destination bucket where you want objects replicated to.

S3 Lifecycle Management

- The automated migration of objects from one S3 storage class to another based on the age of the data
- S3 Lifecycle management provides the ability to define the lifecycle of the object with a predefined policy and is used to reduce your storages costs with AWS by moving objects to lower cost storage options when they are infrequently accessed
- Set lifecycle transition policy to automatically migrate S3 objects to Standard - Infrequent Access (Standard - IA) and/or Amazon Glacier based on the age of the data

- Allows for the creation of S3 transition actions, which you define, what allows objects to transition from one S3 storage class to another
- An example of lifestyle management in S3 would be when storage objects are created in S3 Standard -> 30 days later move the data to Standard Infrequent access -> archive objects to Glacier for one year (after creation) and then delete
- Migrating the data to Amazon S3 Standard-IA after 30 days using a lifecycle policy is common
- The lifecycle section of the S3 bucket (add rule) in the AWS console
- Select what you want to export, choose action to perform on the rule creation
- Set up lifecycle polices in the AWS management console
- The use case is to store the data in the lowest cost S3 service that meets your requirements.
- There are two primary types of Object Lifecycle Management actions in Amazon S3
- Transition Actions: defines the state when an Object transitions from one storage class to another storage class. For example, a new object may transition to STANDARD_IA (infrequent access) class after 60 days of creation. And it can transition to GLACIER after 180 days of creation

- S3 lifecycle configuration rules are a mechanism for controlling objects that have a well-defined lifecycle by moving them between storage classes or deleting them at specific time intervals
- Expiration Actions: Defines what happens when an Object expires. You can configure S3 to delete an object on expiration for example
- If you enable replication on a bucket that already has objects in it, the original objects DO NOT get migrated, only new objects since replication was enabled are replicated
- Automate lifecycle polices for hands off administration
- Set transitions based on the objects age and then delete the object after a time you specify
- Policy actions can be combined together for automation flexibility
- Set policies by bucket, prefix or tags
- Set policies for current versions of the object or non-current versions if you have versioning enabled
- http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

S3 security and encryption

- When creating a new S3 bucket, they are PRIVATE by default
- To make a bucket publically accessible, use access control to make the configuration change
- Bucket policies are global
- Access control list allow you to create very Granular profiles
- Access control for buckets
    - Bucket policies
    - ACLs

- S3 buckets can be configured to create access logs which log all requests to the S3 bucket
- Data in transit is protected using SSL/TLS
- At rest: Server side: Data is encrypted as it is stored on a cloud storage service, also referred to as encryption at rest.
- AWS Key Management Service, Managed Keys – SSE-KMS Audit trail
- Self-manage keys SSE-C server side encryption with customer provided keys
- Client side encryption for transmitting into S3
- Encrypt on the client side and upload to S3

- Encryption – 4 methods
    - In transit – information to/from bucket
        - Uses SSL/TLS
    - S3 data at rest:
        - Server Side Encryption (SSE)
            - S3 Managed keys – SSE-S3 (Server Side Encryption S3) each object encrypted with a unique key, uses strong multifactor encryption, AWS encrypts the key itself with a master key that gets rotated. AES-256 Amazon handles all of the keys for

you. Click on the object and select "encrypt"

- AWS Key Management Service KMS, Managed Keys – SSE-KMS AWS Key Management Service, Managed Keys. Similar to SSE-S3 but adds an envelope key, which is a key that protects your data's encryption key. It also adds an audit trail of when your keys were used and who used them. The audit/logging show you who is decrypting what and when. There is an option to create encryption keys yourself or use the default keys provided to you in that region. There are additional charges for all of this.
  - Provides usage audit trail
- SSE w/ Customer Provided Keys – **SSE-C** You manage the encryption keys and Amazon manages the encryption and decryption process.

- Client Side Encryption – the customer encrypts data prior to uploading to the S3 bucket, sometimes called encryption in flight, or in transit
- Client side encryption is when the data is encrypted by the client before it is transmitted to the remote service and then unencrypted at the S3 bucket upon arrival

- S3 provides the ability to encrypt data at rest and in transit
- All access to S3 resources can be optionally logged to provide audit trails
- S3 is a secure storage service but you must manage the security options available to you.
- Some of the main security mechanisms available in Amazon S3 are as follows:
  - Access: When we create a bucket or an object, only the

owner get the access to the bucket and objects
    - Authentication: Support for user authentication to control who has access to a specific object or bucket
    - Access Control List: Create Access Control Lists (ACL) to define fine grain permissions to users and groups
    - HTTPS: S3 supports the HTTPS protocol for secure upload and download of data from cloud
    - Encryption: S3 supports Server Side Encryption (SSE) to encrypt S3 data
- Protect your data from accidental loss:
    - Enable versioning
    - Enable MFA (Multifactor Authentication) Delete
    - Use Access Control Lists (ACLs)
    - Use Amazon S3 bucket policies
    - Use AWS Identity and Access Management (IAM) policies
- To ensure the maximum performance for high-rate GET, PUT, and DELETE requests on Amazon Simple Storage add a random prefix to your object names
- Four ways to encrypt S3 data at rest:
    - Server Side Encryption (SSE) with AWS-managed keys
    - SSE with AWS Key Management Service (AWS KMS)-managed keys
    - SSE with customer-provided keys
    - client-side encryption
- Initial PUT requests has read after write consistency, all other requests, including overwrite PUT requests, are eventual consistency
- Mechanisms to control access to objects in an Amazon Simple Storage Service:
    - Access Control Lists (ACLs)
    - AWS Identity and Access Management (IAM) policies
    - S3 bucket policies
- 5 steps to create a S3 bucket policy: http://searchaws.techtarget.com/tip/AWS-Management-Console-Five-steps-to-create-an-S3-bucket-policy
- Versioning, if enabled, provides that ability to retrieve every

version of every object ever stored in S3, even if the object has been deleted

- Only the S3 bucket and object owners have access to the resources they create
- Permissions to objects and buckets are granted using policies

S3 versioning

- Keeps multiple variants of an object in the same bucket
- Enable versioning to preserve, retrieve and restore every version of every object stored in a S3 bucket
- Each objects has the same key but use different version IDs
- When you version enable a bucket, it can never revert to an "un-versioned" state, but you can suspend versioning on that bucket which accomplishes the same objective
- Stores all versions of an object (including all writes and deletes)
- Versioning is often used for backup
- Cannot disable versioning once enabled, only suspend
- Integrates with lifecycle rules
- Can use the multi-factor delete (MFA) capability, you can't delete an object without first entering the MFA token
- Cross Region Replication requires versioning – only applies to files manipulated *after* CRR is turned on
- Versioning protects from accidental overwrites and deletes with no performance penalty
- Generates a new version with every object change of a stored object
- Enables easily retrieval of deleted objects or roll back to previous versions
- Three bucket states
    - Un-versioned (default)
    - Versioning-enabled
    - Versioning-suspended
- Can take up a LOT of space on files that change a lot (because it stores each changed version)
- Versioning for backups of data and backups of previous version of an object
- Versioning, even if object is deleted it remains in S3
- Versioning can be used as a backup tool and retaining older versions of data
- When versioning is enabled the existing files to not get replicated,

only new files

S3 Billing

- Storage volume consumed
- Requests I/O
- Storage management pricing
- Data transfer pricing In and Out, however, as a general rule, AWS does NOT charge for data INTO S3 but DOES charge for data OUT of S3
- Transfer acceleration (CloudFront edge to S3 bucket)
- Uses the pay for what you use consumption model
- No minimum fee for S3
- Prices are based on the Region of the bucket and can vary per region
- You can estimate the monthly bill using the AWS simple monthly calculator
- Storage, request and data transfer pricing: data transferred out of S3

S3 Transfer Acceleration / Multipart Upload

- Increase aggregated throughput by sending PUTs in parallel data streams
- Better resiliency as it overcomes network errors with less restarts and smaller retransmission size
- No firewall or client side changes
- The greater the distance and larger the files the more benefits realized
- All global edge locations support multipart upload
- Enable CloudFront and cache S3 objects at the edge for low latency data transfers to the end users
- Speed comparison utility: www.S3speedtest.com

- S3 transfer acceleration utilizes the CloudFront Edge Network to accelerate uploads to S3 buckets
- Uses a specific URL to upload to an edge node close to where you are and instead of directly to the S3 bucket, the edge location then uploads the objects to the bucket using the AWS internal network
- Uses a distinct URL: to upload to Tipofhtehat.s3-accelearate.amazonaws.com
- In the AWS S3 console under transfer acceleration tab

- Chargeable feature
- Offers upload options for more resilient increase throughput of file uploads in addition to handling larger files than single part upload
- Fast recovery from network issues
- You can pause and resume object uploads
- Includes the ability to start uploads before the final object size is known
- Multipart upload is required for file sizes of 100 megabytes and higher
- Multipart upload is the ability to upload a large files in multiple

parts or streams

- Each part is uploaded in parallel and is independent of the others, order of the parts is not relevant, and they are all reassembled at the AWS end
- Uploading the data in parallel increased performance and the time it takes to upload the data
- After the upload is completed, one single object or file from which the parts are created, in other words, they streams are reassembled into the single larger file

- One the file upload is complete, the application makes an API call to assemble the file in S3 and complete the upload process

Glacier

- Archive files off of S3 for long term storage, data preer4vation and replacement for magnetic tape backups
- Offline storage for regulations that require keeping files many years to meet compliance
- Archives takes 3 – 4 hours to retrieve
- Used for long term backup storage that almost never needs to be retrieved
- Amazon Glacier is an extremely low cost cloud based storage service provided by Amazon
- Use Glacier for storing data archives for months, years or even decades
- Can also be used for long term immutable storage based on regulatory and archiving requirements
- Glacier can be used as a standalone service or as an S3 storage class
- Behind the scenes, S3 acts as a front end to the Glacier service and allows glacier to use the S3 feature sets
- Glacier vaults can be locked
- It provides Vault Lock support for this purpose. In this option, we write once but can read many times the same data
- One use case is for storing certificates that can be issued only once and only the original person keeps the main copy
- To restore a file stored in Glacier, there are two options: S3 API or the AWS console
- AWS Glacier stores data in archives, which are contained in vaults
- Archives are identified by system-created archive IDs, not key names
- Less than $0.01 GB/Month (depends on the region)
- Durability : 99.999999999%
- Availability: 99.99%
- Retrieved data from Glacier is stored in S3 RRS (reduced redundancy)
- Data stored in glacier is automatically encrypted using AES-256

- Maximum archive size is 40TB
- Minimum archive size is 1 byte
- Unlimited number of archives
- Uploading data to Glacier is an synchronous operation
- Supports the multipart upload API
- You will get a returned unique Archive ID when the data has been durably stored in Glacier
- Each archive is assigned a unique archive ID at the time of creation
- The content of each archive in glacier is immutable or unchanging / unable to be changed
- Glacier supports the following three operations: Upload, Download and Delete
- When data is uploaded, glacier copies data synchronously to multiple facilities before returning a success message to the user
- CLI command to create a glacier vault: AWS glacier create-vault
- If data was stored in glacier using a S3 lifecycle policy, then you must use the S3 API to list or retrieve the objects as glacier is acting as a storage class for S3

- Glacier, very cheap archival only data store
- Extremely low cost storage for data archives. As little as $.01 per gigabyte per month
- Optimized for data that is infrequently accessed and for which retrieval times of 3-5 hours are acceptable
- Charged for storage amount used, tiered so the more storage used, the less it costs.
- Charged for the number of request and data transfer charges
- Can look in the console of the files in glacier
- Fill out form for AWS to retrieve file
- Glacier restores go to S3 Reduced Redundancy storage (RRS)
- AWS will send e-mail with a link to get the recovered file that is placed in S3
- Data is encrypted on the server side, AWS handles the key management using AES-256 encryption
- Customers can also upload the data already encrypted

EBS: Elastic Block Store

- Amazon AWS EBS allows you to create block based storage volumes and attach them to Amazon EC2 instances
- Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. Think of EBS as a hard disk
- Persistent storage that is independent of any EC2 instance, you can detach a EBS volume from one instance and attach it to another
- Raw unformatted block storage that needs to be formatted to work with any operating system
- Annual failure rate for EBS is between 0.1 and1%
- EBS volumes can range in size from 1GB to 16TB
- Easy to create, attach, back up, restore and delete using the AWS console, command line or APIs
- Can be used as a bootable root volume
- Once attached, you can create a file system on top of these and use them as you would and block storage device
- You pay for what you provision, if you create a 16G EBS volume, that is what you pay for
- EBS volumes are placed in a specific availability zone and are automatically replicated to protect you from the failure of a single component
- EBS is a hard disk in the cloud and you can mount multiple EBS volumes on an EC2 instance
- Very fast I/O (faster than S3)
- EBS is under EC2 in the console since they were created to support EC2 as attached block storage volumes
- You cannot access EBS from the internet (only if attached to a virtual server), by default EBS is not public facing

- If an EBS volume is an additional storage partition (not necessarily the root volume) you can detach EBS volume without stopping the instance, however it may take some time to complete
- Amazon EBS provides following two main types of Volume:
- Solid State Drive (SSD): EBS backed by a Solid State Drive. It is

suitable for transactional work in which there are frequent reads and writes. It's more expensive than the HDD based volume

- Hard Disk Drive (HDD): EBS backed by Hard Disk Drive. It can be used for large streaming workload in which throughput it more important than transactional work. HDD is a cheaper option compared with SSD Volume
- Persistence: The main difference between Instance Store and EBS is that in Instance Store data is not persisted for long-term use. If the Instance terminates or fails, we can lose Instance Store data, this is often called ephemeral storage. Any data stored in EBS is persisted for longer duration. Even if an instance fails, you can use the data stored in EBS to connect it to another EC2 instance
- For EBS persistence to be enabled you must configure this or the EBS instance will delete at termination, this is done when the EBS volume is created
- Some EC2 instances types provide the option of using a directly attached block-device storage. This kind of storage is known as Instance Store that is ephemeral and the data is deleted when the instance is stopped
- In other Amazon EC2 instances, we have to attach an Elastic Block Store (EBS)
- If the Instance terminates or fails, instance Store data is lost. Any data stored in EBS persists
- Detach the EBS volume from an EC2 instance, the EBS instance lives (stored in S3) and then can be re-attached to another EC2 instance
- EBS root volume is deleted if the instance terminates
- EBS volumes persist if the EC2 instance is stopped or restarted
- Boot time is usually under one minute with an EBS volume
- For large workload such as databases use Provisioned IOPS SSD that has faster I/O performance than regular EBS SSD or magnetic drives
- Created under the EC2 section of the AWS console, on the left pane select volumes
- For cold and infrequently accessed data use the EBS magnetic volumes
- For dev/test and small databases it is most common to use a

General purpose SSD EBS volume type

- An EBS optimized instance has additional dedicated capacity for EBS I/O
- Amazon EBS volumes can be encrypted transparently to workloads on the attached instance
- EBS Volumes cannot be attached to an EC2 instance in another AZ
- EBS volumes can be attached to only one EC2 instance at a time, they cannot be shared
- Creating and mounting a EFS volume on Linux:

```
[ec2-user@ip-172-31-1-43 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        488M   60K  488M   1% /dev
tmpfs           497M     0  497M   0% /dev/shm
/dev/xvda1      7.8G  983M  6.7G  13% /

[ec2-user@ip-172-31-1-43 ~]$ echo "/dev/sdf    /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
/dev/sdf    /mnt/data-store ext3 defaults,noatime 1 2

[ec2-user@ip-172-31-1-43 ~]$ cat /etc/fstab
#
LABEL=/     /           ext4    defaults,noatime  1   1
tmpfs       /dev/shm    tmpfs   defaults        0   0
devpts      /dev/pts    devpts  gid=5,mode=620  0   0
sysfs       /sys        sysfs   defaults        0   0
proc        /proc       proc    defaults        0   0
/dev/sdf    /mnt/data-store ext3 defaults,noatime 1 2

[ec2-user@ip-172-31-1-43 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        488M   60K  488M   1% /dev
tmpfs           497M     0  497M   0% /dev/shm
/dev/xvda1      7.8G  984M  6.7G  13% /

[ec2-user@ip-172-31-1-43 ~]$ sudo mkfs -t ext3 /dev/sdf
mke2fs 1.42.12 (29-Aug-2014)
/dev/sdf contains a ext3 file system
```

created on Tue May 22 21:06:34 2018
Proceed anyway? (y,n)

[ec2-user@ip-172-31-1-43 ~]$ sudo mkdir /mnt/data-store

[ec2-user@ip-172-31-1-43 ~]$ sudo mount /dev/sdf /mnt/data-store

[ec2-user@ip-172-31-1-43 ~]$ echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2

[ec2-user@ip-172-31-1-43 ~]$ cat /etc/fstab
#
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2
/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2

[ec2-user@ip-172-31-1-43 ~]$ df -h
Filesystem Size Used Avail Use% Mounted on
devtmpfs 488M 60K 488M 1% /dev
tmpfs 497M 0 497M 0% /dev/shm
/dev/xvda1 7.8G 984M 6.7G 13% /
/dev/xvdf 976M 1.3M 924M 1% /mnt/data-store
[ec2-user@ip-172-31-1-43 ~]$

- https://www.youtube.com/watch?v=DKftR47Ljvw
- Re:Invent 2015 EBS deep dive: https://youtu.be/1AHmTmCkdp8
- Because Amazon EBS volumes are tied to EC2 instances, you will have to click on EC2 from the console dashboard to start working with them. (EBS is not under the storage area)
- EBS volume configurations0 can be changed on the fly except for magnetic standard volumes
- The best practice to stop EC2 instance and then change the volume configuration

- If you change a volume on the fly you must wait for 6 hours before making another change, change the size or type
- Can scale EBS volumes up only
- Encryption must be enabled at the time of EBS volume creation
- EBS volumes are in a single availability zone
- EBS volumes are replicated within a single AZ
- EBS volumes are not accessible from the internet
- EBS Magnetic: 40-200 IOPS
- EBS General purpose: 3 IOPS GB Burstable to 10,000 IOPS
- EBSS Provisioned IOPS SSD: up to 20,000 IOPS consistently up to 320 Mbps throughput
- EBS snapshots do not effect ongoing read and write operations of the production volume
- EBS snapshots can be performed live, while the EBS is being used
- EBS snapshots status is pending until completed
- EBS snapshots are stored in S3
- EBS snapshots are configured in the EC2 dashboard
- EBS snapshots main and then incremental of changed blocks since the last main snapshot
- Excellent AWS presentation slide deck: https://www.slideshare.net/AmazonWebServices/srv413-deep-dive-on-elastic-block-storage-amazon-ebs-78932826

EBS Consists of the following offerings:

- SSD General purpose – GP2 – (Up to 10,000 IOPS) 1Gib – 16 TIB
- SSD Provisioned IOPS – IO1 – Frequently accessed workloads (More than 10,000 IOPS) 1Gib – 16 TIB
- HDD Cold – SC1 – less frequently accessed data 500 Gib – 16 TiB
- HDD Magnetic – Standard – Cheap, infrequently accessed storage
- Pricing varies per region
- Pricing is available for Storage or IOPS
- Pricing is based on allocated storage whether you use it or not

Create EBS volumes in the AWS console.

- In EC2 console under “actions” you attach the volume to the instance
- Should be Elastic Block Storage area under volumes
- On console #lsblk show types and partitions in Linux
- See if there is data on the volume #file –s /dev/xvdf <-based on #lsblk output
- Look to see if it comes back with “data”
- Mount the file system to the EC2 instance: #mkfs –t ext4 /dev/xvdf
- #Mkdir /fileserver
- #Mount /dev/xvdf /fileserver
- #cd /fileserver
- #rm –rf lost+found/
- Create a test file #nano hello_validation
- Create in nano and then Control X, yes on save and enter
- Ls to verify you wrote to the files
- You have just created, formatted and mounted an EBS volume in Linux

EBS Snapshots

- A Volume is a durable, block level storage device that can be attached to a single EC2 instance
- Snapshots are created by copying the data of a volume to another location at a specific time
- You can replicate same Snapshot to multiple availability zones
- A Snapshot is a single point in time view of a volume
- From a Snapshot you can create another EBS volume
- You are charged for storage that is used by a Volume as well as the one used by Snapshots
- Snapshots are stored in a S3 bucket
- There isn't any delay in processing when creating a snapshot and the data on the volume is available to the EC2 instance during the time that the snapshot is being taken
- To move data from one EBS volume to a duplicate volume in a separate region you need to take a snapshot of the EBS volume and copy it to the desired region
- After you've created a snapshot and it has finished copying to Amazon S3, you can copy it from one AWS region to another, or within the same region

- Snapshots are incremental, this means that only blocks that have changes since your last snapshot are moved to S3
- Snapshots of encrypted volumes are encrypted automatically
- Restore an encrypted snapshot then the volume is also encrypted
- Can share snapshots as long as they are not encrypted
- Stop the EC2 instance before you take a snapshot is a best practice
- AWS will stop it for you do not do that
- Snapshot and create new volume allows you to move from magnetic to SSD media for example
- You will need to attach and mount the volume.
- #file –s /dev/xvdf
- #mount /dev/xvdf/ /fileserver
- #cd fileserver

- #ls
- Open EC2 and then on the left panel is the EBS configuration section for volumes and snapshots
- Right click on a volume in the console and select create snapshot is another option you have
- Right click on the snapshot and select “restore” then you can select the volume size and type that is different from the volume the snapshot was created from
- You can change volume types by taking a snapshot and then using the snapshot to create a new volume
- The command line to create a EBS snapshot is: ec2-create-snapshot
- EBS snapshots allow actions to be performed on them with APIs, CLI and AWS console
- Snapshots are incremental, this means that only the blocks that have changed since your last snapshot are moved to S3
- The first snapshot takes a longer time to create than the following incrementally
- Snapshots of encrypted volumes are encrypted automatically and are restored encrypted
- Snapshots can be shared only if they are not encrypted
- If it is a root volume, you should stop the instance before taking the snapshot
- You can change volume types by taking a snapshot and then using the snapshot to create a new volume
- You can create point-in-time snapshots and store them in S3 which will replicate in multiple availability zones per standard S3 replications
- When restoring a volume from a snapshot, the data is available immediately
- Users can share unencrypted snapshots but not encrypted
- Snapshots taken with encryption using the CMK (customer master key)
- There is no direct way to encrypt an existing unencrypted volume. However, you can migrate data between encrypted and unencrypted volumes as follows:
    - Create a new Amazon EBS volume with encryption enabled

- Attach an Amazon EBS volume with encryption enabled to the instance that hosts the data, then migrate the data to the encryption-enabled Amazon EBS volume
- Copy the data from the unencrypted Amazon EBS volume to the Amazon EBS volume with encryption enabled

SSD GP2

- 99.999% availability
- Ratio of 3 IOPS per GB with up to 10,000 IOPS and the ability to burst up to 3000 IOPS for short periods of time for volumes under 1GB
- General purpose block storage
- IOPS is a disk speed metric

Provisioned IOPS SSD

- Designed for I/O intensive applications such as large relational databases or NoSQL databases
- Use if you need more that 10,000 IOPS
- Maximum ratio is 50:1 is permitted between IOPS and volume size, a 8 Gig Volume can support a maximum 400 IOPS (8*50)
- Can be encrypted at the time it is provisioned
- Select the desired availability zone at the time of provisioning
- Select volume size at provisioning (4G min - 16T max)

Magnetic (Standard)

- Lowest cost per gigabyte of all EBS volume types
- Magnetic volumes are ideal for workloads where data is accessed infrequently, and applications where the lowest cost storage is important

EFS: Elastic File Service

- Amazon EFS is a file storage service for Amazon EC2 instances
- Elastic File System is a block based
- file storage service for Elastic Compute Cloud (EC2) instances
- EFS is simple to use and offers a simple interface that allows the creation and configuration of block file systems
- EFS storage capacity is elastic, growing and shrinking automatically as you add and remove files; applications have the storage they need, when they need it
- These is no pre-provisioning of the volume size
- EFS data is copied and stored across multiple AZ's in a region
- Block based storage in EFS (instead of object based in S3)
- Allows for the sharing of files
- Multiple EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for the content of the Web site or any other application running on more than one instance that needs a common data source
- The EFS volume and EC2 instance must be in the same VPC in the same AZ
- A good file/storage system for use by databases and applications, mountable NAS volumes
- EFS can be shared between multiple VM's
- EFS can be accessed by multiple EC2 instances concurrently, acting as a shared storage volume
- Supports the Network File System version 4, (NFSv4) files system
- You are charged for the storage you use, no pre-provisioning is done
- .30 cents per Gig (may have changed)
- Can scale to petabytes in size
- Support for thousands of concurrent NFS connections
- Data in stored across multiple availability zones within a region
- Read after write consistency is standard
- AWS console, Storage and content delivery section and Elastic File System, select region and VPC, mount targets, select subnets, IP

addresses, security groups. Add tags and create

- Actually really easy to set up and configure
- Put EC2 instances in the same security group as the EFS storage
- To mount an EFS drive on EC2 go to EFS in the console and click on the box for the file system you want to use and expand the window, they under file system access there is a EC2 mount instruction icon
- Amazon AMI has the install utilities already installed so all we need to do for the Amazon AMI EC2 instance is #sudo mount –t nfs4 $(curl https://169.254.169.254/latest/metadata/placement/availability-zone.fs-xxxxxx/efs/us-west-2.amazonaws.com:/ /var/www/html
- From the server we can now access the EFS volume #cd /var/www/html #pwd
- You can do this on two EC2 servers, they can both access the same share
- EFS supports thousands of mount points (servers)
- Check file systems, Windows servers may not support NFS4, also EFS may add new files systems as time goes along

AWS Storage Gateway: ASG

- ASG is a Virtual Machine installed on customer premise and communicates with S3 storage in the cloud acting as a front end to AWS
- Storage Gateway connects as an on-premises software appliance with AWS cloud-based storage to provide seamless integration with data security features between on-premises IT environments and the AWS storage infrastructure
- Use the storage gateway service to store data in the AWS Cloud for scalable cost-effective storage that helps maintain data security
- Storage Gateway offers file-based, volume-based and tape-based storage solutions
- AWS Storage Gateway service (ASG) to connect our local infrastructure for files etc. to Amazon cloud services for storage
- AWS Storage Gateway benefits:
    - Local Use: Integrates your data in multiple Amazon Storage Services like- S3, Glacier etc. with our local systems. You continue to use your local systems seamlessly
    - Performance: ASG provides better performance by caching data in local disks. Though data stays in cloud, but the performance we get is similar to that of local storage since frequently accessed data is remotely cached at your datacenter
    - Ease of use: Provides a virtual machine to use with an easy to use interface. There is no need to install any client or provision rack space for using ASG. These virtual machines can live in local system at the remote site or in the AWS cloud
    - Scalable: Storage at a very high scale with ASG. Since backend in ASG is Amazon cloud, it can handle large amounts of workloads and almost unlimited storage capacity
    - Optimized Transfer: ASG performs many optimizations,

only the changes to data are transferred reducing WAN bandwidth

- Main use cases of ASG are as follows:
  - Backup systems: ASG can be used to create backups and store in the cloud. Local storage data can be backed up into cloud services of AWS. On demand, can also restore the data from this backup solution. This can be used as a replacement for Tape based backup systems
  - Variable Storage: With ASG, you can auto-scale to expand and contract storage to match your needs. No local new storage infrastructure is required, there is no capital expenditure. AWS manages the storage systems in the cloud
  - Disaster Recovery: ASG is used for disaster recovery. You can create snapshots of local volumes in Amazon EBS. In the event of a local disaster, use the data in the AWS cloud and recover from the snapshots created in EBS
  - Hybrid Cloud: If you use local applications with cloud services. ASG enable implementing Hybrid cloud solutions that utilize cloud storage services with your local on premises applications

Database basics

Relational databases

- Databases are like a traditional spreadsheet, database, tables, Row, fields (columns).
- Each field always contain the same type of data in a database
- AWS service is called RDS
- AWS Relational database include SQL server, Oracle, MySQL server, PostgreSQL, Aurora, MariaDB
- Read the AWS database FAQ's: https://aws.amazon.com/rds/faqs/
- OLTP is online transaction processing
- SQL uses port 3306 and you must have that open in security groups and access control lists

Non-relational databases

- MongoDB, DynamoDB, NoSQL and others
- DynamoDB is the AWS NoSQL offering
- Database -> Collection, Document, Key Value Pairs
  - Collection = Table
  - Document = Row
  - Key Value Pairs = Fields look like: “firstname” : “pete”

Data warehousing

- Enables business intelligence
- Tools like Cognos, Jaspersoft, SQL server reporting services, Oracle Hyperion. SAP NetWeaver
- Stores very large and complex data sets
- Used for queries on the data

- OLAP: Online Analytic Processing
- Redshift is the AWS big data service
- Data warehousing databases use different types of architecture both from a database perspective and infrastructure layer

RDS: Relational data base

- RDS is a fully managed database offering that includes the instances and database application, they handle resiliency and backups
- AWS RDS Includes SQL database options including SQL, Aurora, MariaDB, MySQL, posgress, Oracle, php with others being added over time
- Used for OLTP: Online Transaction Processing
- RDS instances have automated backups enabled by default on all options
- Automated backups have a 1 – 35 day retention period
- RDS database instance backups are enabled by default
- Full daily snapshot by default is taken and stored in S3
- Snapshots are done manually, user initiated
- Backups are in S3 and included with the cost of the database, no additional charge
- SQL default port number is 3306
- You can force an RDS instance failover if you have multiple availability zones configured
- To have your application check RDS for an error, do an API call then have it look for an **ERROR** code in the response
- By default you can have 40 RDS instances
- Of those 40, up to 10 can be Oracle or SQL Server DB Instances under the "License Included" model
- All 40 can be used for Amazon Aurora, MySQL, MariaDB, Oracle, SQL Server, or PostgreSQL under the "BYOL" model
- If your application requires more DB Instances, you can request additional DB Instances via a request form
- RDS supports a maximum Microsoft SQL server express edition database size of 10GB per database even though RDS databases are 300GB, this Microsoft release only can go to 10GB inside of the 300GB allocation
- Microsoft SQL: There are two different limits -- that of the DB (10GB), and that of the DB instance server storage (300GB). A DB

server instance can host several DBs, or a DB and support files such as logs, dumps, and flat file backups

- Microsoft SQL: allocated storage many not be increased
- See the AWS documentation for full details. Further information: https://d0.awsstatic.com/whitepapers/rdbms-in-the-cloud-sql-server-on-aws.pdf http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_
- With Multi-AZ RDS instances and automated backups, I/O activity is not suspended on the primary during the backup window, backups are taken from the standby. Automated backups do not effect production environments
- All of the AWS RDS database engines support Multi-AZ deployment
- Read replicas are supported by MySQL, MariaDB, PostgreSQL, and Aurora and NOT Microsoft SQL or Oracle
- Queries cannot be run against a multi-AZ secondary copy database it is only the backup copy and not live. Use a read replica instead
- Changing the backup window of any AWS RDS database, takes effect immediately
- When using Multi-AZ, the secondary database is not accessible and all reads and writes must go to the primary or any read replicas created
- RDS failover:
    - If the primary AZ fails, RDS will automatically fail over, the connection string points to the database endpoint, and AWS automatically updates this endpoint to point to your secondary instance.
    - Monitor the environment while Amazon RDS attempts to recover automatically.
    - AWS will update the DB endpoint to point to the secondary instance automatically.
    - You can force a failover from one Availability Zone to another by rebooting the primary instance in the AWS Management Console. This is often how people test a failover in the real world.
- Encrypting an existing DB Instance is not supported if you created the instance without encryption and want to go back and encrypt it

- To encrypt an existing RDS database you should create a new DB Instance with encryption enabled and migrate your data over to it
- By default at no charge, RDS enables automated backups of your DB Instance with a 1 day retention period
- Amazon RDS supports Microsoft SQL Server Enterprise edition and the license is available only under the Bring Your Own License (BYOL) model
- Microsoft SQL server cannot have its storage increased on an active DB instance
- Vertically scaling up using more powerful AMI instance types is one of the simpler options to get more processing power without making any architectural changes
- Read replicas require some application changes but let you scale horizontally
- Most active databases are I/O- bound, by upgrading your storage to General Purpose (SSD) or Provisioned IOPS (SSD) allows for increased request processing
- By default, network access is turned off to a DB Instance. Specify the rules to allow access in a security group that allows access from an IP address range, port, or Amazon Elastic Compute Cloud (Amazon EC2) security group
- Standby replicas (backups) are updated synchronously from the master, read replicas are updated asynchronously
- DB parameter groups allow application of a group of settings to a group of RDS instances
- Create a parameter group in the RDS console, apply parameters and assign the instances into this group that you want to have inherit these parameters

RDS Back-ups, Multi-AZ’s and Read replicas

Backups

- There are two different types of backups for AWS. Automated backups and snapshots.
- Automated backups allow you to recover your database to any point in time within the retention period
- Retention period can be between 1 and 35 days, this is configurable in the console
- Automated backups will take a full daily snapshot and stores transaction logs thought the day
- On a recovery, RDS will first choose the most recent daily backup, and then apply transaction logs relevant to that day, this is the RPO or restore point objective
- This allows a point in time recovery down to a second, within the retention period
- Automated backups are enabled by default. The backup data is stored in S3 and you get free storage space equal to the size of your database
- If you have a RDS instance of 100Gb you will get 100Gb worth of S3 storage for backups for example
- Backups are taken in a user defined window
- During the backup window, storage I/O may be suspended while you data is being backed up and you may experience elevated latency
- Upgrading an RDS instance class will result in the database becoming temporarily unavailable while the DB Instance class is modified. This offline time usually lasts a few minutes, and occurs during the maintenance window for your DB Instance. You can optionally specify the modification should be applied immediately
- Automatic backups of an RDS instance are deleted upon termination of the database. You do have the option of creating a final DB Snapshot upon deletion.
- This allows you to use the Snapshot to restore the deleted DB Instance at a later date
- RDS retains the final user-created DB Snapshot along with all other

manually created DB Snapshots after the DB Instance is deleted

- RDS retains backups of a DB Instance for a limited, user-specified period of time called the retention period, which by default is one day but can be increased up to thirty five days
- Enable automatic backups is a best practice
- Set backup window during low write IOPS times for least impact on operations

Read Replicas

- A read only copy of your production database
- Used for scaling and not disaster recovery
- Used to boost database performance by allowing heavy read intensive operations to read across multiple data stores
- Read from many database replicas instead of just the main database replica
- 5 read replicas per database and then created replicas off of the replicas
- If you have a very heavy read write database this is a good solution
- Read only copy of the production database
- Primarily for very read heavy workloads
- Supports MySQL, PostgreSQL, MariaDB
- Automatic backups must be enabled in order to deploy read replicas
- Read replicas of read replicas are allowed but be careful about latency
- Each read replica will have its own DNS endpoint
- Cannot have read replicas that have Multi-AZ, stay in same availability zone, this may change over time
- However, you can create read replicas of Multi-AZ source databases
- Read replicas can be created in a region or CROSS a region
- Read replicas can be promoted to be their own databases. This breaks the replication from the original database instance
- Read replica in a second region for MySQL and MariaDB but not PastgreSQL.
- Asynchronous replication from the primary RDS instance to the read replicas, so the read replicas have a delay being updated from the read/write replica
- To spread the workload around, use read replicas for read-heavy database workloads
- MySQL and MariaDB server read replicas:
    - Replication is only supported for the InnoDB storage

engine on MySQL
    - XtraDB storage engine on MariaDB.
- Scales read performance without impacting the database, works well for red heavy applications
- Read replicas are for scaling and not disaster recovery
- Can only scale OUT with read replicas, otherwise scale up with a more powerful compute instance
- Scaling RDS is a manual process
- Improves the performance of the primary database by reducing its workload
- RDS read replicas provide increased performance and durability for RDS instances
- Implementing read replicas enables you to scale out beyond the capacity limitations of a single Amazon RDS instance for read intensive databases
- Create one or more replicas of a given RDS source instance and they will serve high-volume application read traffic from multiple copies of your data, this increases the aggregate read throughput
- Read replicas are not supported with Oracle or Microsoft SQL server
- Read replicas are supported on MySQL Server, PostgreSQL, Aurora and MariaDB
- Must enable automatic backups enabled in order to deploy a read replica
- Limit of 5 read replicas per database (except for Aurora)
- Aurora supports up to 15 read replicas
- Can have read replicas off of read replicas (latency considerations), use only if absolutely necessary
- Each read replica will have its own DNS endpoint for access addressing, uses the Route 53 DNS service
- Cannot have read replicas that have multi-AZ turned on (may have changed) this means is that you can create a read replica in a different AZ than the primary database but it is ONLY in the AZ and the read replica are not copied or mirrored out to another AZ
- However, creation of read replicas with Multi-AZ source databases is supported
- Read replicas can be promoted to be their own databases, this stops

the replication from the original database

- Read replicas can be off in another region for MySQL and MariaDB and not for PostgreSQL
- Launch DB instance -> actions and select read replicas, here you can choose a region, AZ and a instance type
- Read replicas can be promoted to the primary instance

Snapshots

- A moment in time copy of the database
- Done manually, you as the user must initiate the operation or off of a script
- Snapshots are stored even after you delete the original RDS instance, unlike automated backups which may or may not be deleted depending on your settings
- Whenever you restore either an Automatic Backup or a manual snapshot, the restored version of the dataset with be a new RDS instance with a new end point
- Instance actions are in the RDS dashboard in the AWS console
- Snapshot section in the console to manage
- Copy and restore snapshots to different regions to move a database to another region
- Scale up take a snapshot and restore to a higher performance EC2 instance to scale up processing power
- Snapshots allow you to build a new database and retain your data
- Snapshots are stored in S3 no additional charge up to 100% of your consumed database storage for an active DB instance
- Snapshots persist until the user deletes them
- API: CreateDBSnapshot and DeleteDBSnapshot
- Cross region snapshots are good for disaster recovery
- Cross region snapshots can be used for migrations to a different region

Database Encryption

- Encryption of data at rest is supported for MySQL, Oracle, SQL server, PostgreSQL and MariaDB
- Encryption uses the AWS Key Management System (KMS) service
- Once your RDS instance is encrypted the data stored at rest and the underlying storage is encrypted, as are the automated backups, read replicas, and snapshots
- As of the present time (2016) encrypting an existing DB instance is not supported
- To encrypt a RDS unencrypted database that is already created, you must create a new DB instance with encryption enabled and migrate your data into it

- Data at rest support for MySQL, Oracle, SQL server, PostgreSQL & MariaDB

Database multi-AZ

- At this time, you cannot have a multi-AZ copy of your read/write replica
- Events would cause Amazon RDS to initiate a failover to the standby replica:
  - Loss of availability in primary Availability Zone
  - Loss of network connectivity to primary
  - Compute unit failure on primary
  - Storage failure on primary
- For RDS MySQL, MariaDB, PostgreSQL and Oracle databases
- Converting the RDS instance from Single-AZ to Multi-AZ, the following happens:
  - Snapshot of your primary instance is created
  - New standby instance created in a different Availability Zone, from the snapshot
  - Synchronous replication is configured between primary and standby instances
- Multi-AZ deployments utilize synchronous replication
  - Database writes concurrently on both the primary and standby
  - Standby remains up-to-date in the event a failover occurs
  - Database is synchronously replicated to another availability zone in the same region
- Failover to a standby automatically occurs if the master fails
- Planned maintenance is applied first to standby databases
- Standby databases are not active and cannot be used, use a read replica instead
- Need to add a subnet group before creating a multi-AZ one subnet per AZ
- If multi-AZ failover is enabled, a duplicate copy of the database is kept in a separate AZ.
- If there is failure in the primary database's AZ, AWS will automatically switch the DNS CNAME (canonical) DNS record from the primary to the failover backup instance to redirect traffic

- The standby is promoted to become the primary
- A best practice to implement database connection retry at the application layer

- SQL Server
- Oracle
- MySQL Server
- PostgreSQL
- MariaDB
- Disaster recovery only
- Not an active Database instance, it is a backup of the primary
- Synchronous replication from the master to the replica
- To automatically failover from one geographic location to another you should use Multi-AZ for RDS
- Primary becomes unavailable so RDS automatically (within 60-120) updates the DNS to point to the secondary

RDS option groups

- Some DB engines offer additional features that make it easier to manage data and databases, and to provide additional security for your database.
- Amazon RDS uses option groups to enable and configure these features.
- An option group can specify features, called options that are available for a particular Amazon RDS DB instance.
- Options can have settings that specify how the option works.

DynamoDB

- NoSQL offering from Amazon
- Non-relational database
- DynamoDB is covered in the developer associate certification and a general knowledge is required for the Solutions Architect Associate certification
- Commonly called a NoSQL database
- Fully managed AWS database service, Software as a Service model
- DynamoDB is a highly scalable NoSQL database
- Extremely scalable with fast performance
- Scalability is its strong use case, outperforms relational databases at petabyte scale
- High performance
- Highly resilient to availability zone failures
- Data is replicated across multiple independent data centers
- Data is stored on high performance SSD disk drives (not S3)
- No practical storage limitations
- HTTPS access supported
- AWS Hardware failures are transparent the customer
- Stores any amount of data with no practical limits
- Since it is SaaS, and managed by AWS, you do not have any access to the underlining OS.
- NoSQL database platform designed by AWS, DynamoDB is a AWS created NoSQL database
- schemaless key-value store
- On initial configuration, the required throughput capacity is configured
- There are no instance sizes or storage types to choose from. AWS manages the scaling of the underlying compute and storage systems to match your workload
- Fast and reliable
- Query operations enables you to query a table using the partition key and an optional sort key filter
- If the table has a secondary index, you can also query the index

using its key

- Query only tables that have a composite primary key (partition key and sort key)
- You can also query any secondary index on such tables
- Query is the most efficient way to retrieve items from a table or a secondary index (as compared to a scan)
- A Primary key, hash key and partition key all appear to be the same thing
- Scan is the least efficient way to retrieve items from a table
- Scan can be done on a table or secondary index, for large tables a scan can consume a large amount of resources
- Scan operations reads every item in a table or secondary index
- A Query operation finds items in a table or a secondary index using only primary key attribute values and a Scan operation reads every item in a table or a secondary index
- There can be more than one local secondary index per table, and all of them must be created when the table is created
- The basic Data Model in Amazon DynamoDB consists of following components:
    - Table: A collection of data items. It is similar to a table in a Relational Database. There can be infinite number of items in a table. There must be one Primary key in a Table
    - Item: An Item is made up of a primary key or composite key and a variable number of attributes
    - The number of attributes in an Item is not bounded by a limit. But total size of an Item can is a maximum 400 kilobytes
    - Attribute: Associate an Attribute with an Item, set a name as well as one or more values in an Attribute. Total size of data in an Attribute is maximum 400 kilobytes
- Auto scaling uses a “push button” can scales with no downtime or reconfigurations.
- DynamoDB is integrated into Cloud Trail for logging API calls
- Uses fast SSD storage
- Latency in the single-digit milliseconds
- No table size or throughput limits

- DynamoDB Integrates with Elastic Map Reduce / Hadoop services for big data deployments
- Replicated across three different AWS data centers automatically
- Every DynamoDB table is automatically replicated across multiple Availability Zones in the Region, this is done automatically by Amazon as part of the service.
- Read Consistency options: You specify at the time of a read request if a read should be eventually or strongly consistent
- Strongly consistent reads mean the data is stored in all AZ's before it can be read.
- Eventual Consistent reads in one second delay approximate, this is the latency or delay to write the data into all tables in all AZ's.
- Performance scales in a linear method and not stacked, so the performance of DynamoDB is predictable.
- Fully integrated with AWS IAM (Identity Access Manager) for granular user access control.
- Has similar characteristics as Key/Value and Document store databases.
- Key benefits of using Amazon DynamoDB:
    - Administration: No time or expertise to administer the databases, AWS manages this service for you. There are no servers to provision or manage. You create your tables and use them. This is a fully manages SaaS offering from Amazon
    - Scalability: There is an option to specify the capacity needed for each table. The scalability is done behind the scenes by AWS for DynamoDB
    - Fast Performance: Delivers very fast performance with low latency from low to extremely high scale. DynamoDB uses fast SSD storage and partitioning behind the scenes to achieve the user specified throughput
    - Access Control: Integrates with IAM to create extremely granular access control to ensure that your data is secure
    - Flexible: Support for both document and key-value data structures
    - Event Driven: Can trigger Lambda with DynamoDB to

support event driven programming

- Supports both document and key based NoSQL databases.
- APIs in DynamoDB are generic enough to support document and key based NoSQL databases
- APIs available in DynamoDB: CreateTable, UpdateTable, DeleteTable, DescribeTable, ListTables, PutItem, GetItem, BatchWriteItem, BatchGetItem, UpdateItem, DeleteItem, and Query Scan
- DynamoDB is used for storing structured data. Data is indexed by a primary key for fast access
- Read and write operations are low latency because of SSD storage media type.
- S3 storage is mainly used for storing unstructured binary large objects based (BLOB) data. It does not have a fast index like DynamoDB
- S3 and Dynamo DB differ in the size of the amount of data each can store. In DynamoDB the size of an item can be maximum 400 kilobytes, compare this to S3 that supports object sizes as large as 5 terabytes
- DynamoDB is better used when you need to store small objects with frequent access and S3 is ideal for storing very large objects with infrequent or slower access requirements
- The AWS DynamoDB library for Python is called boto3 and can be downloaded from AWS.
- You can install DynamoDB on a local workstation for testing
- SSD storage offers low latency and reliability and consistent performance for both reads and write operations
- Each data block is stored three times in the background
- Data write request using the API, when you use the console everything is a AI request, you get a “OK (200)” response back indicating a successful write, there can be 200 return responses for 2 of 3 successful writes, so full consistency is not achieved even when a 200 code goes out
- Eventual consistency is the term and performance takes precedence over consistency. You can get around this to ask the DB to perform a most recent return on reads to ensure fresh data, each read checks multiple locations, WS charges extra for this function

- Eventual consistency is the default mode
- The user can control the consistency
- Eventual consistency is around half the cost of strong consistency
- There are collections of tables in each region
- The namespace is at the region level
- Tables are the highest level data structure
- The schema is not fixed at the table level and can be changed later in production
- Rows are called items in DynamoDB
- Each item contains one or more bits of data called attributes, there is no enforcement of structure
- Attributes include partition key (hash key) and it the primary table index
- A support key attribute, used to be called a range key
- Scaler data type: String “mystring” Number “123” Binary: “mystring123” Boolean: true/false
- Set data type: multiple scalers as a grout, string sets, number sets and must be of the same data set types.
- Document data types of lists and a MAP an unordered structure (JSON)
- Monitoring and statistics are much more detailed if you use CloudWatch
- Capacity is performance of a table, not the size of a table and is measured units. Capacity is a performance directive to DynamoDB, that is defined per table and separates read and write controls. This can be changed at any time and it takes time to make the changes in the background.
- There are a lot of variable in the delay and it has an impact on performance. Over 3,000 read capacity units and 1000 write capacity are important that takes planning
- This activity should be limited to longer term changes, such as 3 to 4 changes per year. For decreases per calendar day and can increase as often as you like and is on a per table basis
- A Read capacity unit is up to 4KB of data and a write capacity unit is up to 1KB in size
- For larger operations you round up to the net highest KB boundary
- There can be more than one local secondary index per table, but

they must all be created when the table is created

- NoSQL databases (like Amazon DynamoDB) are good for scaling to hundreds of thousands of requests with key/value access to user profile and session
- Query is the most efficient operation to find a single item in a large table
- Local secondary indexes can only be created when the table is first created
- DynamoDB automatically partitions data over multiple servers in the background to meet requested capacity you define
- Data is synchronously replicated across multiple availability zones within a region, this is done automatically
- If you receive a ProvisionedThroughputExceededException error. But the DynamoDB metrics show that your table or Index has not been operating at maximum provisioned throughput. This error is triggered by the throughput is not being balanced across the partitions. One partition is being subjected to a disproportionate amount of the traffic and, therefore, exceeding limits
- DynamoDB does not have a server-side feature to encrypt items within a table
- Implement an encryption solution external of DynamoDB such as a client-side library to encrypt items before storing them, or a key management service like AWS Key Management Service to manage keys that are used to encrypt items before storing them in DynamoDB
- DynamoDB replicates data across three facilities in an AWS region to provide fault tolerance in the event of a server failure or Availability Zone outage
- Loosely coupled systems work well with DynamoDB
- Re:Invent 2017 Dynamo Global tables
    - Fully managed, multi-master, multi-region database
    - Replication across regions
    - Global customers
    - Disaster proof, withstands a complete regional failure
- YouTube Dynamo Database presentation https://www.youtube.com/watch?v=FNFRTnp9Qh4
- Re:Invent 2017 Dynamo backup and restores

- New on-demand backup services for archival and compliance
- Point in time restore for short term retention and protection against application errors (2018)
- Backs up hundreds of TB instantaneously with no performance impact

Neptune

- Fully managed graph database
- For big data
- Stores billions of relationships query with milliseconds latency
- 6 replicas across three AZs with full backup and restore to S3
- Queries with Gremlin and SPARQL
- https://aws.amazon.com/Neptune
- Re:Invent 2017 introduction

Database Migration Services: DMS

- Migrate databases into cloud or inside the cloud
- Automatic database migration onto AWS from external databases
- Migrate Oracle to Aurora and AWS does the conversion
- Eliminates Oracle license costs
- Aurora may be a vendor lock-in, verify that if you migrate to Aurora that you have the ability to migrate out in the future
- This service results in a large cost saving from Oracle licensing
- There is no downtime during the migration process
- You can migrate into AWS Aurora from Oracle, SAP, postgress etc.

- This service is used to migrate your production database to AWS
- AWS manages all the details of the migration process such as the data type transformation, compression, and parallel transfer (for faster data transfer) and validates format changes from source to destination
- AWS uses a schema conversion that automates the conversion of the source database schema custom code, including views, stored procedures, and functions, to a format compatible with the target AWS database
- Move away from Oracle to a free open source database using the schema conversion tool
- https://aws.amazon.com/dms/

Aurora

- Amazon created database in the RDS suite
- Competes with Oracle database products
- Advanced database
- Only runs on AWS as it is a AWS developed service
- MySQL compatible relational database engine that combines the speed and availability of high-end commercial databases with simplicity and cost-effectiveness of open source databases
- Aurora provides up to five times better performance than MySQL at a price point $1/10^{th}$ that of a commercial database while delivering similar performance and availability
- Supports up to 15 read replicas across 3 AZs
- Auto-scale new read replicas was added in 2017
- Seamless recovery from read replica failures, one second failover
- Single write replica, if it fails, a read replica gets promoted, you will incur around 30 seconds of downtime
- Scaling starts at 10Gb and scales in 10Gb increments to 64Tb using auto scaling, you pay for the storage you consume
- Compute resources can scale up to 32vCPUs and 244Gb of memory
- Some downtime may be required to vertically scale
- You can import Oracle data into Aurora
- 6 copies of your data is the default number of database copies Aurora stores
- Copies of your data are stored in a minimum of 3 availability zones with 2 copies in each AZ.
- Can lose of up to two copies of data without effecting database write availability and up to three copies without effecting read availability
- Aurora storage is self-healing
- Data block and disks are continuously scanned for errors and repaired automatically
- Replicas: 2 types available:
    - Aurora replicas (up to 15) auto failover

    - MySQL read replicas (currently at 5) no auto-failover
    - Replicas are assigned priorities with 0 the highest, this determines which replica gets promoted to primary in a failover
- Aurora is not covered in the free tier
- May not be available in all regions
- Starts with a r3 large instance and goes up, Aurora is CPU intensive
- The Aurora cluster identifier is used for DNS for application connect strings coding
- Aurora supports encryption
- Initial provision time is 5-10 minutes
- Replicas are read-only
- Instance actions for failover, restore, delete, snapshot and other management tasks
- Aurora scales vertically, you need a larger server instance to scale, not add more servers (does not do horizontal)
- Aurora is rated at 99.99% availability
- Supports instant crash recovery
- Cache layer survives DB restarts improving read responses
- ~10ms replica lag
- Up to 15 read replicas to scale database read queries but just one read/write replica
- Additional information: https://aws.amazon.com/rds/resources
- Aurora multi-master
    - New at re:Invent 2017, Aurora multi-master scale-out for both reads and writes
    - First relational database service with a scale-out across multiple data centers
    - Zero application downtime from any node failure
    - Zero application downtime from any AZ failure
    - Faster write performance
    - Multi-region in 2018
- Aurora serverless re:Invent 2017
    - On demand, autoscaling, serverless database
    - Do not need to provision instances
    - Automatically scales capacity up and down
    - Starts up on demand and down when not in use

- Only pay by the second when the database is in use

Data Migration services

Snowball

- Data transfer to AWS storage offerings such as EBS, S3 etc.
- Standardized method of moving terabytes to the cloud from your on-premise storage
- Snowball edge, edge storage at your data center that then uploads to cloud, snowball edge has compute local, AWS on premise hardware
- Moves large amounts of data into and out of the AWS cloud using portable storage devices for transport, these are storage kiosks sends to you
- Transfers data directly to the AWS cloud bypassing the internet and using Amazon's high-speed internal network
- Send a hard disk to AWS and they will transfer the data(old model), obsolete
- Snowball was released in 2015
- Snowball w originally called import/export before it was called snowball
- If the initial data is in Glacier, that data must be exported to S3 and then over to Snowball
- Import to S3 and then export from S3 to snowball
- In the AWS under migration (not storage)
- Job Dashboard console shows all of the steps, shipping, transfer etc.
- Create a job, plan job, details, set security and so forth
- SNS is used to send emails on snowball shipping and other status updates
- Need to load a software client to connect to the snowball to manage the data transfer process
- https://aws.amazon.com/snowball/

Snowball appliance

- AWS owned hardware appliance that AWS sends you to load your data onto, then you return it and AWSD with transfer the data into their cloud storage offerings
- Snowball data gets imported into AWS S3
- Data transfer into the AWS cloud at much faster than internet speed
- A kiosk that is larger than a briefcase and very heavy
- Snowball is a petabyte-scale data transport solution that uses an AWS owned secure appliances to transfer large amounts of data into and out of AWS
- The snowball appliance addresses common problems associated with large-scale data transfer including network costs, long transfer times, and security concerns
- Transferring data with snowball is simple, fast, secure, and can be as little as one-fifth the cost of high speed internet
- 80TB snowball appliances available in all regions
- AWS sends you the appliance, you migrate your storage data into it and ship it back to AWS where the take the data and sore in in S3
- Uses multiple layers of security designed to protect your data including:
    - tamper resistant enclosures
    - 256-bit AES encryption
    - Industry-standard Trusted Platform Module (TPM) designed to ensure both security and full chain-of-custody of your data
- Once the data transfer has been processed and verified, AWS performs a software erasure of the snowball appliance
- The kiosk includes a Kindle for management, shows status, IP address
- Network interfaces are SFP, copper and optical Ethernet
- Kindle dashboard gives you the unlock code

Snowball Edge appliance

- 100TB data transfer appliance with on-board storage and compute capabilities
- Snowball edge moves large amounts of data into and out of AWS, as a temporary storage tier for large local datasets, or to support local workload in remote or offline locations
- Snowball Edge connects to your existing applications and infrastructure using standard storage interfaces
- Edge streamlines the data transfer process and minimizes setup and integration
- Snowball Edge can cluster together to form a local storage tier and process your data on-premises
- Edge ensures your applications continue to run even when they are not able to access the cloud since it is a local appliance in your data center
- Local processing on Snowball edge brings compute capacity to the edge
- You can launch EC2 or Lambda instances in the appliance, edge computing
- AWS datacenter in a box
- https://aws.amazon.com/snowball-edge/

Snowmobile

- Sea container on a semi
- Petabyte and Exabyte storage transfer
- Can transfer up to 100PB per snowmobile
- 45-foot long shipping container, pulled by a semi-trailer truck
- One use case is data center migrating
- One Exabyte is 10 snowmobile trailers and it will take 6 months to fill up and move

Server Migration Services: SMS

- Migrates your on-premises workloads to AWS
- automate, schedule, and track incremental replications of live server volumes
- Converts VM's such as VMWare machines
- AWS Server Migration Service will automatically replicate live server volumes to AWS and create Amazon Machine Images (AMI) as needed
- Incremental server replication reduces server downtime significantly
- Designed for large-scale migrations
- SMS tracks the progress of each migration
- Perform migrations faster while minimizing network bandwidth, by migrating only incremental changes made to on-premises servers
- 50 VM images can be migrated at a time
- AWS Server Migration Service is free to use
- Pay only for the storage resources used during the migration process
- https://aws.amazon.com/server-migration-service/

Analytics

- Data analytics (DA) is the process of examining data sets in order to draw conclusions about the information they contain, increasingly with the aid of specialized systems and software
- Data analytics technologies and techniques are widely used in commercial industries to enable organizations to make more-informed business decisions and by scientists and researchers to verify or disprove scientific models, theories and hypotheses
- Analytics pipelines for big data, IoT, gaming, logging data
- A family of applications used in business intelligence, reporting and online analytical processing (OLAP)
- AWS analytics product family:
    - Athena
    - Elastic Map Reduce (EMR)
    - Elasticsearch
    - Kinesis
    - Quicksite
    - Redshift
    - Glue
    - Data pipeline
- Not a big part of the AWSCSA exam but good to know

Athena

- SQL services on S3
- Interactive query service used to analyze data in S3 using standard SQL formats
- Serverless, no infrastructure
- Pay for only the queries run
- Point to your data in Amazon S3, define the schema, and start querying using standard SQL
- Flat files turned into searchable database
- Athena quickly analyzes large-scale datasets in S3 using SQL
- AWS handles all of the setup and management of the service
- https://aws.amazon.com/athena/

Redshift

- Data warehouse offering from AWS
- Petabyte scale data warehousing service
- Used for OLAP – Online Analytics Processing
- Transfer to redshift and run query to avoid loading down the live database
- Big data offering
- Used for reports generation and data mining
- Redshift is purpose-built as an OLAP data warehouse
- Redshift is a petabyte scale data warehouse service in the Amazon cloud
- Pricing starts a $0.25 per hour with no commitments or upfront charges and to $1,000 per terabyte per year. About 1/10 the cost of most other big data offerings on the market
- Data warehousing databases use a different type of architecture from a database and infrastructure layer perspective
- Redshift is all about columns
- Columnar storage uses a 1024Kb/1MP block size
- Single node start up to 160GB of storage and scales to huge multimode systems
- Not intended to be used for unstructured NoSQL data or highly dynamic transactional data
- Multi-node consists of a leader node that is used to manage client connections and to receive queries
- Compute nodes store data and perform queries and computations, there can be up to 128 compute nodes
- Columnar data storage: instead of storing data as a series of rows, Redshift organizes the data by column, unlike row based systems, which are ideal for transactional processing, column-based systems are idea for data warehousing and analytics, where queries often involve aggregate performed over large data sets.
- Since only columns are involved in the queries are processed and columnar data is stored sequentially on the storage media, column-based systems require far fewer I/O’s, greatly improving query

performance

- Columnar storage block size is 1,024KB the 1MB block size is much larger than traditional databases and reduces the required disk I/O
- Advanced compression: Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk.
- Redshift implements multiple compression techniques and can achieve compression comparable to traditional relational data stores.
- Redshift doesn't require indexes of materialized views and uses less space than traditional relational database systems.
- When loading data into an empty table, Redshift automatically samples your data and selects the most appropriate compression scheme
- Compression speeds up performance
- Massively Parallel Processing (MPP): Redshift automatically distributes data and query load across all nodes
- Not recommended for data sets under 100GB
- Redshift is based on PostreSQL
- Access Redshift using standard business intelligence reporting tools
- Data gets replicated between nodes in a cluster
- Continuously backed up to S3 with snapshots (stored for 1 - 35 days)
- User initiated snapshots are retained upon cluster deletion
- Redshift quick recovery by using snapshots
- It is easy to add notes to your data warehouse and enable you to maintain fast query performance as your data warehouse grows
- Pricing: compute node hour's total number of hours you run across all your compute nodes during the billing period. You are billed for 1 unit per node hour, so a 3-node data warehouse cluster running persistently for an entire month would incur 2,160 instance hours. You will not be charged for the leader node hours; only compute node hours will incur charges
- You get Charged for backup data
- Charged for data transfer only inside a VPC, not outside of it
- Security in transit is encrypted using SSL, at rest the encryption

used is AES-256

- By default Redshift takes care of the key management
- You can manage your own keys though HSM (Hardware security modules) or AWS Key Management Service
- Not currently multi-AZ, only runs in a single availability zone (may have changed)
- Can restore snapshots to new AZ's in the event of an outage
- Data in Amazon Redshift must be structured by a defined schema, rather than supporting arbitrary schema structure for each row
- Large data files such as BLOB data, which includes digital video, images, or music shouldn't be stored on Redshift
- When you choose AWS KMS for key management with Amazon Redshift, there is a four-tier hierarchy of encryption keys.
- These keys are the master key, a cluster key, a database key, and data encryption keys
- Amazon Redshift clusters reside within one Availability Zone (AZ). If you want to have a multi-AZ setup for Amazon Redshift you can set up a mirror, and then self-manage replication and failover
- Redshift can start with as little as a single 160GB node and scale up all the way to a petabyte or more of compressed user data using many nodes
- https://aws.amazon.com/redshift/

Elastic Map Reduce: EMR

- Managed Hadoop framework
- Big Data processing
- Process large amounts of data
- Hadoop is the application that is managed by AWS and marketed as Elastic Map Reduce (EMR)
- Two offerings, transient and persistent
    - A transient cluster is shut down between analysis jobs
    - A persistent cluster runs continuously
- Data on the Hadoop File System (HDFS) storage is lost when a transient cluster is shut down
- You can use SSH to access the underlying operating systems of EMR and EC2
- EMR uses Apache Hadoop as its distributed data processing engine
- Hadoop is an open source, Java software framework that supports data-intensive distributed applications running on large clusters of commodity hardware
- Hive, Pig, and HBase are packages that run on top of Hadoop
- To increase EMR performance, reduce the input split size of the MapReduce job configuration, then adjust the number of simultaneous mapper tasks so that more tasks can be processed at once
- Data is loaded into S3, EMR processes the stores it back to S3
- Completed jobs are then retrieved from S3 storage
- EMR can shut the cluster down after the job is completed or leave it running for more jobs
- Clusters can expanded or contracted as needed (Elastic)
- Storing data in S3 means it can be accessed from multiple EMR clusters
- EMR can launch inside a VPC
- Multiple pricing options
- EMR is fault tolerant for core node failures and continues job execution if a slave node goes down
- If a master node fails, AWS recommends that you back up your

data to a persistent data store such as Amazon S3

- EMR starts your instances in two EC2 security groups, one for the master and another for the slaves
- The master security group has a port open for communication with the service
- It also has the SSH port open to allow you to securely connect to the instances via SSH using the key specified at startup
- The slaves start in a separate security group, which only allows interaction with the master instance
- By default, both security groups are set up to prevent access from external sources, including Amazon EC2 instances belonging to other customers
- Because these are security groups in your account, you can reconfigure them using the standard Amazon EC2 tools or dashboard
- Using EMR you can launch a persistent cluster that stays up indefinitely or a temporary cluster that terminates after the analysis is complete
- In either scenario you only pay for the hours the cluster is up
- https://Aws.amazon.com/elasticmapreduce
- EMR masterclass: https://youtu.be/zc1_Rfb_txQ

SageMaker

- Introduced re:Invent 2017
- Automated Big Data, hides the complexity
- Brings big data to smaller companies
- Machine learning models
- Pre-built notebooks for common problems
- Built-in high performance algorithms
- Frameworks Caffe2, CNTK, PyTorh, Torch
- https://aws.amazon.com/sagemaker/

Cloud Search / Elastic Service

- AWS managed search service that makes it simple and cost-effective to set up, manage, and scale a search solution for your website or application
- Includes features such as highlighting, autocomplete, and geospatial search
- CloudSearch will automatically provision the required resources and deploy a highly tuned search index
- Autoscaling is part of the service
- Provides automatic monitoring and recovery for your search domains
- Fully managed custom search service. Hardware and software provisioning, setup and configuration, software patching, data partitioning, node monitoring, scaling, and data durability are handled by AWS
- Pricing is low hourly rates, and only for the resources you use
- Integrated with IAM, support encryption and HTTPS
- Angolia is a competitor
- https://aws.amazon.com/cloudsearch/

Data Pipeline

- Moves data from one place to another in AWS
- S3 to DynamoDB for example
- Processes and moves data between different AWS compute and storage services
- Data Pipeline helps you move, integrate, and process data across AWS compute and storage resources, as well as on-premises private data center resources
- Data Pipeline supports integration of data and activities across multiple AWS regions
- Enables the scheduling of regular data movement and data processing activities on a user defined schedule in the AWS cloud
- Integrates with on-premise and cloud-based storage systems to allow developers to use their data wherever it resides, and in their required format
- Data Pipeline allows you to define a dependent chain of data sources, destinations, and predefined or custom data processing activities that is called a pipeline
- AWS pipeline regularly performs processing activities such as distributed data copy, SQL transforms, MapReduce applications, or custom scripts against destinations such as S3, RDS, or DynamoDB
- The scheduling, retry, and failure logic for these workflows is a highly scalable and fully managed service
- Data Pipeline ensures that your pipelines are robust and highly available
- An activity is an action that Data Pipeline initiates on your behalf as part of a pipeline.
- Example activities are EMR or Hive jobs, copies, SQL queries, or command-line scripts
- Amazon Data Pipeline allows you to run regular Extract, Transform, Load (ETL) jobs on Amazon and on-premises data sources
- Good for ETL operations as pipeline can be scheduled and resources (such as EC2 instances) can be created if needed

- https://aws.amazon.com/datapipeline/

QuickSight

- Business analytics
- Dashboards of data
- AWS business analytics service that makes it easy to build visualizations, perform ad-hoc analysis, and quickly get business insights from your data
- QuickSight can connect to your data, perform advanced analysis, and create graphical visualizations, graphical dashboards can be accessed from any browser or mobile device
- Don’t expect to see this on the exam but be aware of this serivce
- https://aws.amazon.com/quicksight/

Security and Identity

AWS Security and compliance

- AWS provides IT control information to customers through either specific control definitions or general control standard compliance
- IT governance and compliance is always the customer's responsibility, despite deploying their IT operations to the AWS platform
- Individual components of a workload can be moved into AWS, but it is always the customer's responsibility to ensure that the entire workload remains compliant with various certifications and third-party attestations
- AWS regularly scans public-facing, non-customer endpoint IP addresses and notifies appropriate parties if any vulnerabilities are found
- AWS does not scan customer instances, and customers must request the ability to perform their own scans in advance or intrusion alarms will be triggered inside the AWS monitoring applications
- AWS security notifies the appropriate parties to remediate any identified vulnerabilities
- Risk and compliance communications responsibilities: AWS publishes information about the AWS security and control practices online, and directly to customers under NDA. Customers do not need to communicate their use and configurations to AWS
- AWS has developed a strategic business plan to identify any risks and has implemented controls to mitigate or manage those risks
- Customers should also develop and maintain their own risk management plans to ensure they are compliant with any relevant controls and certifications
- The collective control environment includes people, processes, and technology necessary to establish and maintain an environment that supports the operating effectiveness of AWS control framework. Energy is not a discretely identified part of the control environment
- Review information available from AWS together with all other information, and document all compliance requirements

- Verify that all control objectives are met and all key controls are designed and operating effectively
- Identify and document controls owned by all third parties
- Amazon Virtual Private Cloud (Amazon VPC) help minimize the DDoS attack surface area
    - The attack surface is composed of the different Internet entry points that allow access to your application
- The strategy to minimize the attack surface area is to
    - Eliminate non-critical Internet entry points
    - Separate end user traffic from management traffic
    - Obfuscate necessary Internet entry points to the level that untrusted end users cannot access them
    - Decouple Internet entry points to minimize the effects of attacks. This strategy can be accomplished with Amazon VPC
- An audit and compliance portal for on-demand access to download AWS' compliance reports and manage select agreements is called artifact
- AWS Artifact provides on-demand access to AWS security and compliance reports and select online agreements.

Identity Access Management: IAM

- Fundamental user access and security component of AWS
- Create users and assign them access rights to AWS services
- Assign users permissions
- Global AWS service for management access to AWS services, one IAM applies to all regions of your account it is a Global service and not connected to any region
- When first created the IAM systems start with no permissions, you then assign the user (or groups) permissions
- Very important to know IAM for the AWS CSSA exam
- The important components of IAM are as follows:
    - IAM User: IAM User is a person or service that interacts with AWS. User can sign into AWS Management Console for performing tasks in AWS
    - IAM Group: A collection of IAM users. Specify permissions to an IAM Group. This helps in managing large number of IAM users. You can add or remove an IAM User to an IAM Group to manage the permissions
    - IAM Role: IAM Role is an identity to which you assign permissions. A Role does not have any credentials (password or access keys). You can temporarily give an IAM Role to an IAM User or service to perform certain tasks in AWS
- An instance profile is a container for an AWS Identity and Access Management (IAM) role that you can use to pass role information to an Amazon EC2 instance when the instance starts
- The IAM role should have a policy attached that only allows access to the AWS Cloud services necessary to perform its function
- Create a Permission to access or perform an action on an AWS Resource and assign it to a User, Role or Group
- You can create Permissions on resources such as an S3 bucket, Glacier vault etc. and specify who has access to the resource
- IAM Policy: An IAM Policy is a document that contains permissions to specify Actions, Resources, services and Effects

- IAM policies are in the JSON format. You attach a Policy to an IAM User or Group.
- Key AWS IAM points
    - A new User in IAM does not have any permissions to do anything until granted access rights
    - AWS IAM assigns an Access Key and a Secret Access Key to a new User
    - Access keys should never be stored on an AMI, use roles instead
    - An Access Key cannot be used to login to AWS Console, use the private key
    - Use the Access Key to access AWS via an APIs or Command Line interface
    - IAM is a universal application. It is global across all the regions in AWS
- When first setting up our AWS account, you get a root account that has complete Admin access.
- You can enable cross-account access with IAM
- To delegate permission to access a resource, create an IAM role that has two policies attached
    - The permissions policy grants the user of the role the needed permissions to carry out the desired tasks on the resource
    - The trust policy specifies which trusted accounts are allowed to grant its users permissions to assume the role
    - The trust policy on the role in the trusting account is one-half of the permissions
    - The other half is a permissions policy attached to the user in the trusted account that allows that user to switch to, or assume the role
- IAM policies get assigned to:
    - Users
    - Groups
    - Roles
- Identities that can assume IAM roles:
    - Users
    - Applications

    - Services
- IAM is not used for OS and application authentication, it is used for AWS services
- Can attach to an on premise directory services using LDAP use SAML (Security Assertion Markup Language) to enable single sign-on between AWS and LDAP
- Users log into IAM with a URL specified in that users IAM configuration page and looks like this: https://1231213123123.signin.aws.amazon.com/console
- This account includes our account ID and you can modify this so it includes a name instead
- https://aws.amazon.com/iam/

Identity Access Management Roles

- Kind of like TACACS where credentials are centralized.
- Under IAM and Roles in the AWS console
- Create the role and then launch EC2 instance
- Enable identity access
- Can only assign a role when the EC2 instance is launched, not when it is running
- However you can change the permissions when the role and EC2 is running
- Roles are more secure than storing your access key and secret access key on individual EC2 instances
- Roles are easier to manage since there is no key management to be performed
- Roles can only be assigned when that EC2 instance is being provisioned
- Role are universal, you can use them in any region (part of AIM)
- AWS re:Invent course on IAM policy https://www.youtube.com/watch?v=y7-fAT3z8Lo
- It is a best practice to use roles for applications that run on EC2 instances

Inspector

- Agent on VM's inspects VM's and provides security reporting
- An automated security assessment service that helps improve the security and compliance of applications deployed on AWS
- Automatically assesses applications for vulnerabilities or deviations from best practices
- Inspector produces a detailed list of security findings prioritized by level of severity
- Includes a knowledge base of hundreds of rules mapped to common security best practices and vulnerability definitions
- Examples of built-in rules include checking for remote root login being enabled, or vulnerable software versions installed
- These rules are regularly updated by AWS security researchers
- Enforces security standards
- Streamlines security compliance
- Not in the associate level exams
- https://aws.amazon.com/inspector/

Certificate Manager

- Provides SSL certificate Authority
- Free SSL certificates for your domain names
- An AWS service that lets you easily provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services
- Certificate Manager removes the process of purchasing, uploading, and renewing SSL/TLS certificates
- Quickly request a certificate, deploy it on AWS resources such as Elastic Load Balancers, Amazon CloudFront distributions, and APIs on API Gateway, and let AWS Certificate Manager handle certificate renewals
- SSL/TLS certificates provisioned through AWS Certificate Manager are free. You pay only for the AWS resources you create to run your application
- https://aws.amazon.com/certificate-manager/

Key Management services: KMS

- Fully AWS managed encryption service
- The KMS service stores what is called the Customer Master Keys (CMK)
- KMS Customer Master Keys (CMKs) are the fundamental resources that AWS KMS manages
- CMKs (customer master key) can never leave AWS KMS unencrypted, but data keys can
- KMS uses envelope encryption to protect data
- KMS encrypts the key you create with the master key
- AWS managed service that enables you to create and manage your own encryption keys
- Allows AIM policies to be defined for granular access the encryption keys
- Located in the IAM section of the console under encryption on the left panel
- If the key has the AWS logo next to it, you cannot delete this master key
- The encrypted data key is stored with the data, but you still need the secure master key
- The Master key is safely stored in KMS and is similar to the private key in asymmetrical encryption
- Many AWS service are directly integrated into KMS for ease of setup and management
- AWS KMS creates a data key, encrypts it under a Customer Master Key (CMK), and returns plaintext and encrypted versions of the data key to you
- You use the plaintext key to encrypt data and store the encrypted key alongside the encrypted data
- You can retrieve a plaintext data key only if you have the encrypted data key and you have permission to use the corresponding master key
- Encryption context is a set of key/value pairs that you can pass to AWS KMS when you call the Encrypt, Decrypt, ReEncrypt,

GenerateDataKey, and GenerateDataKeyWithoutPlaintext APIs

- Even though the encryption context is not included in the cipher text, it is cryptographically bound to the cipher text during encryption and must be passed again when you call the Decrypt (or ReEncrypt) API
- AWS key management services uses Envelope encryption to encrypt data
- KMS creates a data key, encrypts it under a Customer Master Key (CMK) and returns plaintext and encrypted versions of the data key to you.
- Use the plaintext key to encrypt data and store the encrypted key alongside the encrypted data
- You can retrieve a plaintext data key only if you have the encrypted data key and you have permissions to use the corresponding master key
- Invalid cipher text for decryption is plaintext that has been encrypted in a different AWS account or cipher text that has been altered since it was originally encrypted
- KMS integrates with cloud trail for key logging, when, who etc.
- Keys only work in the region they were created in
- When you disable the key, you also disable access to the data the key was encrypting
- The Signature Version 4 signing process describes how to add authentication information to AWS requests
- For security, most requests to AWS must be signed with an access key (Access Key ID [AKI] and Secret Access Key [SAK]
- If you use the AWS Command Line Interface (AWS CLI) or one of the AWS Software Development Kits (SDKs), those tools automatically sign requests for you based on credentials that you specify when you configure the tools
- However, if you make direct HTTP or HTTPS calls to AWS, you must sign the requests yourself
- https://aws.amazon.com/kms/

Directory Service

- Microsoft active directory
- On solutions Architect exam
- Three types available:
  - AWS Directory Service for Microsoft Active Directory (Enterprise Edition) (also referred to as Microsoft AD),
  - Simple AD
  - AD Connector
- Simple AD is a Microsoft Active Directory-compatible directory that is powered by Samba 4
- Simple AD supports commonly used Active Directory features such as user accounts, group memberships, domain-joining Amazon Elastic Compute Cloud (Amazon EC2) instances running Linux and Microsoft Windows, Kerberos-based Single Sign-On (SSO), and group policies
- Use Simple AD if you need an inexpensive Active Directory–compatible service with the common directory features
- Directory Connector enables connecting your existing on-premises Active Directory to AWS
- With SAML-enabled single sign-on the portal first verifies the user's identity in your organization, then generates a SAML authentication response
- The portal acknowledges a SAML authentication response, then verifies the user's identity in your organization
- Simple AD is an inexpensive Active Directory–compatible service with the common directory features
- AWS Directory Service for Microsoft Active Directory (Enterprise Edition) is a feature-rich managed Microsoft Active Directory hosted on the AWS cloud
- AD Connector, connects your existing on-premises Active Directory to AWS
- https://aws.amazon.com/directoryservice/

Edge Services

- Edge services: Many locations but a single service from AWS
- Edge services describe a group of AWS product offerings
- Benefits, horizontal and vertical scalability.
- Optimize for costs to reduce data transfer charges
- Edge services includes:
    - CloudFront
    - Web application firewall (WAF)
    - Route 53
    - AWS Shield
    - Lambda

ElastiCache

- ElastiCache is an in-memory cache in the cloud
- The service improves the performance of web applications by storing and retrieving data from fast, managed, in-memory caches
- Data does not need to be retrieved entirely from slower disk-based databases
- ElastiCache is mainly used for improving the performance of web applications by caching the information that is frequently accessed for fast retrieval
- Provides very fast access to the information by using in-memory caching
- ElastiCache supports open source caching platforms like- Memcached and Redis
- Memcached supports up to 20 nodes per cluster
- No need to manage separate caching servers with ElastiCache
- Provides low latency access to applications that need this data very frequently
- Cache in RAM instead of keeping it in the disk based database for frequently accessed data
- Significantly reduces latency and increases throughput for read intensive application or compute-intensive workloads
- Caching improves application performance by storing critical pieces of data in memory for low-latency access
- Cached information may include the results of I/O intensive database queries or the results of compute heavy calculations
- Takes the load off of the database and web servers
- In the AWS console, ElastiCache is under the database section
- ElastiCache supports two open-source in-memory caching engines:
    - Memcached: A widely adopted memory object caching system.
    - ElastiCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.

  - No cross AZ capabilities
  - Memcached Supports up to 20 nodes per cluster
  - Memcached does not have a backup or snapshot function
  - Memcached is difficult to sort and rank large datasets, use Redis instead
  - Redis: Popular open-source in-memory key-value store that supports data structures such as sorted sets and lists.
  - Redis supports Master/Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy
  - Redis clusters can only support a single node, however, you can group multiple clusters together into a replication group
  - Redis engines allows for both manual and automatic snapshots
- Good for databases that are read heavy and not prone to changes
- EC2 instances must reside in the same VPC as ElastiCache
- EC2 instances bust be given permissions to access ElastiCache, in EC2 security groups, inbound, edit, type customer TCP rule, enter port number of the Cache (Redis default is 6379), look under the ElastiCache properties tab to get the port number. Also add a rule for SSH access in the same security group

- Elsasticache is easy to deploy, operate, and scale an in-memory cache in the cloud
- ElastiCache is API compatible with existing Memchached clients and does not require the application to be recompiled or linked against libraries
- Amazon ElastiCache manages the deployment of the ElastiCache binaries (it is an AWS managed service)
- To migrate from in house to ElastiCache, update the configuration file of the application with the endpoint for the ElastiCache server and configure a security group to allow access from the application servers
- ElastiCache stores the most frequently accessed data
- ElastiCache removes the complexity associated with deploying and managing a distributed cache environment, it is completely managed by AWS

- Memcached restricts API actions using AWS IAM policies and restrict network access through security groups
- When clients are configured to use auto discovery, they can discover new cached nodes as they are added and removed, this allows for ease of scalability
- Auto discovery must be enabled on each client and is not active server side
- MemCacheD
    - D means Linux Damon so the pronunciation is mem-cache-D
    - Open source code that runs in Linux as a daemon
    - Scales horizontally
    - supports parallel operations by dividing the data into shards
    - Up to 20 nodes
    - The Memchached cluster always starts up empty
    - In memory Key/Value data store
    - Use when you need cache objects such as a database, run large nodes,
    - partition across shards,
    - scale out\in by adding and removing processing nodes
- Redis:
    - Supports complex data sets, such as string lists and sets
    - Supports sorted sets, counters and hashes
    - Can save data on disk, then starts up with data and not blank like MemCached
    - Supports manual and automatic snapshots
    - Cluster is always a single node
    - Clusters get grouped together in to replication groups
    - Multi-AZ replication for fail-over and High Availability
    - Replication is asynchronous
    - Cannot scale up to larger node types
    - Cluster can be initialized from the backup
    - Use when there is complex data types, running in memory sorts and ranks,
    - backup and restore in memory data, support multiple databases

  - and run multiple read replicas, multi-AZ fail-over capability, messaging
  - pub/sub capability
- Video introduction: https://www.youtube.com/watch?v=8eD2eNljURE&feature=youtu.be
- https://aws.amazon.com/elasticache/

Web Application Firewall: WAF

- AWS managed firewall service
- Application level protection to websites
- Prevents cross site scripting and SQL injections
- Not on the CSA exam but important to be aware of
- Mitigate risk, layer 7 filtering included with AWS shield
- No changes for WAF, instant on, available globally
- https://aws.amazon.com/waf/

AWS Shield

- Aws managed DDoS (Distributed Denial of Service)protection service
- implemented at each edge location
- No central scrubbing center
- Redundant Interconnects
- Monthly fee
- Two versions:
  - Shield
  - shield advanced
- Shield advanced ads
  - DDoS response team
  - CloudWatch metric
  - AWS WAF
  - Damage protection
- https://aws.amazon.com/shield/

Artifact

- Audit and compliance portal
- On demand AW compliance reporting
- ISO, PCI compliance documents
- In security specialty certification
- https://aws.amazon.com/artifact/

AWS Management tools

Cloud Formation

- Turns infrastructure into code
- A JSON document that describes your environments via templates
- Must know this inside and out in the real world and for the exam
- Templates describe your deployment and that you can provision entire deployments based off of these templates
- It is an automated deployment scripting service
- CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion
- With AWS CloudFormation, you can reuse your template to set up your resources consistently and repeatedly
- Manages a collection of AWS services, offers provisioning and updating in an orderly manner.
- Sample templates or create your own
- AWS takes care of making all of the dependencies work
- Creates versioning of deployments
- A template and its resources are called a stack
- Cloud formation is offered at no extra charge, you pay for the services it deploys
- Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified
- You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don’t choose a role, CloudFormation uses the permissions defined in your account
- The resources tab in the AWS console shows the status and type of resources that the stack deploys
- The stack designer is a graphical representation of your deployment and features a drag and drop interface to create your stack
- Just describe your resources once and then provision the same

resources over and over in multiple stacks

- If a cloud formation deployment fails or there is any type of error, cloud formation rolls back and created resources are deleted as part of the roll back process
- This is a free service. However you are charged for the underlying services used
- http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-description-structure.html
- 2017 support added to deploy across multiple regions and accounts in specific orders of regions, called stack set
- By default, the “automatic rollback on error” feature is enabled
- This will cause all AWS resources that AWS CloudFormation created successfully for a stack up to the point where an error occurred to be deleted
- This is useful when, for example, you accidentally exceed your default limit of Elastic IP addresses, or you don’t have access to an EC2 AMI you’re trying to run
- This feature enables you to rely on the fact that stacks are either fully created, or not at all
- After a deployment has been created with Cloud Formation, it is best practice to make all changes inside of CloudFormation instead of modifying the services directly. You can version the templates to keep track of your changes
- Use the update stack tab in the console to make changes
- https://aws.amazon.com/cloudformation/
- Sample JSON to create a S3 bucket using cloud formation, use a different ID:

```
{
"AWSTemplateFormatVersion" : "2010-09-09",
"Description" : "This template creates an S3 bucket",

"Resources" : {
"S3Bucket" : {
"Type" : "AWS::S3::Bucket",
"Properties" : {
"AccessControl" : "PublicRead",
```

```
"BucketName" : "tipofthehat-2018-123456789123"
}
}
}
}
```

A template that creates a VPC:

```
AWSTemplateFormatVersion: 2010-09-09
Description: Deploy a VPC

Resources:
VPC:
Type: AWS::EC2::VPC
Properties:
   CidrBlock: 10.0.0.0/16
   EnableDnsHostnames: true
   Tags:
   - Key: Name
      Value: Lab VPC
```

The template that creates the internet gateway:

```
InternetGateway:
Type: AWS::EC2::InternetGateway
Properties:
   Tags:
   - Key: Name
      Value: Lab Internet Gateway
```

Attach the internet gateway to the VPC:

```
AttachGateway:
Type: AWS::EC2::VPCGatewayAttachment
Properties:
   VpcId: !Ref VPC
   InternetGatewayId: !Ref InternetGateway
```

Create the subnets:

```
PublicSubnet1:
Type: AWS::EC2::Subnet
```

```
Properties:
   VpcId: !Ref VPC
   CidrBlock: 10.0.0.0/24
   AvailabilityZone: !Select
      - '0'
      - !GetAZs ''
   Tags:
      - Key: Name
        Value: Public Subnet 1

PrivateSubnet1:
Type: AWS::EC2::Subnet
Properties:
   VpcId: !Ref VPC
   CidrBlock: 10.0.1.0/24
   AvailabilityZone: !Select
      - '0'
      - !GetAZs ''
   Tags:
      - Key: Name
        Value: Private Subnet 1
```

Create the route tables:

```
PublicRouteTable:
Type: AWS::EC2::RouteTable
Properties:
   VpcId: !Ref VPC
   Tags:
      - Key: Name
        Value: Public Route Table

PublicRoute:
Type: AWS::EC2::Route
Properties:
   RouteTableId: !Ref PublicRouteTable
   DestinationCidrBlock: 0.0.0.0/0
   GatewayId: !Ref InternetGateway
```

Create the subnet associations:
PublicSubnetRouteTableAssociation1:
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
SubnetId: !Ref PublicSubnet1
RouteTableId: !Ref PublicRouteTable

The cloud formation console outputs tab is configured to show information about the resources that were created:

Outputs:
VPC:
Description: VPC
Value: !Ref VPC

AZ1:
Description: Availability Zone 1
Value: !GetAtt
- PublicSubnet1
- AvailabilityZone

JSON of the template:

```
{
"AWSTemplateFormatVersion": "2010-09-09",
"Description": "Deploy a VPC",
"Resources": {
      "VPC": {
        "Type": "AWS::EC2::VPC",
        "Properties": {
            "CidrBlock": "10.0.0.0/16",
            "EnableDnsHostnames": true,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Lab VPC"
                }
            ]
        }
```

```
},
"InternetGateway": {
  "Type": "AWS::EC2::InternetGateway",
  "Properties": {
      "Tags": [
          {
              "Key": "Name",
              "Value": "Lab Internet Gateway"
          }
      ]
  }
},
"AttachGateway": {
  "Type": "AWS::EC2::VPCGatewayAttachment",
  "Properties": {
      "VpcId": {
          "Ref": "VPC"
      },
      "InternetGatewayId": {
          "Ref": "InternetGateway"
      }
  }
},
"PublicSubnet1": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
      "VpcId": {
          "Ref": "VPC"
      },
      "CidrBlock": "10.0.0.0/24",
      "AvailabilityZone": {
          "Fn::Select": [
              "0",
              {
                  "Fn::GetAZs": ""
              }
          ]
```

```
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": "Public Subnet 1"
            }
        ]
    }
},
"PrivateSubnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
        "VpcId": {
            "Ref": "VPC"
        },
        "CidrBlock": "10.0.1.0/24",
        "AvailabilityZone": {
            "Fn::Select": [
                "0",
                {
                    "Fn::GetAZs": ""
                }
            ]
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": "Private Subnet 1"
            }
        ]
    }
},
"PublicRouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {
            "Ref": "VPC"
```

```
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": "Public Route Table"
            }
        ]
    }
},
"PublicRoute": {
    "Type": "AWS::EC2::Route",
    "Properties": {
        "RouteTableId": {
            "Ref": "PublicRouteTable"
        },
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {
            "Ref": "InternetGateway"
        }
    }
},
"PublicSubnetRouteTableAssociation1": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Properties": {
        "SubnetId": {
            "Ref": "PublicSubnet1"
        },
        "RouteTableId": {
            "Ref": "PublicRouteTable"
        }
    }
},
"PrivateRouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {
            "Ref": "VPC"
```

```
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "Private Route Table"
                }
            ]
        }
      },
      "PrivateSubnetRouteTableAssociation1": {
        "Type": "AWS::EC2::SubnetRouteTableAssociation",
        "Properties": {
            "SubnetId": {
                "Ref": "PrivateSubnet1"
            },
            "RouteTableId": {
                "Ref": "PrivateRouteTable"
            }
        }
      }
},
"Outputs": {
      "VPC": {
        "Description": "VPC",
        "Value": {
            "Ref": "VPC"
        }
      },
      "AZ1": {
        "Description": "Availability Zone 1",
        "Value": {
            "Fn::GetAtt": [
                "PublicSubnet1",
                "AvailabilityZone"
            ]
        }
      }
```

```
}
  }
```

CloudTrail

- CloudTrail is an API logging service that logs all API calls made to AWS
- CloudTrail delivers log files to an S3 bucket
- Stores AWS log files created by AWS apps into a S3 bucket
- Auditing logs storage application
- AWS CloudTrail is an AWS service that records AWS API calls for your account and delivers log files to you.
- The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing
- AWS CloudTrail records important information about each API call, including the name of the API, the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS Cloud service
- Located in the Management tools section of the AWS console
- Used to track changes in AWS
- You can view the logs in the AWS CloudWatch web console in events history and also access them via the AWS command line "aws cloudtrail" help from the CLI.
- Makes API calls using the management console, AWS SDK's, AWS CLI and higher-level AWS service
- It does not matter if the API calls originate from the CLI, SDK, or console, everything gets reduced to an API call.
- Event history log retention is 7 days by default if you do not create a trail
- Enabled by default on all AWS accounts
- Records that last 90 days of activity without specifically setting up CloudTrail
- Enable CloudTrail explicitly to store account activities and events in a S3 bucket
- Logs can help address security concerns
- Allows you to log and view each action performed by users in your AWS account
- Not syslogs, that is CloudWatch, CloudTrail is if you make a

change of a configuration, it goes to cloud trail as a record of the transaction

- Cloud Trail is a global AWS service, you will have a single cloud trail for all regions (global service) or per region but not at the AZ or subnet or subnet level
- Cloud trail by default encrypts log files using SSE (server side encryption) KMS (Key management services)
- Can send SNS messages for every log file delivery
- Does not support SQS simple queueing service
- Not all services support or are able to send log files to CloudTrail, check the online documentation for the latest listing of both supported and unsupported services
- Cloud trail l is used if you need to audit events or changes, or need to look at the change records
- Cloud Trail is a AWS managed service that records AWS Application Program Interface (API) calls for your account and delivers log files to you
- Cloud Trail takes logs and stores them in a S3 bucket, however you will still need an application to look at the data
- Maybe have a EC2 instance to pull the CloudWatch data out of the S3 bucket and store it in a syslog server
- Stores all API access data to any AWS service monitored
- Per AWS account and is enabled on a per region or globally
- Can consolidate logs using an S3 bucket
- S3 buckets store logs in a region/year/month/day tree structure
- Log files are in the JSON format
- AWS CloudTrail – Monitor and audit calls made to the CloudWatch API for your account – CloudTrail can be used to enable CloudWatch to write log files to an S3 buckets for example.
- Cloud trail is logging, CloudWatch is monitoring
- It is common to integrate CloudTrail with CloudWatch
- Use the CloudTrail data in Cloudwatch to get granular SNS updates

- To receive a history of all EC2 API calls (including VPC and EBS) made on your account, you simply turn on CloudTrail in the AWS Management Console
- Use the filter box to view specific related events

- Event source allows you to select a specific source device
- Event history on the CloudTrail console allows you to download the logs in either JSON or CSV, click one download icon on the upper right hand corner of the AWS web console
- AWS CloudTrail records all of the API access events as objects in an Amazon S3 bucket that you specify at the time you enable AWS CloudTrail
- Use can use KMS (Key Management System) to add security to the logs. This is set up in the Trails tab of the console in the storage location section
- Take advantage of Amazon S3's bucket notification feature by directing Amazon S3 to publish object-created events to AWS Lambda
- Sumulogic is a commercial tool that is used to analyze log file https://www.sumologic.com/
- Whenever CloudTrail writes logs to your Amazon S3 bucket, Amazon S3 can then invoke your Lambda function by passing the Amazon S3 object-created event as a parameter
- The Lambda function code can read the log object and process the access records logged by AWS CloudTrail
- All CloudTrail documentation: https://aws.amazon.com/ddocumentation/cloudtrail

Opsworks

- A configuration management service that helps you configure and operate applications of all shapes and sizes using the Chef open source application
- AWS OpsWorks uses Chef Recipes to start new app server instances, configure application server software, and deploy applications
- Chef Recipes automate operations like software configurations, package installations, database setups, server scaling, and code deployment
- Look for the term “chef” or “recipes” or “cook books” and think OpsWorks
- Covered in detail in the Sysops certification track
- AWS OpsWorks uses Chef Recipes to start new app server instances, configure application server software, and deploy applications
- Used Chef to automate deployments
- Orchestration service that uses Chef
- Consists of recipes to maintain a consistent state
- https://www.chef.io/implementations/aws/
- https://puppet.com/products/managed-technology/aws
- https://www.ansible.com/aws
- https://docs.saltstack.com/en/latest/topics/cloud/aws.html

AWS Config: Config Manager

- AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance
- Config Manager is sued to continuously record configurations changes to Amazon RDS DB Instances, DB Subnet Groups, DB Snapshots, DB Security Groups, and Event Subscriptions and receive notification of changes through Amazon Simple Notification Service (SNS)
- AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance
- Monitors and give warnings
- Auditing for your AWS deployment
- Used to set alerts
- With AWS Config you can:
    - Discover existing and deleted AWS resources
    - Determine your overall compliance against rules,
    - Retrieve configuration details of a resource at any point in time
    - Enable compliance auditing
- https://aws.amazon.com/config/

Trusted Advisor

- AWS Trusted Advisor inspects your AWS environment and makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps
- AWS Trusted Advisor draws upon best practices learned from the aggregated operational history of serving hundreds of thousands of AWS customers
- Scans and give tips on improvement
- In the Administration and Security section of the console
- Series of recommendations
- Tips on optimization
- Tips on how to save money
- Four sections in the Trusted Advisor console:
    - Fault tolerance
    - Cost optimization
    - Security
    - Performance improvement
- Each of the four sections have detailed descriptions of the suggested best practice, action suggestions, and useful resources
- Three status check states:
    - Red: action is recommended
    - Yellow: investigation recommended
    - Green: No problems found
- Best practices optimizing costs:
    - EC2 reserved instance optimization
    - Low-utilization Ec2 instances
    - Load balancers that are idle
    - Elastic IP addresses unused or not associated
    - RDS idle databases
- Best practices security:
    - Security groups open ports and subnets
    - Security groups: unrestricted access open rules
    - IAM usage
    - S3 bucket permissions

    - Root account MFA (multifactor authentication) warns if not enabled
    - IAM password policy warn if not enabled of or requirements not enable
    - RDS security group access database security check
- Best practice Fault tolerance:
    - EBS snapshots checks for age and availability
    - Load balancer optimized validates the configuration
    - Auto scaling group resources availability of resources in the launch configuration and auto scaling groups
    - RDS Muti-AZ configured verifies that RDS instances are not in just one availability zone
    - Route 53 DNS name server delegations
    - ELB server connection draining verifies it is not enabled
- Best practice performance:
    - Looks for high utilization of EC2 instances CPU over 90% for four days
    - Service limits above 80%
    - EC2 security groups, looks for a large number of rules
    - EBS magnetic storage volumes, looks for over-utilization
    - CloudFront alternate DNS names
- https://aws.amazon.com/premiumsupport/trustedadvisor/best-practices/

Step functions

- Coordinates the components of distributed applications and micro-services using visual workflows
- An AWS service that coordinates components and step through the functions of your application
- Provides a graphical console to arrange and visualize the components of your application as a series of steps
- Step functions makes it simple to build and run multistep applications
- Automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected
- Logs the state of each step, so you can diagnose and debug problems quickly
- You can change and add steps without even writing code, so you can easily evolve your application and innovate faster
- Visualize what's going on in app, micro services being used
- Part of the AWS serverless platform
- Makes it easy to implement and orchestrate Lambda serverless applications
- Very little, if any mention on the AWS SCA exam
- https://aws.amazon.com/step-functions/

Simple Workflow Service: SWF

- AWS Simple Workflow Service (SWF) is a cloud service that makes it easy to coordinate work across distributed application components
- Expect to see this on the CSAA exam
- Simple Workflow Service is a AWS fully managed service that helps developers build, run, and scale background jobs that have parallel or sequential steps
- SWF is a fully-managed state tracker and task coordinator in the Amazon cloud
- Complete tasks in a synchronous or asynchronous fashion, provides application logic that decides what to do when tasks have been completed
- Amazon.com e-commerce site uses SWF to get your package out to you, for example, click on place order and the workflow includes charging the credit card and sending the order to the warehouse, checks to see if it is eligible for Prime, deliver to the postal area of the distribution center and send conveyer, scan, add address label, send to the shipping dock and so on
- SWF enables applications for a range of uses including:
  - media processing
  - web applications back-ends
  - business process workflows
  - analytics pipelines,
- SWF is 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
- Coordinates automated and human tasks into the workflow
- Does not revolve around EC2 instances, it can be anything in a “flow” of tasks to be completed
- SWF enables a way to coordinate tasks inside of a given framework
- Guaranteed that the workflow message will only be processed once
- The following are “actors”:

  - Activity works
  - Workflow starters
  - Deciders
- One main difference with Simple Queuing Services (SQS) is retention periods SQS is up to 14 days, SWF is up to 1 year for workflow executions
- SWF presents a task-orientated API, whereas SQS offers a message orientated API
- SWF is a way of managing a whole series of complex tasks, SQS is decoupling distributed processing tasks
- Amazon SWF ensures that a task is assigned only once and is never duplicated
- 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
- https://aws.amazon.com/swf/

SWF Actors: Workflow starters

- An application that can initiate (start) a new workflow
- This could be for example, an e-commerce website when placing an order or a mobile app or searching airline departure gates

SWF Actors: Workflow deciders

- Deciders control the flow of activity tasks in a workflow execution
- If something has finished in a workflow (or fails) a decider decides what to do next
- Example: your credit card is declined, the decider sends the order to the alternative payments page, once that clears and is accepted, the decider sends it to an activity worker for the next tasks to be completed
- Deciders control activity tasks, if this happens do this or if that happens then do something else
- Schedules the activity tasks and provides input data to the activity workers
- The decider can also process events that may arrive while the workflow is in progress
- The decider closes the workflow when the objective has been completed
- A decision task is used to communicate (back to the decider) that a given task has been completed

SWF Actors: Activity Workers

- Activity workers carry out the activity tasks
- The activity worker is a software process or thread that performs the activity tasks that are part of your workflow
- Every activity worker will poll the AWS SWF for new tasks that are appropriate for that activity worker to perform; certain tasks can be performed only by certain activity workers
- When a task is received, the activity worker will process the task to completion and then reports to Amazon SWF that the task was completed and provides the result
- The activity task represents one of the tasks that you identified in your application

API Gateway

- Not a topic for the architect associates the exam but important to be aware of
- API = Application Programmable Interface
- API gateway is a fully managed AWS service that makes it easy for developers to publish, maintain, monitor, and secure APIs at scale
- No minimum fees or startup costs
- Charged for only for the API calls you receive and the amount of data transferred out
- Allow the creation or definition of API's that acts as a software interface for applications to access data, business logic, or results from code running on AWS Lambda.
- API Gateway handles all of the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including:
    - traffic management
    - authorization and access control
    - monitoring
    - API version management
- Using the AWS management console, create a API's that acts as a front end for applications to access data, business logic, or functionality from you back-end services, such as applications running on EC2, code running on Lambda, or any web application
- Browsers or applications may make API calls to the AWS API gateway which then routes them down to Lambda or EC2 for processing
- Offers the ability to enable API caching in Amazon API Gateway to cache your endpoints response to increase response times and reduce back-end processing
- API caching allow you to reduce the number of calls made to your endpoints and also improve the latency of the requests to your API
- When you enable caching for a stage, API Gateway caches responses from the endpoint for a specified time-to-live (TTL) period measured in seconds

- Should another endpoint make the same request that just came in, the cache replies and there is no need to process it another time since it was just done, this saves Lambda and EC2 processing
- The cache TTL is a short duration, maybe 60 seconds to keep the data fresh, this also gives a faster response
- API gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint
- API gateway is a low cost and efficient AWS managed services
- All of the endpoints with API gateway are HTTPS
- Verbs supported: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS
- After creating the API, it must be deployed before it can be used
- Deploying an API involves creating a deployment and stage and associating the deployment with the stage
- Scales easily
- You can throttle request to prevent attacks
- Integration allows you to Connect API gateway up to CloudWatch to log all API activity
- The same-origin policy is an important concept in the web application security model. Under the policy, a web browser permits scripts contained in a first web page to access data in a second web page, but only of both web pages have the same origin (same domain name)
- Cross-Origin Resource Sharing (CORS) is one way the server at the other end (not the client code in the browser) can relax the same-origin policy (know this for the exam)
- Cross-Origin Resource Sharing (CORS) is a mechanism that allows restricted resources such as fonts on a web page to be requests from another domain outside the domain from which the first resource was served
- Error – “Origin policy cannot be read at the remote resource?”, Means you need to enable CORS on API gateway
- Stages enables version control for API gateway
- A Stage defines a unique base URL:({of the https://{restapi-id}.execute-api.{region}amazonaws.com/{stageName}format) for users to call the associated API snapshot

- By using different stage-deployment combinations, you can enable version control for the API
- Stage variables are name-value pairs that you define as configuration attributes associated with a deployment stage of an API
- Use stage variables to pass configuration parameters to a Lambda function through your mapping templates
- API gateway is deeply integrated with Lambda
- Cognito can be used to control the API gateway
- Cognito user pools acts as an identity provider to maintain a user directory
- Cognito supports registration and sign-in as provisioning identity tokens for signed in users
- Leverages signature version 4 to authorize access to APIs
- Managed cache to store API requests
- Reduced latency and DDoS protection is offered through CloudFront in front of API gateway
- SDK kit of iOS, Android and Javascript is available
- OpenAPI specification “Swagger” support
- Swagger is a specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services
- Swagger is a request/response data transformation
- Used to access backend services
- You can enable EC2 SSL certificates to ensure the instance only accepts requests for the API gateway
- Use API gateway to generate an SSL certificate and use its public key on the backend to verify HTTP requests to the backend systems are only from your API gateway
- SSL certificates allows the HTTP backend to control and only accept requests originating from API gateway, even if the backend is publicly accessible
- After API gateway receives a response to a request to your back end services, it caches that response in its own cache.
- When the same request comes though again, API Gateway check its cache to this request and returns that without having to check with your back end services

- The default TTL for API caching is 300 seconds, maximum is 3600 seconds, TTL=0 means that caching is disabled
- 429 error is the API gateway failing the limit-exceeding requests, you will need to change the throttling limits for the account
- On architect exam, hands on not needed, for developers
- API caching is not available on the free AWS account tier
- https://aws.amazon.com/api-gateway/

Kinesis

- In the Architect Associate exam covered at a high level, know what it is, what it does and how it can be used.
- Streaming data service into AWS
- Kinesis is a service that ingests large amounts of data into the AWS cloud.
- Amazon Kinesis is a platform on AWS to send your streaming data too
- Kinesis Streams is an AWS managed service to capture and store data. This data can then be consumed by many different AWS services
- Kinesis enables the loading and analyzing of streaming data
- Provides the ability to build custom applications to meet your business requirements
- Streaming data is generated continuously by up to thousands of remote data sources, these remote sources typically transmit data records simultaneously in small Kilobyte quantities
- An example would be purchases from an online retailer like amazon.com, stock prices data, gamer data, social network data, and geospatial data like rideshare service that is constantly telling you where your driver is and IoT (Internet of Things) sensor data
- Know for the exam the three core services and be able to identify when they are used. streams, firehose and analytics, these will be detailed in the following sections
- Amazon Kinesis is a robust ingestion platform, not a storage platform, this is to say that data is only kept for a rolling 24 hour period that give you time to move it to the service you want on the AWS platform. The data can be processed and deleted or shared with multiple AWS services such as database, applications running on EC2, big data and others
- The Kinesis services enables you to work with large data streams, ingest the data, temporarily store the data and process the data in real time
- Within the Amazon Kinesis family of services, Amazon Kinesis

Firehose saves streams to AWS storage services, while Amazon Kinesis Streams provide the ability to process the data in the stream

- Bringing data into the cloud? Kinesis was designed to do this and is a complete AWS managed service
- Commonly used to consume big data
- Streaming of large amounts of data from social media, news feeds, logs etc. into the cloud
- Once the data is in AWS, for processing large amounts of data, use Redshift for business intelligence and Elastic Map Reduce for big fata Processing
- All data is stored for 24 hours and can be increased to 7 days, this give you time to either process or move the data to its intended location in the AWS cloud
- Replay data inside of 24 hour window (default is 24 hours)
- 2017 server side encryption support added
- Kinesis is PCI-DSS compliant for credit card processing and financial services
- Kinesis allows you to work with large data streams
- There are multiple services in the Kinesis family
- Within the Amazon Kinesis family of services, Kinesis Firehose saves streams to AWS storage services, while Amazon Kinesis Streams provide the ability to process the data in the stream
- Kinesis is under the analytics section in the AWS console, all configurations are done here or via API calls
- Use available templates, on github
- It typically takes AWS 10 – 15 minutes to create the stack
- Look into outputs for the ingested data
- Kinesis front end supports Authentication/Authorization for security
- Durable across multiple AZ’s
- Kinesis synchronously replicates data across three Availability Zones in a region for durability
- Receives/Ingests any data in HTTP format
- Kinesis is an AWS managed service for streaming data ingestion, and processing
- Used to transfer the collected data into redshift and other AWS services

- Can dump stream into more than one endpoint or service
- Kinesis is real-time data ingestion on a very large scale, durable, elastic (redshift is better for batch jobs)
- Kinesis offers continuous processing, load balancing of incoming streams, fault-tolerance, checkpoint/replay
- Enables multiple processing by AWS applications in parallel
- Enables data movement into datastores/processing engines
- Kinesis is used to produce data for use by other Amazon services
- Real time is replacing batch processing, Kinesis enables this
- Producers use a PUT call to store data in a stream. Each record is <=50kb
- Put record includes User data, the streamname and PartitionKey
- A partition key is supplied by the producer and used to distribute the PUTs across shards, it is a way to distribute the streams across the shards, routes data to shards
- Kinesis MD5 hashes the supplied partition key over the hash key range of a shard
- A unique Sequence# is returned to the producer upon a successful PUT call into Kinesis
- Sharding is a type of database partitioning that separates very large databases into smaller, faster, more easily managed parts called data shards. The word shard means a small part of a whole
- Shards can be scaled up when they max out and also scale down or resize to save money, they are scalable
- Libraries on GitHub are available to build apps using Kinesis, some further programming and customization is still needed but these are good templates to start building on
- Deploy on your EC2 instances, each app uses a record processor factory, record processor and worker
- One of the main Kinesis use case is to move data into S3 for Redshft (Big data / Hadoop) processing
- Kinesis has connector code for S3, RedShift (must load into S3 first), DynamoDB with many others being added by AWS over time
- There are two pricing components with Kinesis, an hourly charge per shard and a charge for each 1 million PUT transactions
- Intro Video: https://www.youtube.com/watch?

v=MbEfiX4sMXc&feature=youtu.be

- Kinesis deep dive: https://www.youtube.com/watch?v=8u9wIC1xNt8
- Kinesis webinar: https://www.youtube.com/watch?v=FxCF34txNfk
- https://aws.amazon.com/kinesis/

Kinesis Streams

- Producers create the data at remote sites or the cloud. Some examples are sensors, EC2 instances, smartphones, laptops, applications, stock data, Internet of Things devices sending data into AWS, or any other source of data that needs to be imported, or ingested into the AWS cloud
- Stream Ingest massive amounts of data, up to terabytes per hour, from up to hundreds of thousands of streams into the AWS platform
- It can be used to perform real time analytics on streaming data
- Kinesis streams is the receiving end in the AWS platform, it will store the data for 24 hours by default and that can be increased to 7 days of retention time
- Data is processed and stored in shards
- Within 10 seconds, data put into an Amazon Kinesis stream is available for analysis
- An example would be the shard data can be processed by a fleet of EC2 instances (data consumers) that turn the data into something useful
- Then the EC2 instances can store the data in places like DynamoDB, S3 storage, Elastic Map Reduce (EMR), Redshift analytics or anywhere else and away from the Kinesis temporary storage
- Kinesis streams consist of shards, 5 transaction per second for reads, up to a maximum total data read rate of 2MB per second and up to 1,000 records per second for writes, up to a maximum total data write rate of 1MB per second (including the partition keys) by using multiple shards, the capacity is elastic and can be increased or decreased on demand
- Can have multiple shards in a stream to meet your capacity requirements
- The data capacity of your stream is a function of the number of shards that you specify for the stream
- The total capacity of the stream is the sum of the capacities of its

shards

- Streams are shards, unit of scaled data
- Each shard ingests data at 1MB/Sec and output to 1000TPS
- Each shard emits up to 2MB/Sec
- Kinesis Streams helps in creating applications that deal with streaming data
- Kinesis streams can work with data streams up to terabytes of data flow per hour into AWS
- Kinesis streams can handle data from thousands of remote sources
- Common use cases for Amazon Kinesis Streams are:
    - Real-time Analytics: events in real time such as-Big Friday sale or a major sporting events, can generate a large amount of data in a short period of time. Kinesis Streams can be used to perform real time analysis on this data, it performs analysis very quickly. Prior to Kinesis, this kind of analysis would take days. With Kenosis streams within a few minutes we can start using the results of this analysis
    - Gaming Data: In online gaming, many thousands of users play and generate a large amount of data. With Kinesis, we can use the streams of data generated to implement dynamic features based on the actions and behavior of players
    - Log and Event Data: Use Amazon Kinesis to process the large amount of Log data that is generated by different devices and sent into Kinesis
    - Create live and constantly updating dashboards, alarms, triggers based on this streaming data by using Kinesis
    - Mobile Applications: There is huge variety of data available due to the large number of parameters like-location of mobile devices, type of device, time of the day etc. Use Kinesis Streams to process the data generated by mobile applications. The output can be used by the same mobile app to enhance user experience in real time
- https://aws.amazon.com/kinesis/streams/

Kinesis Firehose

- Firehose is a fully managed AWS service
- The Amazon Kinesis family of services provides functionality to ingest large streams of data. Amazon Kinesis Firehose is specifically designed to ingest a stream and save it to S3, Redshift or Elasticsearch
- Cost is based on the amount of data sent through the service
- Firehose is similar to streams in that you have the same data producers as above (EC2, mobile phones, financial transaction services, laptops, IoT, other AWS services)
- Sends data into S3, RedShift and other services that Amazon adds over time
- Firehose is a completely automated processes and a fully manages AWS service
- After the automated process that stores the incoming data into S3 or other locations like redshift (via S3) or elastic search clusters
- No data retention windows apply to firehose, when the data arrives at firehose it is either analyzed by Lambda in real time or sent directly into S3 for storage or other locations like redshift (write to S3 first and then copy over)
- Firehose can transfer data to an elastic search cluster
- Firehose is an automated way of doing Kinesis
- The Kinesis group of services provides the AWS managed functionality to ingest large streams of data
- Firehose is specifically designed to ingest a stream and save it to S3, Redshift or Elasticsearch and other services that AWS will add over time
- Amazon Kinesis Firehose allows you to ingest massive streams of data and store the data on Amazon S3 (as well as Amazon Redshift and Amazon Elasticsearch)
- Firehose can batch, encrypt and compress data in flight through the service
- In the console, create a delivery stream, select the destinations and start streaming real time data

- Firehose takes care of stream management including scaling, sharding and monitoring
- Firehose has the ability to load the same data into multiple destinations, it is integrated with S3 and Redshift and AWS will continue to add other services that integrate with the firehose service
- In the console, you can point firehose to a S3 bucket or Redshift table, or both
- Good online demo: https://youtu.be/MMJ1T9Obw0c
- Monitoring: Services -> cloudwatch -> firehose
- https://aws.amazon.com/kinesis/data-firehose/

Kinesis Analytics

- Solutions Architect Associates exam covered in a high level concepts only
- Analyzes data in the Kinesis streams
- Run analytics on the data as it exists in streams or firehose, process it as it fly's past
- Allows you to perform SQL queries real time on the streams and then store the results in S3, Redshift or an ElasticSearch cluster
- Allows you to analyze data inside of Kinesis using SQL type languages
- https://aws.amazon.com/search/?searchQuery=kinesis+analytics

Developer tools

- This is for the developer series of AWS certification exams. For the AWS certified solutions architect exams, just be aware that these services exist and what they are used for

CodeCommit

- Github
- Place to store code in the cloud

CodeBuild

- Compile code
- Pay by minute

CodeDeploy

- Pushes to AWS services

CodePipeline

- Versioning tracker

Mobile Services

Mobile Hub

- For mobile apps development
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/mobile/

Cognito

- AWS managed identity service
- A centralized service for authentication (username and password) that allows you to use an external trusted authority instead of having to create your own username and password system for your online applications
- Sign up and sign into apps
- Amazon Cognito Identity supports public identity providers—Amazon, Facebook, and Google—as well as unauthenticated identities
- Use their user credentials as a trusted source to log into your applications
- Stores first, last name and email address
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/cognito/

Device Farm

- Encrypt devices on smart phones
- Test apps on physical devices in AWS data center
- AWS has a whole pile of devices from every conceivable manufacture and every version and model that you can possibly imagine, you can used these devices that AWS owns to test your application to validate interoperability and performance
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/device-farm/

Mobile Analytics

- Data on app usage
- AWS managed data analytics offering
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/mobileanalytics/

PinPoint

- Google analytics look alike
- Investigates what users are doing with your apps and provides usage data and reports
- PinPoint reports on topics such as where the devices are located and user behavior to highlight a few.
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/pinpoint/

Messaging

- https://aws.amazon.com/messaging/

SNS Simple Notification Services

- This is a very critical AWS services and is central to the AWS Certified Solutions Architect Associate exam
- A push messaging system that allows you to do a wide variety of processing options
- SNS allows you to group multiple recipients using topics
- SNS include publishers, subscribers and topics
- The PUBLISHER Sends a TOPIC Message to a SUBSCRIBER
- Email or text alerts, HTTP endpoints or handoff to Lambda or SQS
- SNS is In the mobile section of the AWS console and can be accessed via the command line or API calls
- Simple Notification Service (SNS) is a service that makes it easy to set up, operate, and send notifications from AWS that are triggered by events you define
- 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
- It’s an automated way of sending messages to selected service
- For example, if there is an auto-scaling event, it will trigger a SNS message that will either e-mail you or send you a text about the event
- Push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android in China with Baidu cloud push
- Push notifications to publisher and subscriber client types
- In addition to pushing cloud notifications directly to mobile devices, SNS can also deliver notifications via SMS text messages or email, to Amazon Simple Queue Service (SQS) queues, or to any HTTP endpoint
- Triggers on AWS Lambda function, Amazon Simple Queue Service (SQS) queue, HTTP endpoint, HTTPS endpoint, Email, Email-JSON, and SMS
- When you create a topic in SNS, an Amazon Resource Name (ARN) is created automatically
- Topic: The group of subscriptions that you send a message to

- Subscription: An endpoint that the message is sent and is created inside of the topic
- Publisher: The entity that triggers the sending of a message that include humans, S3 events, CloudWatch alarms and API calls
- When a message is published, it is an SNS topic that has a Lambda function subscribed to it, the Lambda function is invoked with the payload of the published message for processing
- The Lambda function receives the message payload as an input parameter and can manipulate the information in the message, publish the message to other SNS topics, or send the message to other AWS services
- 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 type. For example, you can group together iOS, Android, e-mail and SMS recipients
- When you publish once to a topic, SNS delivers appropriately formatted copies of you message to each subscriber
- To prevent messages from being lost, all messages published to Amazon SNS are stored redundantly across multiple availability zones
- Setting up SNS in the AWS console is straightforward to set up
- Email requires a response to a subscribe verification email reply before it takes effect
- SNS is commonly used with CloudWatch alarms
- Instantaneous, push-based delivery (no polling)
- No recall, when a message is delivered there is no recall feature
- 64KB per message in XML, JSON and unformatted text (non-SMS) is the maximum payload size
- 256KB Max (four requests of 64KB each), each 64KB chunk of published data is billed as one request, so an API call of 265 is billed as four requests
- pushed (active) delivery
- SNS provides message persistence
- SNS sends time critical messages to multiple subscribers though a push mechanism
- Fan-out function, one message gets sent to multiple SNS

subscribers

- A SNS topic can be recreated with a previously used topic name after about 30-60 seconds after the previous topic with the same name has been deleted. The time depends on the number of subscriptions active on the topic with a few subscribers will be available instantly for reuse while topics with larger subscribers lists taking longer
- Simple APIs and easy integration with applications
- A message cannot be deleted or recalled after it has been published to a topic
- Flexible message delivery over multiple transport protocols
- Inexpensive. Pay-as-you-go AWS managed service with no up-front costs
- Web based AWS management console offers the simplicity of a point-and-click interface
- User pays $0.50 per one million SNS requests
- $0.06 per 100,000 notification deliveries over HTTP
- $0.75 per 100 notification deliveries over SMS
- $2.00 per 100,000 notification deliveries over Email
- https://aws.amazon.com/sns/

SQS Simple Queueing Services

- Very important topic for the CSAA exam
- AWS managed distributed message queuing systems
- SQS is a web service that gives you access to a managed message queue that can be used to store messages while waiting for a computers 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 thereby decoupling the applications from each other
- This decouples the applications and acts as a buffer should the receiving process get behind in processing the data
- A queue is a temporary repository for messages that are awaiting processing
- Applications look in the SQS Queue for jobs to run, they poll the queue for waiting messages
- Decouples apps, no dependencies between the applications since there is a queue in the middle that hold messages between applications
- SQS was the first AWS service offered by Amazon
- Using Amazon SQS you can decouple the components of an application so the run independently, with SQS easing message management between components
- Any component of a distributed application can store messages in the fail-safe SQS queue
- Messages can contain up to 256KB of text in any format (JSON, XML, plain text etc.)
- Messages can be held in the queue from one minute to fourteen days, the default is four days
- Any component (typically EC2 or Lambda) can later retrieve the messages programmatically using the Amazon SQS API
- The SQS queue acts as a buffer between the component producing and saving data, and the component receiving the data for

processing

- This means the queue resolves capacity issues that arise of the producer is producing work faster than the consumer can process it, or if the producer of consumer are only intermittently connected to the network
- EC2 instances poll the message queue looking for jobs to do
- SQS guarantees message will be processed at least once
- When the EC2 instance finishes a job, it notifies the queue to delete the job and looks in the SQS queue for more jobs to do
- Messages are generated by one component to be consumed by another
- Messages can contain up to 256KB of text in any format
- SQS provides a loosely coupled system where components have been decoupled with an SQS queue
- AWS offers the first one million queueing requests per month free
- Queues act as a buffer between components which produce and receive data
- It is a good design practice to use auto scaling groups that sit behind the SQS queue and if there are a lot of messages backed up, more instances can be spun up (auto scaling) to handle the additional workload and scaled down when the workload subsides
- If an EC2 instance fails, you will not lose the job since it is still sitting in the SQS queue for another EC2 instance to pick up and run. These is a visibility period where the message is hidden while an instance processes it and comes back and deletes it. If the instance fails and does not delete it, it will become visible again for another process to take it
- A use case example would be a user who fills out a web table looking for a flight and hits submit, it goes to an EC2 instance and the app packages the request in the correct format and sends it to an SQS queue, another EC2 instance polls the SQS queue for the job that the first EC2 instance placed there. The EC2 instance polls all the travel sites and returns the results back to the first EC2 that then sends it back to the user, then the message in the queue gets deleted
- SQS is a pull based system only, it is based on polling the queue, (not a push like SNS)
- When the application polling the queue takes the message from the

SQS queue, the message is marked as “invisible” and a timer starts that is known as the “visibility timeout window”. If it does not get cleared (the EC2 server crashed), it will then become visible in the queue again for another instance to poll and complete

- The visibility timeout is the amount of time in seconds that the message is invisible in the SQS queue after a reader picks up that message
- Provided the job is processed before the visibility timeout expires, the message with then be deleted from the queue, usually the process is responsible from deleting it from the queue
- The default visibility timeout window 30 seconds
- The maximum visibility timeout value is 12 hours
- Queue URL will be required to perform any action on a SQS queue
- Amazon SQS is a distributed queuing system
- Requesting server/worker servers are used to send messages within Amazon SQS
- A requesting server sends the message to be performed, and a worker server receives the message, locks it, performs the task and then deletes the message when completed
- Amazon SQS can also be viewed as a class of temporary data storage for many classes of applications
- A dead letter queue is a queue that other (source) queues can target to send messages that for some reason could not be successfully processed
- A primary benefit of using a dead letter queue is its ability to sideline and isolate the unsuccessfully processed messages
- You can then analyze the messages sent to the dead letter queue to try and determine why they were not successfully processed, this is done offline by removing the dead messages from an active SQS queue
- Using Amazon CloudWatch, to monitor metrics for Amazon SQS and trigger an alarm when a threshold is met. This could be used to autoscale to add and remove capacity of EC2 or Lambda instances servicing the queue
- To prevent messages from being lost or becoming unavailable, all messages are stored redundantly across multiple servers and data centers

- SQS is designed to enable an unlimited number of messaging services to read and write an unlimited number of messages at any time
- SQS pricing is based on number of requests and the amount of data transferred in and out
- Optimized for horizontal scalability and is a fully manages AWS service
- When you create a new queue, you must specify a queue name that is unique within the scope of all your queues
- SQS assigns each queue you create an identifier called a queue URL that includes the queue name and other Amazon SQS components. Whenever you want to perform an action on a queue, you provide its queue URL
- When the application needs more time for processing, the invisible timeout can be changed dynamically via the ChangeMessageVisibility operation
- Valid SQS properties are Message ID that is assigned by SQS and the message body that is composed of a name/value pair and are unstructured, uninterpreted content
- If the job pulled from SQS is not processed in that time, the message will become visible again after the visibility timeout period expires and another reader will process it, this could result in the same message being delivered twice
- SQS long polling is a way to retrieve messages from the SQS queue. While regular short polling returns messages immediately, even if the message queue being polled is empty, long polling doesn't return a response until a message arrives in the message queue, or the long poll times out
- Long polling saves money since there is no response if there is no data waiting in the queue and you are not charged for a large number of pools
- Long polling allows the SQS service to wait until a message is available in a queue before sending a response, and will return all messages from all SQS services
- Long polling sits there until a message arrives and then sends the reply as soon as there is a message, it does not wait for the timeout unless there are no messages

- Long polling reduces API requests (over short polling) and saves money
- To reduce costs use long polling by supplying a WaitTimeSeconds to a value greater than 0 seconds when calling a RecieveMessage (1 to 20 seconds)
- Delay queues make messages unavailable upon arrival to the queue
- Short polling SQS samples a subset of servers and returns messages from just those servers
- Short polling will not return all possible messages in a poll
- Short polling Increases the number of API requests (over long polling) increases costs
- Longest SQS message retention period is 14 days
- Default SQS retention period is 4 days
- SQS automatically deletes messages in queue that are older than the retention period and there is no way to retrieve them
- You must provide the receipt handle for the message in order to delete it from a queue
- Autoscaling, set a trigger like 100 messages in the queue, fire up additional processes to handle the workload using AWS autoscaling services
- Cool downs work when, for example, the messages in the queue drops to one, then shut down all but one EC2 instance to save costs
- Two types of queues standard and FIFO
- Read the FAQs before the exam
- SQS uses multiple hosts, and each host holds only a portion of all the messages. When a member calls for their next message, the consumer process does not see all the hosts or all the messages. As such, messages are not necessarily delivered in the order in which they were generated, the standard SQS offering may deliver messages out of order
- If an agent abandons a message or takes a break before finishing with a message, it will be offered in the queue again. In order to ensure that no message is lost, a message will persist in the SQS queue until it is processed successfully
- To allow a user in a different AWS account permission to your SQS queue you must create a SQS policy that grants the other account access to the SQS service

- Asynchronous integration is a common pattern for loose coupling between services. It is used for interactions that do not need an immediate response and where an acknowledgement that a request has been registered will be sufficient
- Valid SQS queue URL format:
    - Use a queue URL to send, receive, and delete queue messages
    - A queue URL is constructed in the following format:
    - https://{REGION_ENDPOINT}/queue.|api-domain|/{YOUR_ACCOUNT_NUMBER}/{YOUR_QUI
    - http://sqs.us-west-1.amazonaws.com/123456789012/QueueName
- SQS automatically deletes messages that have been in a queue for more than maximum message retention period.
- The default message retention period is 4 days. However, you can set the message retention period to a value from 60 seconds to 1209600 seconds (14 days) with SetQueueAttributes.
- The max total size of all the messages that you send in a single call to SQS SendMessageBatch is 256k.
- https://aws.amazon.com/sqs/

Standard SQS queues

- Best effort order but no guarantee
- SQS standard queuing option is the default queue type
- Standard lets you have a nearly-unlimited number of transactions per second
- Standard queues guarantee that a message is delivered at least once but may be out of order or delivered more than once
- Occasionally, due to the distributed architecture that allows high throughput, more than one copy of a message might be delivered out of order
- Standard queues provide best-effort ordering which ensures that messages are generally delivered in the same order as they are sent
- If this breaks your application, you can insert sequence numbers into the SQS payload

FIFO SQS queues

- Guarantees that the first in and first out delivery and exactly once processing
- The order in which messages are sent and received is strictly preserved and a message is delivered one and remains available until a consumer processes and deletes it
- Duplicates are not introduced in the queue
- FIFO queues also support message groups that allow multiple ordered message groups within a single queue
- FIFO queues are limited to 300 transactions per second but have all the capabilities of standard queues
- You can select FIFO over standard when setting up your queue

Dead letter queues

- A holding queue for messages that could not be processed in a queue
- Sidelines and isolates the unsuccessfully processed messages
- DLQ's must reside in the same account and region as the other queues that use the dead letter queue
- Receives messages after the maximum number of processing attempts have been reached in a queue
- Just like other SQS queues, messages can be sent to it and received from it
- Create a DLQ using an API call or the SQS console
- Generally a misformed or corrupted message that could not be processed, this allows you to not have these messages remain in the processing queue and to be moved off to the DLQ for investigation

SNS/SQS Differences

- Both messaging services in AWS
- SNS is a push based service (SNS sends out messages)
- SQS pull (polling) based service that subscribers query for data
- SQS stands for Simple Queue Service
- SNS stands for Simple Notification Service
- SQS is used for implementing messaging queue solutions in an application
- Applications are decoupled in the AWS cloud by using SQS
- Since all the messages are stored redundantly in SQS, it minimizes the chance of losing any message
- SNS is used for implementing Push notifications to a large number of users
- SNS can deliver messages to Amazon SQS, E-mail, Lambda or any HTTP endpoint
- Amazon SNS is widely used in sending messages to mobile devices as well
- SNS can send SMS messages to cell phones

Business Productivity

WorkDocs

- Store work documents in AWS S3
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/workdocs/

Workmail

- Exchange like E-mail service on AWS
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/workmail/

SES Simple E-mail services

- POP and IMAP services
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/ses/

Internet of Things

- IOT managed service offerings from AWS
- Amazon series of managed services for the IoT marketspace
- Not on the AWS Certified Solutions Architect Associate exam

Desktop and App Streaming

Workspaces

- VDI Virtual desktop in cloud
- Thin client access Windows or Linux devices in the cloud
- Your Windows desktop in the cloud
- Can access from anywhere
- Alternative to having to manage a large number of staff desktop machines, just host them in AWS and access them remotely
- Think Citrix remote desktop and you got it
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/workspaces/

Appstream

- Desktop streaming apps to users
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/appstream2/

Artificial Intelligence

Alexa

- AWS voice recognition managed service and appliance
- Think of it as you talking to a Lambda instance
- Amazon offers Alexa access via its Echo and Dot products as well as many software version
- You are actually talking to LEX
- LEX is the back-end application for Alexa
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/alexa/

Polly

- Text to speech service
- Well over 50 different voices in many different languages
- The voices and languages are constantly being added to
- Creates text to MP3 into S3
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/polly/

Elastic Transcoder

- Change video format to accommodate all client video device formats
- Elastic transcoder is a fully manages AWS service
- On Associate Architect exam at a high level, you just need to be aware of this service and what it does
- Media transcoder in the cloud
- Convert media files from their original source format into different formats that will play on smartphones, tablets, PCs, Etc.
- Provides transcoding presets for popular output formats, which means that you don't need to guess about which settings work best on particular devices. Formats such as IPad, Kindle, laptops, various models of smartphones
- Charges are based on the minutes that you consume transcoding and the resolution at which you transcode
- For example, you create a MP4 file and after the creation and edit, upload it to a S3 bucket that will then trigger a Lambda function that calls the Elastic transcoder that converts the MP4 to many optimized device formats, save the converted video file back to the S3 bucket
- https://aws.amazon.com/elastictranscoder/

Machine Learning

- Data Collection on data sets and predict outcomes based on data
- Predict data based on previous performance
- Not on the AWS Certified Solutions Architect Associate exam, however you really need to be aware of this as it is a growing and important AWS service for the future

Rekognition

- Upload a picture and it breaks it all down
- Facial recognition
- Not on the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/rekognition/

Console services

- The AWS graphical user interface aws.amazon.com
- Web based interface used to manage all AWS services with a API backend to the actual services
- To log into the AWS console with AIM accounts go to aws.amazon.com and open My account (top left) and scroll down to IAM User and Role Access to billing information and click edit, then select “activate IAM access” remember your IAM accounts and log in as any that you have defined
- Save the certificate when you create your account and store in in a VERY safe place, Amazon does not keep it and if you lose it, you will be locked out of the console, this is very bad

Import/Export

VM Import/export

- VM Import/Export can import existing virtual machines as:
  - AMIs (Amazon Machine Images)
  - EC2 instances
- Can be used to import virtual machines into AWS
- Know at a general level for the AWS Certified Solutions Architect Associate exam
- https://aws.amazon.com/snowball/

Snowball

Storage Import/Export Snowball

- Snowball is a petabyte-scale data transport hardware appliance that uses secure enclosures shipped to you from Amazon to transfer large amounts of data into and out of AWS
- Using snowball addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns
- Transferring data with snowball is simple, fast, secure, and can be as little as one-fifth the cost of high-speed internet
- 50TB per snowball currently and may change over time
- Multiple layers of security, 256 bit encryption, industry standard trusted platform module for security and chain-of-custody of your data
- Tamper proof enclosure
- Only works with S3
- Rent snowball, cannot own one it is a loaner from AWS to load up your data and ship it to them, they will then transfer it into S3
- Once the data is transferred and validated, AWS performs a software erasure of the snowball appliance
- More cost effective than import/export disks which is now obsolete
- Use snowball instead of the outdated import/export service
- Multiple appliances in parallel can be used
- Snowball S3 only import and export
- Import to EBS, S3 and Glacier
- Multiple options to choose from
- https://aws.amazon.com/snowball/

Storage Gateway

- Connects an on premise software appliance with a cloud-based storage to provide seamless and secure integration between and organizations on premise IT environment and AWS's storage infrastructure
- Storage gateway enables you to securely store data to the AWS cloud for scalable and cost effective storage
- Storage gateway is a virtual appliance that runs as a VM in your data center and uploads data to AWS S3 or Glacier
- Storage Gateway with Gateway-Cached Volumes stores the most frequently accessed data on premise, and writes your other data to S3
- AWS Storage gateways server appliance is available for download as a virtual machine image that you install on a host in your data center
- Storage gateway virtual machines support either VMWare ESXi or Hyper-V
- Once installed and associated with your AWS account through an activation process, you can use the AWS management console to create the desired storage gateway option

Four options of storage gateways:

1. Gateway stored volumes
2. Gateway Cached volumes / Volumes gateway iSCSI block based (OS, VM's) virtual hard disk. Two types stored volumes (entire data set is stored on site) and cached volumes (recently accessed files cached on site and the rest is backed up in the cloud)
3. Gateway Virtual Tape libraries (VTL) backup and archive solution to create virtual backup tapes and apply lifecycle management to them.
4. File Gateway (NFS) flat files in S3 (word, pdf etc)

- https://aws.amazon.com/storagegateway/

Volume Gateway (Gateway Stored Volumes)

- Local storage with replication to S3 in the AWS cloud
- Virtual hard disk
- Keep your entire data set on site
- Storage Gateway then backs this data up asynchronously to Amazon S3 in the background
- Creates snapshots and stores them on S3 buckets
- The volume interface presents your applications with disk volumes using the iSCSI block protocol
- Data written to these volumes can be asynchronously backed up as point-in-time snapshots of your volumes, and stored in the cloud as Amazon EBS snapshots
- Snapshots use incremental backups that capture only changed blocks
- All snapshot storage is also compressed to minimize your storage space and AWS charges
- Gateway Stored volumes provide durable and inexpensive off-site backup that you can recover locally or from Amazon an Ec2 instance inside of AWS

Volume gateways

- Two types
  - Stored, entire dataset is stored on-site and is asynchronously backed up to S3 in the background
  - Cached, entire dataset is stored in S3 and the most frequently accessed data is cached on site for fast access

Volume gateways (Stored Volumes)

- Most frequently accessed data is stored locally while asynchronously backing up that data to AWS
- Stored volumes provide your on-premise application with low-latency access to their entire datasets, while providing durable, off site backups in S3
- Create storage volumes and mount them as iSCSI devices from your data center application servers
- Data written to your stored volumes is stored on your on-premise storage hardware and also asynchronously backed up to Amazon S3 in the form of Elastic Block Store (EBS) snapshots
- Application server is the iSCSI initiator and the Gateway VM is the iSCSI target that then forwards the data to local storage arrays and then does a multipart upload of the snapshots to S3
- 1GB – 16TB in size for Stored Volumes
- 100% of your data remains on site and gets incrementally backed up to S3

Volume Gateway (Cached Volumes)

- S3 is your primary data storage while retaining frequently accessed data cached locally in your on premise storage gateway
- Cached volumes minimize the need to scale your on-premise storage systems, while providing your applications with low-latency access to their frequently accessed data locally
- Create volumes up to 32TB in size and attach to them as iSCSI devices from your data center application servers
- The gateway stores the data that you write to these AWS volumes in Amazon S3 and retains recently read data in your on-premise storage gateway's cache and upload buffer storage
- 1 GB – 32 TB in size

Gateway Virtual Tape libraries (VTL)

- Tape gateway is a cost-effective solution to archive your data in the AWS cloud
- The Virtual Tape Library interface it provides lets you retain the existing tape-based backup application infrastructure to store data on virtual tape cartridges that you create on your tape gateway
- The virtual tape gateway is preconfigured with a media changer and tape drives, which are available to your existing client backup applications as iSCSI devices
- You add virtual tape cartridges as you need to archive your data
- Unlimited library of virtual tapes
- Each virtual tape library backup by S3 or a virtual tape shelf backed by Glacier
- The virtual tape library exposes an iSCSI interface to your local server
- Provides your backup application with on-line access to the virtual tapes
- Supported by NetBackup, Backup Exec, Vaem Etc.
- Get rid of your tape library systems but retain the backup applications

File Gateway

- Files are stored as objects in your S3 buckets, accessed through a network file system (NFS) mount point
- Flat files stored directly onS3 mp3, word, xls, pdf
- Ownership, permissions, and timestamps are durably stored in S3 in the user-metadata of the object associated with the file
- Once objects are transferred to S3, they can be managed as native S3 objects, and bucket policies such as versioning, lifecycle management, and cross region replication apply directly to objects stored in the bucket
- A virtual machine running in your data center that is connected to AWS over a direct connect or VPN connection to S3
- The VM can also be a EC2 instance running in a VPC at AWS
- Flat files only, word files as such

Security groups

- Console to EC2 instances and select security groups
- All outbound flows are enabled and not much inbound enabled by default, mainly port 22 SSH
- When you edit a security group, the rule takes effect immediately
- One security group can support many EC2 instances
- Stateful if there is an inbound rule then traffic is allowed outbound even if there is no outbound rule
- Best to configure the outbound rule as all
- The AWS firewall is imbedded inside the hypervisor and secures all access including VM to VM on the same hypervisor, this is essentially what security groups are, hypervisor level access control lists
- When there are multiple security groups associated with an instance, all the rules are aggregated
- Permit rules only
- Security groups are associated with network interfaces in the hypervisor
- After you launch an instance, you can change the security groups associated with the instance, which changes the security groups associated with the primary network interface (eth0).
- You can also change the security groups associated with any other network interface
- Changes can be made on the fly and take effect immediately

Creating an Amazon Machine Image: AMI

- Operating system images that run in the AWS cloud
- EC2 images are AMI based
- AMI's provide information to launch a virtual server in the Amazon cloud
- Specify an AMI when you launch and instance
- You can launch as many instances from each AMI as needed
- You can also launch instances from as many different AMI's as you need
- An AMI is a template for the root volume for the instance (for example an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- The block device mapping that specifies the storage volumes to attach to the instance once it's launched
- The EC2 section of the AWS console is where AMI's are created
- To create an AMI:
    - Go into running instance
    - Create snapshot under volumes tab
    - Give it a name
    - Actions and then create image in snapshots
    - Then under images there is an AMI tab that the EC2 instance is
- Under public AMI public images there are thousands to choose from
- You can only launch an AMI from the region that they are stored
- To launch an AMI in a different region, you must copy that AMY over to the region and then launch it
- Launch permissions, S3 bucket permissions, and user-defined tags must be copied manually to an instance based on an AMI
- User data is part of the AMI, itself, and does not need to be copied

manually

AMI types (EBS vs Instance store)

- Select your AMI based on:
  - Region
  - Operation system
  - Architecture (32 bit or 64 bit)

- Define the launch permissions
- Storage for the root device (Root Device Volume)
- Instance store (Ephemeral storage), will go away when the EC2 instance stops or is terminated
- Instance store (Ephemeral storage), very high IOPS, storage on the local server that the EC2 is running on in the AWS data center
- EBS backed volumes can be defined and data is retained when the instance is stopped or terminated (if configured to do so)
- Go to EC2 and select and image
- Cannot stop an instance store, only terminate it
- Instance store is older technology and has mostly been replaced by EBS AMI's since instance storage cannot be stopped and moved
- All AMI's are categorized as either backed by Amazon EBS backed by instance store
- For EBS volumes: The root device for an instance launched from the AMI is an Amazon EBS volume from an Amazon EBS snapshot
- For Instance store volumes: The root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3
- Instance store volumes are sometimes called Ephemeral Storage which means temporary
- Instance store volumes cannot be stopped. If the underlying host fails, you will lose your data
- EBS backed instances can be stopped. You will not lose your an EBS volume instance is stopped
- You can reboot both types and you will not lose your data
- By default, both root volumes will be deleted on termination,

however with EBS volumes, you can configure the instance to keep the root device volume (checkbox)

- Data stored on Amazon EBS is automatically replicated within an Availability Zone
- Amazon EBS volumes can be encrypted transparently to workloads on the attached instance
- Amazon EBS volumes persist when the instance is stopped
- Amazon EBS volumes can be encrypted upon creation and used by an instance in the same manner as if they were not encrypted (transparent to the operating system and applications)
- Amazon EBS There is no delay in processing when commencing a snapshot
- Amazon EBS There is no delay in restoring a snapshot

SDK Software development kits

- AWS supplied software development kits
- Supports the following programming languages:
  - Java
  - .NET
  - Node.js
  - Php
  - Python
  - Ruby
  - Go
  - C++
  - AWS mobile SDK
- https://aws.amazon.com/tools

CloudWatch

- An AWS managed monitoring service for cloud based AWS resources
- Performance monitoring is its main function
- Amazon provides a monitoring service to oversee its core resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances and many other services
- Used to monitor AWS services such as EC2
- Amazon CloudWatch metrics provide hypervisor visible metrics such as CPU utilization
- A fully managed AWS service
- With Amazon CloudWatch, you can specify parameters for a metric over a time period and configure alarms or automated actions when a threshold is reached
- Provides centralized logging and performance metrics for AWS resources
- Alarms can be used as “triggers” in AWS (i.e. to trigger an auto scaling event)
- CloudWatch can be used to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources
- Cloud watch retain metrics for 14 days (changed to 15 months 2017)
- Provides system-wide visibility into resource utilization, application performance, and operational health
- CloudWatch is a metrics depository, AWS products put metrics into the repository and you retrieve statistics based on the metrics
- Statistics can be graphically presented in the CloudWatch console
- Alarms can be generated on specific metrics and automated actions defined
- Cloudwatch options:
    - Logs: monitor and store logs generated by EC2 instances and our application in CloudWatch
    - Store the log data for a defined time period

  - Dashboard: Create visual Dashboards in the form of graphs to monitor our AWS resource
  - Alarms: Set alarms in CloudWatch
  - Alarms can notify by email or text when a specific metric crosses a threshold
  - Alarms can detect the event when an Instance starts of shuts down
  - Events: Events that are triggered by an Alarm
  - Events can take an automated action when a specific Alarm is triggered
- Cloudwatch is under the management tools section of the management console
- Monitors CPU usage and other metrics
- Option to turn on Cloudwatch when you launch an instance

- Monitor performance of your AWS deployment
- Monitors at the Hypervisor level, CPU, I/O (bytes and operations), network (bytes and operations)
- Custom metrics can be defined
- Access Amazon CloudWatch using:
  - Amazon CloudWatch Console
  - AWS CLI
  - CloudWatch API
  - AWS SDKs
- Scripts reading EC2 instance data http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html
- Does not monitor inner virtual server workings including the operating system, middleware, application and so on, you need another monitoring system specific to that operating system
- AWS does supply agents that can be installed on servers that will report metric to Cloudwatch
- Use the AWS API to get monitoring information
- Use Nagios, Zabbix, Munin or others
- http://en.clouddesignpattern.org/index.php/CDP:Monitoring_Integra
- CloudWatch cannot look inside your instance to monitor the errors in the Windows Application Log, but Amazon Cloud Watch Logs

gives you the ability to install an agent that can export otherwise internal logs into Cloud Watch

- By default basic monitoring is enabled and polls every 5 minutes for free tier and one minute for the detailed tier
- 5 Gigs of data free per month
- Detailed is charged $3.50 per instance per month one minute monitoring per instance
- In console, dashboards logs metrics, dashboard
- Tons of metrics to pick from
- Create a dashboard and then add metrics you select
- Graph or text widgets for the dashboard
- Not monitoring memory but CPU and networking
- Can monitor S3 storage buckets, DynamoDB load balancers and many other AWS services, this list is always being added to
- Add as many widgets as you want on the dashboard
- Events, you create rules to make changes such as adding URL into Route53 DNS with update CloudWatch of the change
- Alarms section, create alarm based on metrics such as CPU utilization
- Actions can include sending an e-mail, or triggering an event such as Lambda or autoscaling
- Can take action on the EC2 instance, reboot, terminate, stop, recover
- An example to actions that CloudWatch can take is to terminate and EC2 instance if the CPU is below 5% for one day for example, to save costs on load balanced servers
- Logs are generated
- Basic monitoring and detailed monitoring. Basic monitoring collects metrics at five-minute intervals, and metrics are stored for two weeks. Detailed monitoring collects metrics at one-minute intervals, and metrics are stored for two weeks
- Metric Data is stored in CloudWatch for 2 weeks
- CloudTrail is for auditing and CloudWatch is monitoring
- You can use the Amazon CloudWatch Logs agent installer on existing Amazon EC2 instances to install and configure the CloudWatch Logs Agent

- The CloudWatch agent when installed on an EC2 instance allows that instance to generate and send logs to CloudWatch
- CloudWatch supports cross account and cross region operations, greater granularity of interval times, high resolution custom metrics added for things like unused memory measured every second
- Analyze log data from CloudWatch using your own applications or solutions from AWS marketplace
- https://aws.amazon.com/cloudwatch/

Services Used with CloudWatch

- Amazon Simple Notification Service (SNS) – to send out messages to subscribing end points. For example to get email alerts when the CPU for your instances goes over 80% for a period of time
- Auto Scaling – automatically launch or terminate EC2 instances based on policies, health checks and schedules
- AWS Identity and Access Management (IAM) – enable authentication and access control for Amazon CloudWatch
- RDS databases are integrated with CloudWatch and displays default database visible metrics such as the current number of database connections
- AWS is constantly adding new services to CloudWatch, check the online documentation

Cloudwatch Custom Metrics

- While you have visibility to metrics that affect the AWS host servers themselves, you do not by default have access to instance specific metrics such as memory consumption or disk metrics visible to the Operating System
- Create custom metrics to meet your requirements
- CloudWatch supports API calls where programs and scripts can make PUT requests into CloudWatch, using name-value pairs
- CloudWatch can be used to create alarms or trigger action as required

CloudWatch Logs

- Amazon CloudWatch Logs can then be used to monitor and access log files from EC2 instances, CloudTrail and other services
- CloudWatch can archive older log files in S3 and Glacier for long term retention
- Log Agents can be installed on certain EC2 instances to automatically send log data to CloudWatch
- With CloudWatch Logs allow:

- Real time Application and System Monitoring
- Store log data for as long as needed in highly durable and cost effective storage
- Use EC2Config service to send a variety of data and log files to CloudWatch including: custom text logs, Event logs, Event Tracing (ETW) logs, and Performance Counter data.
- CloudWatch Logs agents send log data every five seconds by default and that CloudWatch Logs can ingest, aggregate and monitor any text based common log data or JSON-formatted logs
- You can retrieve any of your log data using the CloudWatch Logs console, API or through the CLI

CloudWatch Alarms

- CloudWatch Alarms can be setup to send Amazon SNS messages when an alarm is active
- An alarm agent monitors a metric over a period of time and performs one or more user defined actions depending on the value of the metric and when it crosses a threshold for a period of time specified
- A notification is then sent to an SNS topic or another endpoint such as an auto scaling policy
- Note that:
    - An alarm will invoke an action if the state of change exist for a period of time specified
    - After an alarm has been invoked, addition behaviors is determined by the type of action that was associated with the alarm
    - Alarms invoking SQS policy notifications will continue for periods that the alarm remains active
    - Alarms invoking SNS notifications are only triggered once and no additional action is invoked

An alarm can be in the following three states:

- OK
- Alarm
- Insufficient Data (check still in progress)

Cloudwatch Integration with IAM

- CloudWatch integrates with AWS Identity and Access Management (IAM) so that you can specify which CloudWatch actions a user in your AWS Account can perform
- IAM policies can be created to give only certain users in your organization permission to use **GetMetricStatistics**
- They could then use the action to retrieve data about your cloud resources
- You cannot use IAM to control access to CloudWatch data for specific resources which is to say, you can't give a user access to CloudWatch data for only a specific set of instances or a specific Load Balancer
- Permissions granted using IAM cover all the cloud resources you use with CloudWatch

- You cannot use IAM roles with the Amazon CloudWatch command line tools
- You can retrieve CloudWatch metrics using Get requests
- You can aggregate metrics across length of time etc. when using Detailed Monitoring
- Cloud Watch cannot be used to aggregate data across regions but can be used to aggregate data across Availability Zones within a Region

Cloudwatch Limitations

- AWS Accounts are limited to 5000 alarms
- CloudWatch launched extended retention of metrics in November 1, 2016. The feature enabled storage of all metrics for customers from the previous 14 days to 15 months
- CloudWatch Metrics now supports the following three retention schedules
    - 1 minute data points are available for 15 days
    - 5 minute data points are available for 63 days
    - 1 hour data points are available for 455 days

AWS Command line

- The command line interface allows you to configure, manage and monitor your AWS cloud instead of using the web based graphical user interface
- AWS command line is included in the Amazon Linux AMI by default
- Search “AWS CLI installation” in help to install on your local machine
- If you create a test user in IAM, assign it to “programmatic access”
- Then look at the users access credentials you need the “Access Key ID” & “Secret access key”
    - $ssh ec2-user@<ip address> -I MyEC2KeyPair
    - #sudo su
    - #aws <master command>
    - #aws configure
    - <asks for access key ID> then <secret access key ID>
    - <default region> us-west-2 (Oregon)
    - All done with initial configuration
    - Aws s3 ls <test by listing all S3 buckets>
    - Aws s3 help (or any service to get online help) (Control C to get out of the help)
    - Cd ~, ls, cd .aws, ls = config and credential files
- Nano credentials <- you can see your access key information if it remains stored on the instance.
- Using AIM roles are more secure than credentials
- “Aws ec2 describe-instances”, shows all EC2 instances alive and terminated
- Bootstrap Scripts
- Need to add permissions for the user in IAM s3full access ec2fullaccess etc.
- Need to install python as well for the CLI to work
- Use AWS CLI for scripting, chron, start stop backups and much more
- Use AWS CLI command reference document for all commands

AWS Support offerings

- Basic, Developer, Business and Enterprise
- Business level premium response maximum is 1 hour
- https://aws.amazon.com/premiumsupport/business-support/
- https://aws.amazon.com/premiumsupport/enterprise-support/

Well Architected Framework

- Very important to read the white papers AWS provides
- https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf

- Security
  - Protect and monitor systems
- Reliability
  - Recover from failure and mitigate disruption
- Performance Efficiency
  - Use resources sparingly
- Cost optimization
  - Eliminate unneeded expenses
- General design principles:
  - Stop guessing capacity needs
  - Test systems at production scale
  - Lower the risk of architectural change
  - Automate to make experimentation easier
  - Allow for evolutionary architectures

Well architected framework Security

- Protect information, systems, assets
- Value deliveries
  - Risk assessments
  - Mitigation strategies
- Isolate parts of the network
- Encrypt data in transit and at rest
- Enforce access control granularity using least privilege principles
- Use multifactor authentication
- Leverage AWS managed services
- Log access to resources in CloudTrail
- Automate deployments to keep security consistent
- Apply security at all layers:
  - Every virtual server
  - Every load balancer
  - Every network subnet
  - And so on S3 buckets etc.
- Enable traceability log and audit all action and changes to your environment and access to your services
- Automate responses to security events monitor and trigger responses to event driven of condition driven alerts, CloudWatch helps here
- Focus on securing you systems using the AWS shared responsibility model
  - AWS secure infrastructure and services
  - You focus on your application, data, and operating systems
  - https://aws.amazon.com/compliance/shared-responsibility-model/
- AWS Inspector is an automated security assessment tool https://aws.amazon.com/inspector/ that includes a knowledgebase with hundreds of rules

Well architected framework Reliability

- Recover from infrastructure and service failures
- Dynamically acquire computing resources on demand
- Mitigate disruptions such as:
  - Misconfigurations
  - Transient network issues

Well architected framework Performance Efficiency

- Using computing resources efficiently to meet requirements
- Maintain that efficiency as demand changes and technologies evolve

Well architected framework Performance Cost Optimization

- Avoid or eliminate
  - Unneeded costs
  - Suboptimal resources

Appendix AWS links: updates, pdf’s all AWS internals

AWS White papers for AWS-CSA Exam Prep:

- http://media.amazonwebservices.com/AWS_Cloud_Best_Practices.
- http://media.amazonwebservices.com/AWS_Security_Best_Practice
- http://d0.awsstatic.com/whitepapers/Security/AWS%20Security%20
- http://media.amazonwebservices.com/AWS_Storage_Options.pdf
- http://media.amazonwebservices.com/AWS_Cloud_Architectures.p
- http://media.amazonwebservices.com/AWS_Development_Test_En
- https://d0.awsstatic.com/whitepapers/Backup_Archive_and_Restore
- http://media.amazonwebservices.com/AWS_Amazon_VPC_Connec
- http://d0.awsstatic.com/whitepapers/aws_pricing_overview.pdf
- https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf
- https://aws.amazon.com/getting-started/tutorials/
- https://aws.amazon.com/architecture/
- http://docs.aws.amazon.com/general/latest/gr/aws_service_limits.ht

AWS Blogs and presentations:

https://Aws.amazon.com/blogs/aws

https://aws.amazon.com/blogs/aws/category/week-in-review/ Amazon Monthly webinar series: https://aws.amazon.com/about-aws/events/monthlywebinarseries/

https://aws.amazon.com/answers/ Instructional documents on AWS, internally created, very informational.

https://aws.amazon.com/getting-started/

https://aws.amazon.com/architecture/

https://awsdevops.io/

Great design page that gives examples for many different AWS architectures: http://en.clouddesignpattern.org/index.php/Main_Page

DNS lookup and web site for network testing: https://networking.ringofsaturn.com

AWS Github site: https://github.com/awsLABS/

Scripts:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html
Linux load of an apache web server:
#! /bin/bash
yum update –y (applies all the security patches)
yum install https –yes (this installs the apache web server)
Service httpd start (this starts the service)
Chkconfig httpd on (this makes sure that the HTTPD starts on reboot)

#! /bin/sh
Yum –y install httpd
Chkconfig httpd on
/etc/init.d/httpd start

Cd /var/www/html (web content directory)
Ls (notice blank directory)
Nano index.html (create the index file)
<html><h1> Hello everyone!</h1></html> (Save the text file control X in nano)

```
#!/bin/bash
yum update -y
yum install -y httpd24 php56 mysql55-server php56-mysqlnd
service httpd start
chkconfig httpd on
groupadd www
usermod -a -G www ec2-user
chown -R root:www /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} +
find /var/www -type f -exec chmod 0664 {} +
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Windows Web Server script in load web services:
<powershell>

Import-Module ServerManager
Install-WindowsFeature web-server, web-server
Install-WindowsFeature web-mgmt-tools
</powershell>

- or -

<powershell> Install-WindowsFeature Web-Server -IncludeManagementTools -IncludeAllSubFeature
</powershell>

- Go to Windows Server manager and select “Add roles and features”, select “Installation type” on the left pane, select “Role based or feature based installation” click next, select the server you want to install the web server onto, then on the left pane, click on server roles and select “Web Server” role. Click Next and access the .NET Framework installation. Click next a number of time until you see “install” and then click to install IIS on the server.

- Now, under server manager on the left pane you will see IIS. Rich click on your server and select Information Services (IIS) Manager. Under sties, select “default web site” (left pane), right click, go to “manage” and select “browse”. You should see the servers default web page come up in a browser to verify operations. The web content will be here: %SystemDrive%\inetpub\wwwroot.
- In the IIS console, select this web server and scroll down to Description and copy the Public DNS name into your browser. You should be able to access the web page over the internet.

- Accessing Windows EC2 metadata:

- Open the RDP application
- Go to IIS manager / default web site home
- Left pane select “explore”

- File explorer opens showing the WWWroot directory and show you the web server files
- Right click on iisstart and open with notepad
- Erase the existing config and add this:

```
<html>
<body>
<h2>EC2 Instance Metadata</h2>
<a href="http://169.254.169.254/latest/meta-data/">Instance Metadata</a><BR/>
<a href="http://169.254.169.254/latest/meta-data/hostname">Instance Hostname</a><BR/>
<a href="http://169.254.169.254/latest/meta-data/public-ipv4">Instance Public IP Address</a><BR/>
<a href="http://169.254.169.254/latest/meta-data/placement/availability-zone">Instance Availability Zone</a> <BR/>
<a href="http://169.254.169.254/latest/user-data">Instance User Data</a>
</body>
</html>
```

- Save the file
- Open in Internet explorer, http://localhost
- And see the instance metadata in your browser:

EC2 Instance Metadata

Instance data from the local server:
http://169.254.169.254/latest/user-data
http://169.254.169.254/latest/meta-data

$curl http://169.254.169.254/latest/user-data
$GET http://169.254.169.254/latest/user-data

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html

Putty access to an EC2 instance

Get the DNS link for PuTTY

Ssh ec2-user@x.x.x.x –I MyEc2Key.pem

Chmod 600 MyEc2Key.pem

Sudo su (on the instance)

Yum update –y

Amazon AMI has the install utilities already installed so all we need to do for the Amazon AMI EC2 instance is

#sudo mount –t nfs4 $(curl https://169.x.x.x/latest/metadata/placement/availability-zone.fs-xxxxxx/efs/us-west-2.amazonaws.com:/ /var/www/html

From the server we can now access the EFS volume #cd /var/www/html #pwd

Make a filesystem on an ECB volume:

#lsblk
#mkfs –t ext4 /dev/xvdb
#mkdir t/tipofthehat
#mount /dev/xvdb /tipofthehat
#lsblk
#cd /tipofthehat
#Ls
#nano mytest.html
#umount /dev/xvdb (data remains in the volume even when not mounted)

To check files on an UNMOUNTED volume:
#file –s /dev/xvdf

Install a web server on EC2:

- Sudo su
- Yum install httpd –y
- Service httpd status
- Service httpd start
- Will not start up on restart unless:
- Chkconfig httpd on
- Cd /var/www/html (content)
- Nano index.html
- Hello Cloud gurus
- Control x and yes to save
- Ls will show the file
- Go to your local browser and type the public IP address
- Make sure the security group allows http access
- Security does not allow deny, it only supports permits
- State full if you allow a port in it automatically is allowed back out even though there is not outbound rule for it
- Best to leave all outbound traffic out

PuTTY tutorial

- PuTTY does not support the AWS PEM key files and you must use PuTTY KeyGen to convert the pem key file into the ppk file format that PuTTY uses
- Need Puttygen.exe download in addition to Putty.exe
- Get PuTTY KeyGen or puTTYgen generates RSA keys
- https://winscp.net/eng/docs/ui_puttygen
- http://www.putty.org
- Launch your EC2 instance
- Open PuTTYGen and load key. Select “all files” changes the AWS EC2 private key *.pem format to *.ppk files that PuTTY needs
- When you load the *.pem file into PuTTY KeyGen you will see nothing when it opens, select “all files (*.*) to see the file you downloaded when creating the EC2 instance.
- Sure save private key command. (Do not use Generate) just save private key.
- You can but it is not needed to select a passphrase. Password to open the key.
- Click “SAVE PRIVATE KEY” (this in important) to generate the needed *.ppk private key
- Save as a *.ppk file (click icon)
- Open PuTTY
- Host IP address in the opening PuTTY menu: ec-user@52.51.120.70 on SSH port 22
- Then load the ppk key by clicking on the SSH section on the left of the PuTTY menu and clicking on the AUTH tab
- Select “private key” and browse to your *.ppk file
- After the key is generate, “save as” and rename the extension to *.ppk
- Save the session in the main screen
- Open PuTTY
- Get the IP address from the AWS EC2 server information screen. Copy/Paste into PuTTY
- In the PuTTY host block to connect to “ec2-user@<IP address of

Ec2 instance>

- On the left pane, go to SSH -> Auth in the lower part of the menu
- Enter your newly created Private Key. By clicking on “Browse” in the Private Key file for authentication block at the bottom of the dialog box. Leave everything else at default
- Now connect and accept the warning about the key and you should be logged into the EC2 instance
- Ec2-user with no password is the default login
- IAM security must let SSH incoming

Browser troubleshooting utilities

- Hit F12 to pull up the browser troubleshooting utilities and analyzer
- Developer.mozill.org is a great reference page for mime types and all browser related data

HTML5 sample web page downloads

- http://Freehtml5.co
- http://Html5up.net

Bash Scripting

- #!/bin/bash

Windows Bash Scripting

- Install bash on windows (Ubuntu image) https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/
- Windows bash: Type “Bash”in search window
- Works better than PuTTY?
- No SUDO, you are in root
- You’ll need to use the apt-get command to install and update the Ubuntu
- Search for Available Packages: sudo apt-cache search word (Replace “word” with a word you want to search package names and descriptions for.)
- Download and Install the Latest Versions of Your Installed Packages: sudo apt-get upgrade
- he best solution would either to disable the SSH Broker for Windows 10
- https://stackoverflow.com/questions/38789354/ssh-remote-access-on-bash-windows-10

Installing the apache webserver

- Lab to install apache, update and create a web page
- Text editor
- Type “Hello” save as HTML (index.html)
- AWS console
- Advanced details in the EC2 instance
- #!/bin/bash
- Yum install httpd –y (installs apache web server
-
- Yum update –y
- Aws s3 cp s3://<bucketname>/index.html /var/www.html
- Service httpd start
- Chkconfig httpd on
- Next and add storage and add security group
- Use Bash scripts to move data around to bring up complete servers
- #!/bin/bash
- yum install httpd -y
- yum update –y (adds all patches)
- aws s3 cp s3://YOURBUCKETNAMEHERE/index.html /var/www/html/ --recursive
- service httpd start
- chkconfig httpd on

Drawing and documentation applications for AWS

- www.draw.io lets you create AWS diagrams, similar to Visio
- https://aws.amazon.com/architecture/icons/ Great site with icons and links
- www.creately.com great site has templates for $49.00/year
- www.lucidchart.com/pages/aws AWS network mapping tools, pulls configs and creates a map
- https://cloudcraft.co free entry level accounts
- https://sal.dcsolutionfactory.com looks expensive
- http://www.visualops.io/ amazing site, you drag and drop and it builds the AWS configuration and deploys