AWS Certified Solutions Architect - Associate Notes 2018-19

Curated documentation/study notes on going through the [Udemy, Certified Solutions Architect - Associate 2018] (https://www.udemy.com/AWS-certified-solutions-architect-associate/) course. These notes are to help myself, as well as, anyone else going through this same course study and prepare for the exam. Thanks!

My Study Practice

While planning out my study schedule, I felt it would be necessary to watch one section a week, starting with Section 2: 1000 ft overview, followed up by 6 days of review (1 - 2 hours a day) solidifying everything I learned. This seems to work for me but might be either too slow or fast for someone else. That's OK! Learn at your own pace until you feel comfortable with the concepts, practices and information given!

The Exam Blueprint

AWS has recently released the updated/new version of the AWS Certified Solutions Architect - Associate exam (released February 2018). The old associate exam will no longer be available starting August 12, 2018.

Let's have a look at the details of the exam...

New Exam

Generally easier than previous exam. Across 5 different domains.

Objective	Weighting
Design Resilient Architectures	34%
Define Performant Architectures	24%
Specify Secure Applications and Architectures	26%
Design Cost-Optimized Architectures	10%
Define Operationally-Excellent Architectures	6%

_Details about this exam:

- 130 minutes in length
- 65 questions
- $150 USD
- Multiple choice
- Pass mass based on bell curve
- Aim for 70%
- Qualification is valid for 2 years
- Scenario based questions

Have a look at the [Certified Solutions Architect - Associate homepage] (https://AWS.amazon.com/certification/certified-solutions-architect-associate/) to get an in-depth look at what to expect for your exam!

Table of Contents

1,000 ft Overview

Part 1. Regions, Availability Zones (AZ), Edge Locations

Regions

AWS Region is a physical, geographical area or location, consisting of 2 or more Availability Zones.

Current regions across the world:
- US East (N. Virginia) - `us-east-1`
- US East (Ohio) - `us-east-2`
- US West (Northern California) - `us-west-1`
- US West (Oregon) - `us-west-2`
- Canada (Central) - `ca-central-1`
- EU (Frankfurt) - `eu-central-1`
- EU (Ireland) - `eu-west-1`
- EU (London) - `eu-west-2`
- EU (Paris) - `eu-west-3`
- Asia Pacific (Tokyo) - `ap-northeast-1`
- Asia Pacific (Seoul) - `ap-northeast-2`
- Asia Pacific (Osaka-Local) - `ap-northeast-3`
- Asia Pacific (Singapore) - `ap-southeast-1`
- Asia Pacific (Sydney) - `ap-southeast-2`
- Asia Pacific (Mumbai) - `ap-south-1`
- South America (Sao Paulo) - `sa-east-1`

Availability Zones (AZ)

AWS Availability Zones are one or more discrete data centers, each with redundant power, networking and connectivity housed in separate facilities. Deploying your application across multiple Availability Zones is useful for redundancy, low latency and fault tolerance.

_Regions with multiple Availability Zones:

- US East

  - Ohio (3)
  - North Virginia (6)
- US West
  - Oregon (3)
  - Northern California (3)
- Canada
  - Central (3)
- South America
  - Sao Paulo (3)
- Europe
  - Ireland (3)
  - Frankfurt (3)
  - London (3)
  - Paris (3)
- Asia Pacific
  - Singapore (3)
  - Seoul (2)
  - Tokyo (4)
  - Mumbai (2)
  - Sydney (3)
  - Beijing (2)
  - Ningxia (2)

Edge Locations

AWS Edge Locations are locations around the world meant for caching content, enhancing the user experience, reducing latency. Edge locations are specifically used by AWS Cloudfront and AWS CDN. Every Region is has its own set Availability Zone's and Edge Locations.

Part 2. AWS Services Overview

Compute:

EC2 - elastic compute cloud
EC2 Container Services - containerization docker
Elastic Beanstalk - plug and play - for developers

Lambda (server less) - code/functions uploaded to the cloud to run at different points
Lightsail - plug and play
Batch - batch computing in the cloud

Storage:
S3 - simple storage service - object based storage - buckets
EFS - elastic file system
Glacier - data archival
Snowball - large amounts of data to AWS data center
Storage gateway - VM installed in datacenter or office - replicate info to S3

Databases:
RDS - relation database service - postgres, mysql, oracle
DynamoDB - non-relational db
ElasticCache - cache things from db
Redshift - data warehousing business intelligence, complex queries

Migration:
AWS Migration Hub - tracking service for moving to AWS
Application Discover Service - track applications and dependency
Database Migration Service - migrate db from on premise to AWS
Server Migration Service - migrate server to AWS cloud
Snowball - in between storage and migration

Networking and Content Delivery:
VPC (highlight) - Amazon virtual private cloud - virtual datacenter - configure avail zones, firewall, network acl etc.
Cloudfront - AWS content delivery network, store assets specific regions around the world
Route 53 - AWS DNS service - lookup ip to get ipv4 and ipv6 address
API Gateway - Serverless way of creating own api
Direct Connect - Dedicated line from office directly into amazon, connects to VPC

Developer Tools:
Codestart - project management, CI toolchain, collaborate

Codecommit - store code, like GitHub
Codebuild - compile and run tests, produce package
Code deploy - deployment service to ec2 instance
Codepipeline - automate and visualize steps to release software
X-ray - debug and analyze server less application
Cloud9 - IDE environment in browser

Part 3. AWS Services Overview (Continued)

Management tools:
Cloudwatch - Monitoring service
Cloudformation - solutions architect specific - scripting infrastructure - turn infrastructure to code
Cloudtrail - log changes to AWS environment
Config - monitors config of AWS environment
Opswork - similar to elastic beanstalk - chef and puppet to automate environments
Service Catalog - manage a catalog of IT services
Systems manager - interface for managing AWS resources - group resources
Trusted Advisor - advice around security, advice for AWS services and resources, accountant like
Managed Services - manage service for AWS cloud

Recap for exam - cloudformation, cloudtrail, cloudtrail, trusted advisor

Media Services:
Elastic transcoder - takes media and resizes on different devices
Media convert - file based video transcoding with broadcast grade features
Media live - broadcast grade live video processing service. tv internet connected multiscreen
Media Package - protect content over internet
Media Store - media storage, optimized for media
Media Tailor - target advertising into video streams without harming broadcast

Machine Learning:
Sage maker - easy for deep learning when coding for environment
Comprehend - sentiment analysis on products. good or bad?

Deep lens - computer vision on camera, recognition, physical piece of hardware
Lex - powers alexa, AI
Machine Learning - throw dataset to AWS cloud and predict outcome
Polly - text to speech, voices sound real, accents
Rekognition - upload file, tells you what is in the file
Amazon translate - translate to other langs
Amazon transcribe - hard of hearing, speech recognition, speech to text

Analytics:
Athena - SQL queries ins S3 buckets, serverless
EMR - elastic map reduce - processing large amounts of data, chops data up for analysis
Cloudsearch - search service
Elastic Search service - search service
Kinesis - solutions architect highlight, ingesting large amounts data
Kinesis Video streams - ingesting streams and analyze
Quicksight - business intelligence tool
Datapipeline - moving data between different services
Glue - ETL (extract transform load)

Part 4. AWS Services Overview (Continued)

Security Identity and Compliance:
IAM - identity access management
Cognito - device authentication, oath, after authenticated, use AWS services
Guard Duty - monitor for malicious activity
Inspector - install on vm or instances, test against it, schedule
Macie - Scan s3 buckets and looks for sensitive info and alert
Certificate Manager - ssl cert for free, manage ssl cert
Cloud HSM - cloud hardware security module - dedicate bits of hardware to store keys to authenticated
Directory Service - integration ms active service to AWS services
WAF - web application firewall - at application layer to stop attacks, XSS, sql injection
Shield - by default for cloud front - ddos mitigation, prevent ddos attacks
Artifact - portal to download AWS client reports, manage agreements

Key security services for exam: IAM, inspector, cloudHMS, directory services, waf, shield, cert manager

Mobile Services:

Mobile hub - management console for mobile app for AWS services
AWS Pinpoint - targeted push notifications
AWS Appsync - atomically updates data in web or mobile in real time
Device Farm - test apps on real device, iOS, android
Mobile Analytics - analytics service for mobile

AR/VR:

Sumerian - tools to create environment, super new

Application Integration:

Step functions - manage lambda functions and ways to go through it
Amazon MQ - message queue
SNS - notification services
SQS - decouple infrastructure, queue
SWF - workflow job creation

Customer Engagement:

Connect - contact center as a service, call center
Simple Email Service - email service, send grid, mailchimp

Business Productivity:

Alexa for business - manager for business needs
Amazon chime - google hangouts like
Work Docs - dropbox for AWS
Work Mail - Office 365 like

Desktop and App streaming:

Workspaces - VDI solution, run OS in AWS cloud
App stream 2.0 - streaming application to desktop of device

IOT:

iOT - devices sending sensor information
iOT Device Management - device management

Amazon FreeRTOS - OS for microcontrollers
Greengrass - ??

Game Development:
Gamelift - service to develop games

What Services Will Be Tested On The Exam??

Analytics
Management Tools
Migration
Compute
AWS Global infrastructure
Storage
Databases
Network and Content delivery
Security and Identity compliance
Application Integration
Desktop and App streaming

Links

- https://docs.AWS.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

- https://www.linuxnix.com/amazon-AWS-regions-vs-availability-zones-vs-edge-locations-vs-data-centers/

IAM - Identity Access Management

What is IAM?

Allow you to manage users and their level of access management to the AWS console. Tested for exam and co. AWS account in real life. IAM is globally available and not specified to region

What can you do with IAM?

- Centralized control of your AWS account
- Shared Access to your AWS account
- Granular permissions
- Identity Federation
  - Access to 3rd party service, Active Directory, Facebook, LinkedIn
- Multifactor Authentication (MFA)
- Provide temporary access for users/devices and services where necessary
- Set up and manage password rotation
- Integrates with many different AWS services
- Supports PCI, DSS compliance

Terminology

- Users - End users (people)
- Groups - Collection of users under one set of permissions
- Roles - Permissions defined for AWS resources (i.e. EC2 etc.)
- Policy Documents - Document that defines one or more permissions - JSON format
- Root account - user used to sign into AWS account

General Notes

- Universal. Does not apply to regions at this time.
- Attach permissions to users as well as groups
- New users have NO permissions when first created
- New users are assigned and Access Key ID and Secret Key when first created
  - Keys are not the same as passwords

  - Must regenerate keys if lost
- ALWAYS setup multifactor AUTH on root account
- Customize password rotation policies
- Unable to set billing alarm in cloud watch because of new account

Links

- https://AWS.amazon.com/iam/faqs/
-
https://docs.AWS.amazon.com/IAM/latest/UserGuide/introduction.html

AWS Object Storage & CDN

S3 is a safe place to store your static files being one the oldest services of AWS. It is an object-based storage where your data is spread across multiple devices.

S3 allows you to upload, where files can be from 0 bytes to 5TB. If an upload is successful, you will receive an HTTP status code of `200`.
It is capable of unlimited storage. All files are stored into 'Buckets' which is basically an S3 term for folders.

S3 uses a universal namespace meaning all names must be _globally_ unique.

_Example S3 URL:

`https://s3-eu-west-1.amazonAWS.com/[bucket-name] `

Data Consistency

S3 maintains _Read After Write_ consistency for PUTS of new objects. Meaning, as soon a new object is uploaded or written, it is available to read/view.

When performing overwrite PUTS and DELETES, these updated and/or deleted objects can take time to propagate because, also known as _Eventual Consistency_. These types of updates are known as _Atomic_ - fetching these resources could be old or new.

S3 Object - Key, Value Store

- Key - Name of object to be stored
- Value - Data being stored - made up of a sequence of bytes
- Version ID - Version signifier
- Metadata - Data about the data you are storing - date stored, size,
- Sub resource
  - Access Control Lists
  - Torrents

S3 Basics

- Built for 99.99% availability for the S3 platform
- Amazon guarantee 99.9% availability - always available
- Amazon guarantees 99.99999999999% (11, 9’s) durability for S3 information
- Tiered storage
- Lifecycle management
- Versioning
- Encryption
- Secure data using Access Control Lists bucket policies

Storage Tiers

- S3 (Normal)
    - 99.99% availability, 99. (11 9’s)
    - durable, reliable - stored redundantly across multiple devices in multiple facilities and is designed to sustain the loss of 2 facilities concurrently

- S3 IA (Infrequent Access)
    - Used for data that is accessed less frequently but requires rapid access when needed
    - Lower fee than S3 but, are charged a retrieval fee

- S3 Reduces Redundancy Storage (RRS)
    - Designed to provide 99.99% durability and 99.99% availability of objects over a given year.
- Glacier (Separate product from S3)
    - Very cost effective but used for data archival only
    - Generally, takes 3 - 5 hours to restore from glacier
  - Stores data for as low as .01G a month
  - Optimized for data that is infrequently accessed and for which retrieval times of 3 to 5 hours are suitable (slow retrieval).

S3 Charges

- Storage
- Requests
- Storage Management Pricing
- Data transfer pricing

- Transfer Acceleration

Transfer Acceleration

- Enables fast, easy and secure transfers of files over long distances between you and your end users and an S3 bucket.
- Takes advantage of AWS CloudFront global, distributed edge locations.
- When data arrives at an edge location, it is then routed to Amazon S3 over an optimized network path.

S3 Encryption and Security

By default, all newly created buckets are PRIVATE. You need to manually change permissions to access resources.

You can set policies and permissions using either Access Control Lists or Bucket Policies.

You have the ability to make a bucket private but all certain objects in that bucket to be public.

Logging

S3 buckets can be configured to create access logs which log all requests made to that bucket. This can be done to another bucket through cross account access.

Encryption

4 different methods and 2 types of encryption for S3 buckets.

1. In Transit - from client uploading to S3 bucket.
    - Using SSL/TLS encryption. HTTPS

2. At Rest
    - Server Side Encryption

- SSE-S3 - S3 Managed key. Each object is encrypted with a unique key employing strong multi-factor encryption with rotating master key (AES-256 encryption).
- SSE KMS - AWS Key Management Service, Managed Keys. Similar to SSE-S3. Separate permissions for envelope key - key that protects data encryption key. Audit trail - when keys were used and who were using.
- SSE-C - Server-Side Encryption with Customer Provided Keys. You manage encryption key.

- Client-Side Encryption
    - Encrypt data on client side and upload to S3

Links

- https://AWS.amazon.com/s3/
- https://docs.AWS.amazon.com/AmazonS3/latest/dev/Welcome.html
- https://AWS.amazon.com/s3/faqs/
- https://AWS.amazon.com/s3/storage-classes/
- https://AWS.amazon.com/glacier/faqs/

Storage Gateway

Understand at theoretical level

What is Storage Gateway?

A Service that connects an on-premise software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premise IT environment and AWS's storage infrastructure.

The service enables you to securely store data to AWS cloud for scalable and cost-effective storage. Replicates your data to specifically S3 bucket.

Downloaded as virtual machine (VM) that you install on a host in your datacenter. Storage Gateway supports either VMware ESXi or MS Hyper-V. Once you've installed your gateway and associate with AWS account through activation process, you can use the AWS Management Console to create the storage gateway option this is right for you.

Four Types of Gateway Storage

File Gateway (NFS)

Store flat files in S3 through a Network File System (NFS) mount point. 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 your bucket.

Volumes Gateway (iSCSI)

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 AWS EBS (Elastic Block Store - VM) snapshots.

Snapshots are incremental backups that capture only the changed blocks. All snapshot storage is also compressed to minimize your storage charges.

NOTE: iSCSI is block based storage. Store OS, DB's. Think of as virtual hard disk

Stored Volumes

Stored volumes let you store your primary data locally, while asynchronously backing up that data to AWS. Stored volumes provide your on-premise applications with low-latency access to their entire datasets, while providing durable, off-site backups.

You can create storage volumes and mount them as iSCSI devices from your on-premises application servers. Data written to your stored volumes is stored on your on-premises storage hardware.

This data is asynchronously backed up to S3 in the form of AWS EBS (Elastic Block Store) snapshots 1 GB - 16 TB in size for Stored Volumes.

Cached Volumes

Cached volumes let you use S3 as your primary data storage while retaking frequently accessed data locally in your storage gateway.

Cached volumes minimize the need to scale your on-premise storage infrastructure, while still providing your applications with low-latency access to their frequently accessed data.

You can create storage volumes up to 32Tb in size and attach to them as iSCSI devices from your on-premises application servers. Your gateway stores data that you write to these volumes in S3 and retains recently read data in your on-premises storage gateways cache and upload buffer storage. 1 GB - 32 TB size for cached volumes.

Volume Gateway takes virtual hard disks that are on premise and back them up to AWS_

Tape Gateway (VTL)

Offers a durable, cost-effective solution to archive your data in AWS cloud. The VTL interface it provides lets you leverage your existing tape-based backup application infrastructure to store data on virtual tape cartridges that you create on your tape gateway.

Each tape gateway is preconfigured with a media changer and tape drivers, which are available to your existing client backup applications as iSCSI devices. You add tape cartridges as you need to archive your data. Supported by Netbackup, Backup Exec, Veeam etc.

Tips (Summary)

- File Gateway - For flat files, stored directly to S3
- Volume Gateway:
  - Stored Volumes - Entire dataset is stored on site and is asynchronously backed up to S3
  - Cached Volumes - Entire dataset is stored on S3 and the most frequently accessed data is cached on site.
- Gateway Virtual Tape Library (VTL)
  - Used for backup and uses popular backup applications like NetBackup, Backup Exec, Veeam etc.

Links

- https://AWS.amazon.com/storagegateway/faqs/
- [https://AWS.amazon.com/blogs/AWS/the-AWS-storage-gateway-integrate-your-existing-on-premises-applications-with-AWS-cloud-storage/](https://AWS.amazon.com/blogs/AWS/the-AWS-storage-gateway-integrate-your-existing-on-premises-applications-with-AWS-cloud-storage/

CDN (Content Delivery Network)

What's a CDN?

A system of distributed servers that deliver webpages and other content to a user based on the geographic locations of that user, the origin of the webpage and a content delivery server

CloudFront

CloudFront can be used to deliver your entire website, including dynamic content, static, streaming and interactive content using a global network of edge locations.

Requests for your content are atomically routed to the nearest Edge Location, so content is delivered with the best possible performance.

CloudFront is optimized to work with other Amazon Web Services like S3, EC2, Elastic Load Balancing and Route 53. CloudFront also works seamlessly with any non-AWS origin server which stores the original, definitive versions of your files.

Key Terminology

- Edge Location - Location where content will be cached. Separate to and AWS region (See [1000-ft-overview/Edge-locations](https://github.com/NigelEarle/AWS-CSA-Notes-2018/tree/master/1000-ft-overviewedge-locations))
- Origin - This is the origin of all the files that the CDN will distribute. Can be S3 bucket, EC2 instance, Elastic Load Balancer or Route 53.
- Distribution - Given name of CDN which consists of a collection of Edge Locations
- Web Distribution - Typically used for websites
- RTMP (Real Time Message Protocol) - Used for media streaming

Snowball

What are the different types of Snowballs available?

Snowball

Petabyte-scale data transport solution that uses secure appliances 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 1/5 the cost of high-speed internet.

80TB Snowballs are available in all regions. Snowball uses multiple layers of security designed to protect your data including tamper-resistant enclosures, 256-bit encryption, and an industry-standard Trusted Platform Module (TPM) designed to ensure both security and full chain-of-custody of your data.

Once the data transfer job has been processed and verified, AWS performs a software erase of the Snowball appliance

Snowball Edge

Snowball Edge is a 100TB data transfer device with on-board storage and compute capabilities. You can use Snowball Edge to move large amounts of data into and out of AWS, as a temporary storage tier for large datasets, or to support local workloads in remote or offline locations.

Snowball Edge connects to your existing applications and infrastructure using standard storage interfaces, streamlining the data transfer process and minimizing setup and integration.

Snowball Edge can cluster together to form a local storage tier and process your data on-premises, helping ensure your applications continue to run even when they are not able to access the cloud.

Snowmobile

Snowmobile is an Exabyte-scale data transfer service used to move EXTREMELY large amounts of data to AWS.

You can transfer up to 100PB per Snowmobile, a 45ft long ruggedized shipping container, pulled by a semi-truck.

Snowmobile makes it easy to move massive volumes of data to the cloud, including video libraries, image repositories, or even a complete data center migration. Transferring data with Snowmobile is secure, fast and cost effective.

Links

- https://AWS.amazon.com/snowball/

Exam Tips

S3, Glacier

General

- S3 is object based, allows you to upload files
- Files can be 0B up to 5TB
- Unlimited storage
  - Files are stored in Buckets (folder)
  - S3 uses universal namespace. bucket names must unique
- Control access to buckets using either a bucket ACL routing Bucket Policies
- By default, BUCKETS ARE PRIVATE AND ALL OBJECTS STORED INSIDE THEM ARE PRIVATE

Reads and Writes

- Read after Write consistency for PUTS of new objects
- Eventual consistency of overwrite PUTS and DELETES (can take time to propagate)

Storage Class Tiers

- S3 (normal) - durable, immediately available, frequently used
- S3 IA (infrequent access) - like normal S3 tier but infrequently accessed
- S3 Reduced Redundancy Storage (RRS) - data storage that is easily reproducible, such as thumb nails etc.
- Glacier (separate product from S3) - Used to archive data. Low and slow retrieval

Core fundamentals of S3 Object

- key (name)
- value (data)

- version id
- metadata
- sub resources
    - ACL
    - Torrent
- Object based storage only
- Not installable on apps, DB or OS
- Success uploads will generate HTTP 200 status code
- Read S3 FAQ before taking the exam. it comes up a lot

Encryption

- Client-side encryption
- Server-side encryption
    - encryption with amazon s3 managed keys (SSE-S3)
    - encryption with KMS (SSE-KMS)
    - encryption with Customer Provided Keys (SSE-C)

Versioning

- Stores all version of an object (all writes/updates and even if you delete the object). Must manually delete object if you wish to delete a version
- Great back up tool
- Once enabled, cannot be disabled, only suspended
- Integrates with Lifecycle rules
- Versioning MFA Delete capability, uses multi-factor authentication, can be used to provide an additional layer of security

Cross Region Replication

- Versioning must be enabled on source and destination buckets
- Regions must be unique, Cannot cross region to same region
- Files are not replicated automatically. All subsequent updated files will be replicated automatically.
- You cannot replicate to multiple buckets - daisy chaining (currently).
- Delete markers are replicated
- Deleting individual versions or delete markers will not be replicated

- Understand what CRR at high level

Lifecycle management

- Can be used with or without versioning
- Can be applied to current version as well as previous versions
- Acceptable actions
    - Transition to Standard - IA Storage Class (128kb and 30 days after creation date)
    - Archive to Glacier - 30 days after IA Storage if relevant
    - Permanently delete
- Understand at high level

CDN Cloudfront

- Edge Location - Location where content will be cached - separate from AWS Region
- Origin - Origin of all files the CDN will distribute. Can be S3, EC2, Elastic Load Balancer, Route 53 or your own custom server.
- Distribution - Name given to the CDN which consists of a collection of Edge Locations
    - Web Distribution - Typically used for websites
    - RTMP - Used for media streaming
- Edge Locations are not just for READ only, you can write (PUT) too!
- Object are cached for life of TTL (Time To Live)
- Can clear cached objects, but you will be charged

Storage Gateway

- File Gateway - For flat files, stored directly on S3.
- Volume Gateway:
  - Stored Volumes - Entire dataset is stored on site and is asynchronously backed up to S3
  - Cached Volumes - Entire dataset is stored on S3 and the most frequent accessed data is cached on site.
- Gateway Virtual Tape Library

- Used for backup and uses popular backup applications like NetBackup, Backup Exec, Veeam etc.

Snowball

- Understand what a Snowball is
- Understand what Import Export is
- Snowball can
    - Import to S3
    - Export from S3

EC2 – The Backbone of AWS

AWS EC2 is a web service that provides resizable compute capacity in the cloud. EC2 reduces 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.

EC2 has changed the economics of cloud computing by allowing you to pay only for capacity that your actually use. EC2 provides developers the tools to build failure resistant applications and isolate themselves from common failure scenarios.

Pricing Options

On Demand

Allows you to pay a fixed rate by the hour (or by the second) with no commitment.

Use Cases

- Perfect for users that want the low cost and flexibility of EC2 without any of the up-front payment or long-term commitment
- Applications with short term, spikey or unpredictable workloads that cannot be interrupted
- Applications being developed or tested on EC2 for the first time

Reserved

Provides you with a capacity reservation, and offer a significant discount on the hourly charge for an instance. 1 year or 3-year terms.

Use Cases

- Applications with steady state or predictable usage
- Applications that require reserved capacity

- Users can make up front payments to reduce their total computing costs even further
  - Standard RIs (Up to 75% off on-demand)
  - Convertible RIs (Up to 54% off on-demand) feature the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Ability to go from CPU intensive instance to Memory intensive.
  - Scheduled RIs are available to launch within the time window you reserve. This option allows you to match your capacity reservation to predictable recurring schedule that only requires a fraction of a day, a week, or a month.

Spot

Enables you to bid whatever price you want for an instance capacity, providing for even greater savings if your applications have flexible start and end times.

Use Cases

- Applications that have flexible start and end times
- Applications that are only feasible at very low compute prices
- Used for single compute instances to save on costs compared to 9-5 during the week.
- Users with an urgent need for a large amount of additional computing capacity.

Dedicated Hosts

Physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses.

Use Cases

- Useful for regulatory requirements that may not support multi-tenant visualization.
- Great for licensing which does not support multi-tenancy or cloud deployments
- Can be purchased On-Demand (hourly).
- Can be purchased as a Reservation for up to 70% off the On-Demand price.

EC2 Instance Types

No need to memorize for associate exams

Family	Specialty	Use Cases
F1	Field Programmable Gate Array	Genomics research, financial analytics, real-time video processing, big data etc.
I3	High Speed Storage	NoSQL DBs, Data warehousing
G3	Graphics Intensive	Video Encoding / 3D Application Streaming
H1	High Disk Throughput	MapReduce-based workloads, distributed file systems such as HDFS and MapR-FS
T2	Lowest Cost General Purpose	Web Servers / Small DBs
D2	Dense Storage	Fileservers / Data Warehousing / Hadoop
R4	Memory Optimization	Memory Intensive Apps/DBs
M5	General Purpose	Application Servers
C5	Compute Optimized	CPU Intensive Apps / DBs
P3	Graphics / General Purpose GPU	Machine Learning, Bit Coin Mining etc.
X1	Memory Optimized	SAP HANA / Apache Spark

How to remember EC2 instance types F.I.G.H.T.D.R.M.C.P.X (after 2017 reinvent):

- _F_ - FGPA
- _I_ - IOPS
- _G_ - Graphics
- _H_ - High Disk Throughput
- _T_ - Cheap General Purpose (think T2 Micro)
- _D_ - Density
- _R_ - Ram
- _M_ - Main choice for general purpose applications
- _C_ - Compute
- _P_ - Graphics(Pics)
- _X_ - Extreme Memory

EBS - Elastic Block Storage

Amazon EBS allows you to create storage volumes and attach them Amazon EC2 instances. Once attached, you can create a file system on top of theses volumes, run a database, or use them in any other way you would use a block device. EBS volumes are placed in a specific Availability Zone, where they are automatically replicated to protect you from the failure of a single component.

TLDR; A disk in the cloud that you attach to your EC2 instances

EBS Volume Types

- General Purpose SSD (GP2)
  - General purpose, balances both price and performance.
  - Ratio of 3 IOPS per GB with up to 10,000 IOPS and the ability to burst up to 3000 IOPS for extended periods of time for volumes at 3334 GB and above
- Provisioned IOPS SSD (IO1)
  - Designed for I/O intensive applications such as large relational or NoSQL databases.
  - Use if you need more than 10,000 IOPS
  - Provision up to 20,000 IOPS per volume
  - Super high performance
- Throughput Optimized HDD (ST1)
  - Big Data
  - Data warehouses
  - Log processing
  - Cannot be a boot volume
- Cold HDD (SC1)
  - Lowest cost storage for infrequently accessed workloads
  - File server
  - Cannot be a boot volume
- Magnetic (Standard)
  - Lowest cost per GB of all EBS volume types that is bootable. Magnetic volumes are ideal for workloads where data is accessed infrequently, and applications where the lowest storage cost is important

Let's get our hands dirty! Launch an EC2 instance lab!

Summary

- Termination protection is turned off by default, you MUST turn it on.
- On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated
- EBS Root Volume of you DEFAULT AMI's cannot be encrypted. You can also use a third-party tool (such as bit locker) to encrypt the root volume, or this can be done when creating AMI's (future lab) in the AWS console or using the API.
- Additional volumes can be encrypted.

Security Groups

What is a Security Group?

A security group is a virtual firewall that's controlling traffic to your EC2 instance. When you first launch as EC2 instance you associate it to 1 or more instances. You have the ability to add rules to these security groups that allows traffic to or from these instances.

Security Groups - General

1. Any security group rules apply immediately
2. Security groups are _STATEFUL_. Inbound rules automatically add outbound rules
3. All traffic is blocked by default and included through the rules. Whitelist
4. All outbound traffic is allowed
5. You can have multiple EC2 instances within a security group.
6. You can have multiple security groups attached to EC2 instances.
7. You cannot block specific IP addresses using Security Groups, use Network Access Control Lists.
8. You can specify allow rules, but not deny rules.

RAID, Volumes & Snapshots

RAID - Redundant Array of Independent Disks

- RAID 0 - Striped, no redundancy, good performance. If one fails, you lose all
- RAID 1 - Mirrored, redundant. If one fails, others available
- RAID 5 - Good for reads, bad for writes, AWS does not recommend ever putting RAID 5's on EBS. Strongly discouraged.
- RAID 10 - Striping & Mirrored, good redundancy, good performance.

How can I take a Snapshot of a RAID Array?

- Problem - Taking a snapshot excludes the data held in cache by applications and the OS. This doesn't really matter on single volume, however when using multiple volumes in a RAID Array, this can be a problem due to interdependencies of the array.

- Solution - Take an application specific snapshot.
  - Stop application from writing to disk.
  - Flush all caches to the disk.
  - How can we do this?
    - Freeze the file system
    - Unmount the RAID Array
    - Shutting down the associated EC2 instance.

Create an AMI lab - Volumes vs. Snapshots

Snapshots of Root Device Volumes

- To create a snapshot for Amazon EBS volumes that server as root devices, you should stop the instance before taking the snapshot.

Security

- Snapshots of encrypted volumes are encrypted automatically
- Volumes restored from encrypted snapshots are encrypted automatically.
- You can share snapshots, but only if they are unencrypted.
  - Said snapshots can be shared with other AWS accounts of made public

AMI Types

What should you select your AMI based on?

- Region
- OS
- Architecture
- Launch Permissions
- Storage for the Root Device (Root Device Volume)
  - Instance Store (Ephemeral Store)
  - EBS Backed Volumes

EBS vs. Instance Store

All AMIs are categorized as either backed by Amazon EBS or backed by instance store.

_For EBS Volumes:

The root device for an instance launched from the AMI is an Amazon EBS volume created 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.

Elastic Load Balancers

What is a load balancer?

A virtual appliance that balances the load of HTTP traffic etc. of your web application/web servers.

Types of Load Balancers

- Application Load Balancers
- Network Load Balancers
- Classic Load Balancers

Application Load Balancer _(Intelligent)_

Best suited for load balancing of HTTP(S) traffic. They operate at Layer 7 (OSI) and are application aware. They are intelligent, and you can create advanced request routing, sending specified requests to specific web servers.

Network Load Balancer _(Performance)_

Best suited for load balancing of TCP traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancers are capable of handling millions of requests per second, while maintaining ultra-low latencies.

Classic Load Balancer (OG, Legacy Load Balancer)

Used to load balance HTTP(S) applications and use Layer 7-specific features, such as X-Forwarded and stick-sessions. You can use strict Layer 4 load balancing for applications that rely purely on the TCP protocol.

504 Error

- If no response or timeout, the ELB (Elastic Load Balancer) responds with status code 504.
- Internal Server Error type - DB Layer or Web Server Layer.
- Solution: Identify issue where failing and scale up or out where possible.

Placement Groups (Exam MUST KNOW!!)

Two Types of Placement Groups

Clustered Placement Group

A cluster placement group is a grouping of instances within a single Availability Zone. Placement groups are recommended for applications that need low network latency, high network throughput, or both.

_NOTE: Only a certain number of instances can be launched in to a Clustered Placement Group.

Spread Placement Group

Opposite of a Clustered Placement Group. A Spread Placement Group is a group of instances that are each placed on distinct underlying hardware. Spread Placement Groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.

EFS (Elastic File System)

AWS EFS is file storage service for AWS EC2 instances. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With AWS EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

EFS Features

- Supports the Network File System version 4 (NFSv4) protocol
- You only pay for the storage you use (no pre-provisioning required)
- Can scale up to the petabytes
- Can support thousands of concurrent NFS connections
- Data is stored across multiple AZ's within a region
- Read After Write Consistency

Lambda

What is Lambda?

AWS Lambda is a compute service where you can upload your code and create Lambda function. AWS Lambda takes care of provisioning and managing the servers that you use to run the code. Worry free from OS, patching, scaling, etc.

Use Cases

- As an event-driven compute service where AWS Lambda runs your code in response to events. These events could be changes to data in an Amazon S3 bucket or an Amazon DynamoDB table.

- As a compute service to run your code in response to HTTP requests using Amazon API Gateway or API calls made using AWS SDKs.

Encapsulation of the following:

- Data Centers
- Hardware
- Assembly Code/Protocols
- High Level languages
- Operation Systems
- Application Layer/AWS API's
- AWS Lambda

Compatible Languages:

- C
- Java
- Node.js
- Python

How is Lambda priced?

- Number of requests

- First 1m request are free. $0.20 per 1m requests thereafter.

- Duration
  - Duration is calculated from the time your code begins execution 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 $0.00001667 for every GB-second used.

Why is Lambda cool?
- No SERVERS!!
- Continuous Scaling
- Super cheap

Exam Tips

EC2 Instance Run Down

- On Demand - allows you to pay a fixed rate by the hour (or second) with not commitment

- Reserved - provides you with the capacity reservation, and offer a significant discount on the hourly charge for an instance. 1 year or 3-year terms

- Spot - Enables you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times

- Dedicated Hosts - Physical EC2 server dedicated for your use. Dedicated Hosts can help reduce costs by allowing you to use your existing server-bound software license

Important Note!!

If a Spot instance is terminated by Amazon EC2, you will not be charged for a partial hour of usage. However, if you terminate the instance yourself, you will be charged for the complete hour in which the instance ran.

Instance Types

- F. - FGPA
- I. - IOPS
- G. - Graphics
- H. - High Disk Throughput
- T. - Cheap General Purpose (think T2 Micro)
- D. - Density
- R. - Ram
- M. - Main choice for general purpose applications
- C. - Compute
- P. - Graphics (Pics)
- X. - Extreme Memory

Volume Types

SSD

- General Purpose (SSD) - balances price and perf. for a wide variety of workloads

- Provisioned IOPS (SSD) - Highest perf. SSD volume for mission-critical low-latency or high-throughput workloads

Magnetic

- Throughput Optimized HDD - Low cost HDD volume designed for frequently accessed, throughput-intensive workloads

- Cold HDD - Lowest cost HDD volume designed for less frequently accessed workloads

- Magnetic - Previous Generation. Can be a boot volume.

Upgrading EBS Volume Types - Lab

Volumes & Snapshots

- Volumes exist on EBS
  - Virtual Hard Disk
- Snapshots exist on S3
- Snapshots are a point in time copies of Volumes
- Snapshots are incremental - this means that only the blocks that have changed since your last snapshot are moved to S3. Only recording the changes
- If it's 1st snapshot, takes time to create

Snapshots of Root Device Volumes

- To create a snapshot of Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot, however you can take a snapshot while instance is running.
- However, you can take a snap while the instance is running.
- You can create AMI's from EBS-backed Instances and Snapshots.
- You can change EBS volume sizes on the fly, including changing the size and storage type.
- Volumes will ALWAYS be in the same availability zone as the EC2 instance.
- To move and EC2 volume from one AZ/Region to another, take a snap or an image of it, then copy it to the new AZ/Region.

Volumes vs Snapshots - Security

- Snapshots of encrypted volumes are encrypted automatically.
- Volumes restored from encrypted snapshots are encrypted automatically.
- You can share snapshots, but only if they are unencrypted.
  - These snapshots can be shared with other AWS accounts or made public.

EBS vs. Instance Store

- Instance store volumes are sometimes called _Ephemeral Storage_.
- Instance store volumes cannot be stopped. If the underlying host fails, you will lose all your data.
- EBS backed instances can be stopped. You will not los the data on this instance if it is stopped.
- You can reboot both, you will not lose your data.

- By default, both ROOT volumes will be deleted on termination, however with EBS volumes, you can tell AWS to keep the root device volume.

Load Balancers

- 3 Types of Load Balancers
  - Application Load Balancers
  - Network Load Balancers
  - Classic Load Balancers

- 504 Error means the gateway has timed out. Application is not responding within the idle timeout period
  - Trouble shoot the application. Web Server or Database Server?

- If you need IPv4 address of your end user, look from the X-Forwarded-For header.
- Instances are monitored but ELB are reported as `InService` or `OutofService`.
- Health Checks check the instance health by talking to it.
- ELB's have their own DNS name. You are never given an IP address
- Read the ELB FAQ for Classic Load Balancers

Note: ELB's do not have IP Addresses, only found by DNS namespace

CloudWatch

- Standard Monitoring - 5 minutes
- Detailed Monitoring - 1 minute

What can you do with CloudWatch? (Not to be confused with CloudTrail)

- Dashboards - Creates awesome dashboards to see/monitor what is happening with your AWS environment.
- Alarms - Allows you to set Alarms that notify you when a particular threshold are hit.
- Events - Helps you to respond to state changes in your AWS resources.

- Logs - Helps you to aggregate, monitor and store logs.

Placement Groups

- A Clustered Placement Group cannot span multiple Availability Zones.
- A Spread Placement Group can.
- The name you specify for a placement group must be unique within your AWS account.
- Only certain types of instances can be launched in a placement group (Compute Optimized, GPU, Memory Optimized, Storage Optimized)
- AWS recommend homogenous instances within placement groups.
- You can’t merge placement groups
- You can’t move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

Lambda

- Lambda scales horizontally (not vertically) automatically. Redundancy
- Lambda functions are independent, 1 event = 1 function
- Lambda is serverless
- Know what services are serverless!!
  - S3
  - API Gateway
  - DynamoDB
- Lambda functions can trigger other lambda functions, 1 event can = x functions if functions trigger other functions.
- Architectures can get extremely complicated, AWS X-ray allows you to debug what is happening
- Lambda can do things globally, you can use it to back up S3 buckets to other S3 buckets etc.
- Know your triggers - connecting AWS services

Summary (TLDR;)

- Know the differences between EC2 instances
  - On Demand

- Spot
- Reserved
- Dedicated hosts

Remember with Spot Instances

- If you terminate the instance, you pay for the hour
- If AWS terminates the instance, you get the hour it was terminated for free.

EC2 Instance Types
F.I.G.H.T.D.R.M.C.P.X (Use Reference)

EBS (Elastic Block Storage)
Consists of:
- SSD, General Purpose - GP2 - Up to 10,000 IOPS
- SSD, Provisioned IOPS - IO1 - More than 10,000 IOPS
- HDD, Throughput Optimized - ST1 - frequently accessed workloads
- HDD, Cold - SC1 - Less frequently accessed data
- HDD, Magnetic - Standard - Cheap, Infrequently accessed storage.

IMPORTANT NOTE: You cannot mount 1 EBS volume to multiple EC2 instances; Instead use EFS (Elastic File Storage)

Lab Tips!

- Termination Protection is turned off by default, you must turn this on!
- On a EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated.
- EBS backed Root volumes can now be encrypted using AWS API or console, or you can use a third-party tool (bit locker etc.) to encrypt the root volume.
- Additional volumes can be encrypted

Volumes vs. Snapshots

- Volumes exist on EBS; Virtual Hard Disks
- Snapshots exist on S3
- You can take a snapshot of a volume, this will store that volume on S3

- Snapshots are point-in-time copies of volumes
- Snapshots are incremental. This means that only the blocks that have changed since your last snapshot are moved to S3
- If taking your first snapshot, may take some time

Security

- Snapshots of encrypted volumes are encrypted automatically
- Volumes restored from encrypted snapshots are encrypted automatically
- You can share snapshots, but only if they are unencrypted
  - These snapshots can be shared with other AWS accounts or made public

Snapshots or Root Device Volumes

- To create a snapshot for EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

EBS vs Instance Store

- Instance Store Volumes are sometimes called Ephemeral Storage
- 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 the data on this instance if it is stopped.
- You can reboot both, you will not lose your data
- By default, both ROOT volumes will be deleted on termination. However, with EBS volumes, you can tell AWS to keep the root device volume.

How can you take a snapshot of a RAID Array?

Problem - Take a snapshot, the snapshots excludes data held in the cache by applications and the OS. This tends not to matter on a single volume. However, using multiple volumes in a RAID array, this can be a problem due to interdependencies of the array.

Solution - Take an application consistent snapshot.

- Stop the application from writing to disk
- Flush all caches to the disk.

How is this accomplished?

- Freeze the file system
- Unmount the RAID array
- Shutting down the associated EC2 instance.

AMI (Amazon Machine Image)

AMIs are regional. You can only launch an AMI from the region in which its stored. However, you can copy AMIs to other regions using the console, command line, or the Amazon EC2 API

- Standard monitoring - 5 min
- Detailed monitoring - 1 min

- Cloudwatch is for performance monitoring
- Cloudtrail is for auditing

Cloudtrail

- Dashboards - Cloudwatch creates awesome dashboards to see what is happening with your AWS environment
- Alarms - Allows you to set alarms when particular thresholds are hit.
- Events - Helps you to respond to state changes in your AWS resources.
- Logs - Helps you to aggregate, monitor, and store logs

Roles

- Roles are more secure than storing your access key and secret access key on individual instances.
- Roles are easier to manage
- Roles can be assigned to an EC2 instance after it has been provisioned using both the command line and the AWS console
- Roles are universal - they can be used in any region

Instance Metadata

- Used to get information about an instance (public IP, DNS etc.)
  - `curl http://169.254.169.254/latest/meta-data`
  - `curl http://169.254.169.254/latest/user-data`

EFS (Elastic File System)

- Supports the Network File System version 4 (NFSv4) protocol
- You only pay for the storage you use (no pre-provisioning required)
- Can scale up to petabytes
- Can support thousands of concurrent NFS connections
- Data is stored across multiple AZs within a region
- Read after Write consistency

Lambda

- Lambda is a compute service where you can upload you code and create a Lambda function.
- Takes care of provisioning and managing servers that you use to run your code.
- Need not worry about OS, patching, scaling etc.

_Use Lambda as:

- Event driven compute service where Lambda runs your code in response to events. These events could be changes in an S3 bucket or Dynamo DB table.
- A compute service to run your code in response to HTTP requests using API Gateway or API calls made using AWS SDKs

Placement Groups
Know the differences between and why you would use...

- Clustered Placement Groups
- Spread Placement Groups

Databases on AWS

Types of Databases

Relational Databases

Relational databases are what most of us are all used to. They have been around since the 70's and you can think about them like spreadsheets!

- Database
- Tables
- Columns
- Rows

id	name	age	location
1	nigel	30	San Diego
2	jim	28	NYC
3	betty	31	San Francisco

Relational Databases Examples

- SQL Server
- Oracle
- MySQL
- PostgreSQL
- Aurora
- MariaDB

Non-Relational (NoSQL)

- Database
  - Collection => Table
  - Document => Row
  - Key, Value Pairs => Columns

Non-Relational Databases Examples

```json
{
  "_id": "394ejojaj903091881dnna",
  "name": "nigel",
  "age": 30,
  "location": "San Diego"
}
```

Data Warehousing

Used for business intelligence. Tools like Cognos, Jaspersoft, SQL Server, Reporting Services, Oracle Hyperion, SAP NetWeaver.

Used to pull in very large and complex data sets. Usually used by management to do queries on data (such as current performance vs targets etc.).

OLTP (Online Transaction Processing) vs. OLAP (Online Analytics Processing)

OTLP differs from OLAP in terms of the types of queries you will run.

OLTP Example

Used for transactional type queries.

```
Order number: 2120121

Pulls up a row of data such as Name, Date, Address to Deliver to, Delivery Status etc.
```

OLAP Example

Used for business logic type queries.

```
Net Profit of given product or device
Pulls in large number of records

Sum of products sold in region
Sum of products sold in continent
Unit cost of product in each region
Sales price of each product
Sales price - unit cost
```

Data Warehousing databases use different type of architecture both from a database perspective and infrastructure layer.

ElasticCache

ElasticCache is a web service that makes it easy to deploy, operate and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

ElasticCache supports two open-source in-memory caching engines...

1. Memcached
2. Redis

Backups, Multi-AZ & Read Replicas

Automated Backups

Automated Backups allow you to recover your database to any point in time within a 'retention period'. The retention period can be between one and 35 days.

Automated Backups will take a full daily snapshot and will also store transaction logs throughout the day.

When you do a recovery, AWS will first choose the most recent daily backup, and then apply transaction logs relevant to that day. This allows you to do a point in time recovery down to a second, within a retention period.

Database Snapshots

DB Snapshots are done manually (i.e. they are user initiated) They are stored even after you delete the original RDS instance, unlike automated backups.

Restoring Backups

Whenever you restore either an Automatic Backup or a manual Snapshot, the restored version of the database will be a new RDS instance with a new DNS endpoint

\`original.us-west-1.rds.amazonAWS.com\` -> \`restored.eu-west-1.rds.amazonAWS.com\`

Encryption

Encryption at rest is supported for MySQL, Oracle, SQL Server, PostgreSQL, MariaDB & Aurora.

Encryption is done using the AWS Key Management System (KMS) service. Once your RDS instance is encrypted, the data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas and snapshots.

At the present time, encrypting an existing DB Instance is not supported. To use RDS encryption for an existing database, you must first create a snapshot, make a copy of that snapshot and encrypt the copy.

Multi-AZ

Multi-AZ allows you to have an exact copy of your production database in another Availability Zone. AWS handles the replication for you, so when your production database is written to, this write will automatically be synchronized to the stand by database.

In the event of planned database maintenance, DB instance failure, or AZ failure, RDS will automatically failover to the standby so that database operations can resume quickly without admin intervention.

NOTE: It is not primarily used for improving performance, really only disaster recovery. For performance improvement, you need Read Replicas

Multi-AZ Available DBs

- SQL Server
- Oracle
- MySQL Server
- PostgreSQL
- MariaDB

Read Replicas

Read replicas allow you to have a read-only copy of your production database. This is achieved by using async replication from the primary RDS instance to the Read Replica. You use Read Replicas primarily for very read-heavy database workloads.

- Used for scaling, not disaster control!
- Must have auto backups turned on in order to deploy a Read Replica
- You can have up to 5 Read Replica copies of any database.
- You can have Read Replicas of Read Replicas _(inception)_ - mindful of latency
- Each Read Replica will have its own DNS end point.
- You can have Read Replicas that have Multi-AZ
- You can create Read Replicas of Multi-AZ source databases
- Read Replicas can be promoted to be their own databases. This breaks the replication.
- You can have a Read Replica in a second region.

Read Replica Available DBs

- MySQL Server

- PostgreSQL
- MariaDB
- Aurora

DynamoDB

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

- Stored on SSD Storage
- Spread Across 3 geographically distinct data centers

- Eventual Consistent Read (Default)
  - Consistency across all copies of data is usually reached within a second. Repeating a read after a short amount of time should return the updated data. (Best Read Perf.)

- Strongly Consistent Reads
  - A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.

NOTE: Super easy to scale! Push button scaling

Pricing

Pricing is based on provision throughput capacity

- Write Throughput $0.0065 per hour for every 10 units
- Read Throughput $0.0065 per hour for every 50 units
- Storage costs of $0.25G per month

_Pricing Example:

```
Constraint: 1 million WRITEs and 1 million READs per day, while storing 3G of data.

First, calculate how many writes and reads per second you need.

1 million evenly spread writes per day is equivalent to 1,000,000 (writes) /24 (hours) / 60 (minutes) / 60 (seconds) = 11.6 writes per second.

-- BREAKDOWN --

DynamoDB WRITE Capacity Unit - 1 per second = 12
DynamoDB READ Capacity Unit - 1 per second = 12

READ Capacity Units - billed in blocks of 50
WRITE Capacity Units - billed in blocks of 10

Calculation:  WRITE Capacity Units = (0.0065 / 10) x 12 x 24 = $0.1872
Calculation: READ Capacity Units = (0.0065 / 10) x 12 x 24 = $0.0374
```

Redshift

Amazon Redshift is a fast and powerful, fully managed petabyte-scale data warehouse service in the cloud.

Customers can start small for just $0.25 per hour with no commitments or upfront costs and scale to a petabyte or more for $1,000 per terabyte per year, less than 1/10 of most data warehousing solutions.

Configuration

- Single Node (160Gb)
- Multi-Node
  - Leader Node _(manages client connections and receives queries)_
  - Compute Node _(store data and perform queries and computations)_ - Up to 128 Compute Nodes

Columns

Columnar Data Storage - Instead of storing data as rows, Redshift organizes the data by column.

Unlike row-based systems, which are ideal for transaction processing, column-based systems are ideal for data warehousing and analytics, where queries often involve aggregates performed over large data sets.

Since only the columns involved in the queries are processing and columnar data is stored sequentially on the storage media, column-based systems require far fewer I/Os, greatly improving query performance.

Compression

Advanced Compression - Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk.

Redshift employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores. In addition, Redshift doesn't require indexes or materialized views and so 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.

MPP

Massive Parallel Processing (MPP) - Redshift automatically distributes data and query load across all nodes. Redshift makes it easy to add nodes to your data warehouse and enables you to maintain fast query performance as your data warehouse grows.

Pricing

How is Redshift priced?

- Compute Node Hours
  - Total number of hours you run across all your compute nodes for the billing period
  - Billed for 1 unit per node per 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 leader node hours; only compute nodes will incur charges

- Backups
- Data transfers (Only within a VPC, not outside of it)

Security

- Encrypted in transit using SSL
- Encrypted at rest using AES-256 encryption
- By default, Redshift takes care of key management
  - Manages your keys through HSM (Hardware Security Module)
  - AWS Key Management Service (KMS)

Availability

- Currently only available in 1 AZ - Realistically only for business logic
- Can restore snapshots to new AZ's in the event of outage.

ElasticCache

ElasticCache is a web service that makes it easy to deploy, operate and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

Why ElasticCache?

ElasticCache can be used to significantly improve latency and throughput for many read-heavy applications workloads - i.e. networking, gaming, media sharing and Q&A portals 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 computationally intensive calculations

Types of ElasticCache

- Memcached
  - A widely adopted memory object caching system. ElasticCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.

- Redis
  - A popular open-source in-memory key-value store that supports data structures such as sorted sets and lists. ElasticCache supports Master/Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy.

Aurora

What is Aurora?

Aurora is a MySQL-compatible, relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost effectiveness of open source databases. Aurora provides up to 5x better performance than MySQL at a price point of 1/10 that of a commercial database while delivering similar performance and availability

Scaling

- Start with 10G, Scales in 10G increments to 64 TB (Storage Autoscaling)
- Compute resource can scale up to 32vCPUs and 244G of Memory.
- 2 copies of your data is contained in each availability zone, with minimum of 3 AZ -> 6 copies of your data! Highly redundant
- Designed to transparently handle the loss of up to 2 copies of data without affecting database write availability and up to 3 copies without affecting read availability.

- Aurora storage is also self-healing. Data blocks and disks are continuously scanned for errors and repaired automatically.

Aurora Replicas

- 2 Types of Replicas are available
- Aurora Replicas - Up to 15 replicas currently
- MySQL Replicas - Up to 5 replicas currently

Exam Tips

ElasticCache

You will be given a scenario where a particular database is under a lot of stress/load. You may be asked which service you should use to alleviate this.

ElasticCache is a good choice if your database is particularly read heavy and not prone to frequent changing.

Redshift is a good answer if the reason your database is feeling stress is because management keep running OLAP transactions on it etc.

Summary

Types

- RDS - OLTP
  - SQL
  - MySQL
  - PostgreSQL
  - Oracle
  - Aurora
  - MariaDB
- DynamoDB - NoSQL
- Redshift - OLAP
- ElasticCache - In Memory Caching
  - Memacached
  - Redis

READ FAQ RDS SECTION IN DOCUMENTATION!!

https://AWS.amazon.com/rds/faqs/

Application Services

SQS - Simple Queue Service

First EVER AWS Service!

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

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

SQS Breakdown

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

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

What do you mean by "Queue"?

The 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 issues that arise if the producer is producing faster than the consumer can process it, of if the producer or consumer are only intermittently connected to the network.

Queue Types

Standard Queue (default)

Amazon SQS offers standard as the default queue type. A standard queue lets you have a nearly-unlimited number of transactions per second. Standard queues guarantee that a message is delivered at least once. However, because of the highly 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.

FIFO Queues (First In, First Out)

The FIFO queue complements the standard queue. The most important features of this queue type are FIFO delivery and exactly one processing: The order in which messages are sent and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it; duplicates are not introduced into 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

```|_5_| ---> |_4_| ---> |_3_| ---> |_2_| ---> |_1_|```

Key Facts

- SQS is pull-based, not pushed based
- Messages are 256Kb in size
- Messages can be kept in the queue from 1 minute to 14 days
- Default retention period is 4 days
- SQS guarantees that your messages will be processed at least once.

Visibility Timeout

- The Visibility Timeout is the amount of time that the message is invisible in the SQS queue after the reader picks up that message. Provided the job is processed before the visibility timeout expires, the message will then be deleted from the queue. If the job is not processed within that time, the message will become visible again and another reader/worker will process it. This could result in the same message delivered twice
- Default visibility timeout is 30 seconds
- Increase it if your task takes >30 seconds

- Maximum is 12 hours

Long Polling

- Amazon SQS long polling is a way to retrieve messages from your Amazon SQS queues
- While the regular short polling returns 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.
- Waits till message is in the queue.
- As such, long polling saves you money.

SWF - Simple Workflow Service

Amazon Simple Workflow Service is a web service that makes it easy to coordinate work across distributed application components. Amazon SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Tasks represent invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, scripts.

Starters

An application that can initiate a workflow. Could be your e-commerce website when placing an order or a mobile app searching for bus times

Workers

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

Deciders

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

Workers and Deciders Interaction

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

At the same time, Amazon SWF stores tasks, assigns them to workers when they are ready and monitors their progress. It ensures that a task is assigned ONLY ONCE and is NEVER DUPLICATED (key difference from SQS).

Since Amazon SWF maintains the applications state durably, workers and deciders don't have to keep track of execution state. They can run independently, and scale quickly.

SWF Domains

Your workflow and activity types and the workflow execution itself are all scoped to a domain. Domains isolate a set of types, executions, and task lists from others within the same account.

You can register a domain by using the AWS Management Console or by using the Register Domain action in the Amazon SWF API.

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

JSON Domain Registration Example

```JSON
{
  "name": "92034",
  "description": "music",
  "workflowExecutionRetentionPeriodInDays": "60"
}
```

SWF vs. SQF

- Amazon SWF has a retention period of 1 year vs SQS's 14 days retention
- Amazon SWF presents a task-oriented API, whereas Amazon SQS offers a message-oriented API
- Amazon SWF ensures that a task is assigned ONLY ONCE and is NEVER DUPLICATED. With SQS, you need to handle duplicated messages and may also need to ensure that a message is processed only once.
- Amazon 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.

SNS - Simple Notification Service

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

May push notifications to Apple, Google, Fire OS and Windows devices as well as Android devices in China with Baidu Cloud Push.

Besides pushing cloud notifications directly to mobile devices, SNS can also deliver notifications by SMS text message or email, to SQS queues, or to any HTTP endpoint.

SNS notifications can also trigger Lambda functions. When a message is published to and SNS topic that has a Lambda function subscribed to it, the Lambda function is invoked with the payload of the published message. 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.

SNS Structure

SNS allows you to group multiple recipients using topics. A topic is an "access point" for allowing recipients to dynamically subscribe for identical copies of the same notification.

One topic can support deliveries to multiple endpoint types - for example, you can group together iOS, Android and SMS recipients. When you publish once to a topic, SNS delivers appropriately formatted copies of your message to each subscriber.

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

Subscribers - Who may subscribe to notifications?

- HTTP
- HTTPS
- Email
- Email-JSON
- SQS
- Application
- Lambda

SNS Benefits

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

SNS vs SQS

- Both messaging services in AWS
- SNS = push; SQS = polls (pulls)

Pricing

- User pays $0.50 per 1 million SNS Requests
- $0.06 per 100,000 notification deliveries over HTTP
- $0.75 per 100 notifications deliveries over SMS
- $2.00 per 100,000 notification deliveries over email

Elastic Transcoder

- Media Transcoder in the cloud.
- Convert media files from their original source format in to 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 bets on particular devices.
- Pay based on the minutes that you transcode and the resolution at which you transcode.

API Gateway

API Gateway’s a fully managed service that makes it easy for developers to publish, maintain, monitor and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create and API that acts as a "front door" 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.

Caching

You can enable API caching in API Gateway to cache your endpoints response. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of the requests to your API.

When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified TTL period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.

- Low cost & efficient
- Scales effortlessly
- You can throttle requests to prevent attacks
- Connect to Cloudwatch to log all requests

Kinesis

What is streaming data?

Streaming data is data that is generated continuously by thousands of data sources, which typically send in the data records simultaneously, and in small sizes (order of KB)

Examples of usage:

- Purchases from online stores
- Stock prices
- Game data
- Social network data
- Geospatial data - uber, google maps
- iOT data

What is Kinesis?

AWS Kinesis is a platform on AWS to send your streaming data to. Kinesis makes it easy to load and analyze streaming data, and also providing the ability for you to build your own custom applications for your business needs.

Core Kinesis Services?

Kinesis Streams

- Streams consist of shards

- 5 transactions 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 1 Mb per second (including partition keys).
- 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.

Kinesis Firehose

- Handles stream data automatically, no need to specify shards.

Kinesis Analytics

- Allows you to run SQL queries, analyzing the data and store said data in to another storage service like S3

Exam Tips

SQS

- SQS is a distributed message queueing system
- Allows you to decouple the components of an application so that they are independent
- Pull-based, not push- based
- Standard queues (default) - best effort ordering; message delivered at least once
- FIFO Queues (First In First Out) - ordering strictly preserved, message delivered once, no duplicates. e.g. good for banking transactions which need to happen in strict order.

NOTE: Read FAQ section of SQS to help with exam

SNS

Subscribers:

- HTTP
- HTTPS
- Email
- Email-JSON
- SQS
- Application
- Lambda

API Gateway

- Remember what API Gateway is at a high level
- API Gateway has caching capabilities to increase performance
- API Gateway is low cost and scales automatically
- You can throttle API Gateway to prevent attacks
- You can log results to CloudWatch
- If you are using JS/AJAX that uses multiple domains with API Gateway, ensure that you have CORS enabled on API Gateway

Kinesis

- Know the difference between Kinesis Streams and Kinesis Firehose. You will be given scenario questions and you must choose the most relevant service

- High level understanding of Kinesis Analytics

Route 53

DNS

What is DNS? (Domain Name Service)

If you've used the internet, you've used DNS. DNS is used to convert human friendly domain names `(http://acloud.guru)` into an Internet Protocol (IP) address `(http://92.123.92.1)`

IP addresses are used by computers to identify each other on the network. IP addresses commonly come in 2 different forms, IPv4 and IPv6

IPv4 vs IPv6

The IPv4 space is 32-bit field and has over 4 billion different addresses (4,294,967,296)

IPv6 was created to solve this the depletion issue and has an address space of 128 bits - which is in theory 340,282,366,920,938,463,463,374,607,431,768,211,456 different addresses! _340 undecillion addresses_

Top Level Domains

If common domain names such as google.com, bbc.co.uk etc. you'll notice a string of characters separated but a `.`. The last work in the domain name represents the 'Top Level Domain'. The second word in the domain, known as the 'Second Level Domain' is optional

_Example Top Level and Second Level Domains:

```
.com
.edu
.gov
.org
.co
```

```
.co.uk
.gov.au
```

These top level domains are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database _(database of all available top level domains)_. You can view this database by going to https://www.iana.org/domains/root/db

Domain Registrars

Because all the names in a given domain have to be unique there needs to be a way to organize all of this so that domains are duplicated - hence Domain Registrars.

A registrar is an authority that can assign domain names directly under one or more top level domains. 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 Records

SOA Records store information related to a domain about:

- The name of the server that supplied data for that zone.
- The admin of that zone.
- The current version of the datafile.
- The number of seconds a secondary name server should wait before checking for updates.
- The number of seconds a secondary name server should wait before retrying a failed zone transfer.
- The maximum number of seconds that secondary name server can use data before it must either be refreshed or expire.
- The default number of seconds for the TTL file on resource records.

NS Records

NS stands for Name Server records and are used by top level domain servers to direct traffic to the Content DNS server which contains the authoritative DNS records.

A Records

An A Record is the fundamental type of DNS record and the 'A' in A record stands for 'Address'. The A Record is used by the computer to translate the name of the domain to the IP address. For example, `https://google.com` -> `https://92.123.12.1`

TTL

The length that a DNS record is cached on either the Resolving Server o the users own local PC is equal to the value of the 'Time To Live' _(TTL)_ in seconds. The lower the time to live, the faster changes to DNS records take to propagate throughout the internet.

CNAMES

A Canonical Name (CName) can be used to resolve one domain name to another. For example, you may have a mobile website with a domain name `http://m.acloud.guru` that is used for when users browse to your domain name on their mobile devices. You may also want the name `http://mobile.acloud.guru` to resolve to this same address.

Alias Records

Alias resource record sets can save you time because AWS Route 53 automatically recognizes changes in the record sets that the alias resource record set refers to.

For example, suppose an alias resource record set for example.com points to an ELB load balancer at lb1-1234.us-west-1.elb.amazonAWS.com. If the IP address of the load balancer change, AWS Route 53 will automatically reflect those changes in DNS answers for example.com without any changes to the hosted zone that contains resource record sets for example.com

Routing Policies

Simple

This is the default routing policy when you create a new record set. This is the most commonly used when you have a single resource that performs a given function for your domain, for example, one web server that serves content for the `http://acloud.guru` website.

Weighted

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 EU-WEST-1

Latency

Latency based routing allows you to route your traffic based on the lowest network latency for your end user (i.e. which region will give them the fastest response time)

To use latency-based routing you create a latency resource record set for the EC2 (or ELB) resource in each region that hosts your website. When Route 53 receives a query for your site, it selects the latency resource record set for the region that gives the user the lowest latency. Route 53 then responds with the value associated with that resource record set

Failover

Failover routing policies are used when you want to create an active/passive set up. For example, you may want your primary site to be in EU-WEST-2 and your secondary DR site in AP-SOUTHEAST-2

Route 53 will monitor the health of your primary site using a health check.

A health check monitors the health of your endpoints.

Geolocation

Geolocation routing lets you choose where your traffic will be sent based on the geographic location of your users (i.e. the location from which DNS queries originate).

For example, you might want all queries from Europe to be routed to a fleet of EC2 instances that are specifically configured for your European customers. These servers may have the local language of your European customers and all prices are displayed in Euros.

Exam Tips

DNS

- ELB's do not have pre-defined IPv4 addresses, you resolve to them using a DNS name
- Understand the difference between an Alias Record and a CNAME.
- Given the choice, always choose and Alias Record over a CNAME.

Remember the different routing policies and their use cases.

- Simple
- Weighted
- Latency
- Failover
- Geolocation

VPC - Virtual Private Cloud

Think of VPC as virtual data center in the cloud!

VPC Definition

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

You have complete control over your virtual network environment, including selection of your own IP address range, creation of subnets and config of route tables and network gateways.

You can easily customize the network config for your VPC. For example, you can create a public facing subnet of 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.

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

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

NOTE: Private and public subnets within a VPC can only have one subnet per AZ

Use (cidr.xyz)[https://cidr.xyz/] to figure out subnet ranges within a VPC

What can you do with a VPC?

- Launch instances into a subnet of your choosing
- Assign custom IP address ranges in each subnet
- Configure route tables between subnets
- Create single internet gateway and attach it to our VPC

- Much better security control over your AWS resources
- Instance security groups
- Subnet network access control lists (ACLS)

Default VPC vs Private VPC

- Default VPC is user friendly, allowing you to immediately deploy instances
- All subnets have a route to internet
- No private subnets in default VPC
- EC2 instance has both a public and private IP address.

VPC Peering

- Allows you to connect one VPC with another via a direct network route using private IP addresses
- Instances behave as if they are on the same private network.
- You can peer VPCs with other AWS accounts as well as with other VPCs in the same account
- Peering is in a star config: i.e. 1 central VPC peers with 4 others. NO TRANSITIVE PEERING!!

NAT - Network Address Translation

NAT Instances

- When creating a NAT instance, Disable Source/Destination Check on the instance.
- NAT instances must be in a public subnet
- There must be a route out of the private subnet to the NAT, in order for this to work.
- The amount of traffic that NAT instances can support depends on the instance size. If you are bottlenecking, increase the instance size.
- You can create high availability using Autoscaling Groups, multiple subnets in different AZs, and a script to automate failover.
- Behind a Security Group.

NAT Gateways

- Preferred by the enterprise
- Scale automatically up to 10G
- No need to patch OS
- Not associated with security groups
- Automatically assigned public IP
- Must update root tables and point them to NAT Gateway
- Having one NAT Gateway in one AZ is not good enough, must me redundant in multiple AZs
- No need to disable Source/Destination Checks
- More Secure than NAT Instance

NACL - Network Access Control Lists

- Can only associate 1 subnet to a Network ACLs
- Your VPC automatically comes with a default NACL, and by default it allows all inbound and outbound and traffic.
- You can create custom NACLs. By default, each custom NACL denies all inbound and outbound traffic until you add rules
- Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default NACL.
- You can associate a NACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a NACL with a subnet, the previous association is removed.
- NACLs contain a numbered list of rules that is evaluated in order, starting with the lowest numbered rule
- NACL have separate inbound and outbound rules, and each rule can either allow or deny traffic.
- NACL are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic

ALB

- You need at least 2 public subnets in order to deploy and application load balancer

VPC Flow Logs

VPC Flow Logs is a feature that enables you to capture info about the IP traffic going to and from network interfaces in your VPC. Flow log data is stored using Amazon Cloudwatch Logs.

After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.

Flow logs can be created at 3 levels:

- VPC
- Subnet
- Network Interface Level

Exam Tips

VPC Intro

- Think of a VPC as a logical data center in AWS
- Consists of IGW(or Virtual Private Gateways), route tables, network access control lists (NACL), subnets, security groups
- 1 subnet = 1 AZ
- Security groups are Stateful; NACLs are Stateless
  - Must open both inbound and outbound ports for NACLs
- NO TRANSITIVE PEERING!!
- Allowed 5 VPC's in each AWS Region by default

Flow Logs

- You cannot enable flow logs for VPCs that are peered with your VPC unless the peer VPC is in your account.
- Cannot tag a flow log
- After you've created a flow log, you cannot change its configuration; for example, you can’t associate a different IAM role with the flow log

Not all IP traffic is monitored

- Traffic generated by instances when they contact the Amazon DNS server. If you use your own DNS server, then all traffic to the DNS server is logged.
- Traffic generated by a Windows instance for Amazon Windows license activation.
- Traffic to and from 169.254.169.254 for instance metadata
- DHCP traffic
- Traffic to the reserved IP address for the default VPC router

NAT vs Bastion

- A NAT is used to provide internet traffic to EC2 instances in private subnets
- A Bastion is used to securely administer EC2 instances using SSH or RDP

`bastion host` -> `private server`

The Well Architected Framework

This section aggregates the well architected framework white paper

https://AWS.amazon.com/architecture/well-architected/
https://d0.AWSstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf

Best Practices

Business Benefits of the Cloud

- Almost zero upfront infrastructure investment
- Just-in-time infrastructure
- More efficient resource utilization
- Usage-based costing
- Reduced time to market

Technical Benefits of the Cloud

- Automation - "Scriptable Infrastructure"
- Auto-Scaling
- Proactive Scaling
- More Efficient Development lifecycle
- Improved Testability
- Disaster Recovery and Business Continuity
- "Overflow" the traffic to the cloud

Design for Failure

Rule of thumb:

Be a pessimist when designing architectures in the cloud - assume things will fail. In other words, always design, implement and deploy for automated recovery from failure.

Assume that...

- your hardware _will_ fail
- disaster _will_ strike your application
- you _will_ slammed with more than the expected number of requests per second someday.
- with time your application software _will_ fail too.

Being a pessimist, you end up thinking about recovery strategies during design time, which helps in designing overall system better.

Decouple Your Components

The key is to build components that do not have tight dependencies on each other, so that if once component were to die(fail), sleep (not respond) or remain busy(slow to respond) for some reason, the other components in the system are built so as to continue to work as if no failure is happening.

In essence, loose coupling isolates the various layers and components of your application so that each component interacts async with the others and treats them as a "black box".

For Example,

In the case of web application architecture, you can isolate the app server from the web server and from the db. The app server does not know about your web server and vice versa, this gives decoupling between these layers and there are not dependencies code wise or functional perspectives.

In the case of batch processing architecture, you can create async components that are independent of each other.

Implement Elasticity

The cloud brings a new concept of elasticity in your applications. Elasticity can be implemented in 3 ways.

1. Proactive Cyclic Scaling: Periodic scaling that occurs at a fixed interval (daily, weekly, monthly, quarterly)

2. Proactive Event-base Scaling: Scaling just when you are expecting a big surge of traffic requests due to a scheduled business event (new product launch, marketing campaigns)
3. Auto-scaling based on domain: By using monitoring service, you system can send triggers to take appropriate actions so that if scales up or down based on metrics (utilization of servers or network I/O)

The Well Architected Framework

What is the well architected framework?

This has been developed by the Solutions Architecture team based on their experience with helping AWS customers. The well architected framework is a set of questions that you can use to evaluate how well your architecture is aligned to AWS best practices.

5 Pillars of the Well Architected Framework

- Security
- Reliability
- Performance Efficiency
- Cost Optimization
- Operation Excellence

Structure of each pillar

- Design Principles
- Definition
- Best Practices
- Key AWS Services
- Resources

General Design Principles

- Stop guessing your capacity needs
- Test systems at production scale
- Automate to make architectural experimentation easier

- Allow for evolutionary architectures
- Data-driven architectures
- Improve through game days

Pillar 1 - Security

Design Principles

- Apply security at all layers!
- Enable traceability
- Automate responses to security events
- Focus on securing your system
- Automate security best practices

Definitions

Security in the cloud consists of 4 areas...

Data Protection

Before you begin to architect security practices across your environment, basic data classification should be in place. You should organize and classify your data in to segments such as publicly available, available to only members of your organization, available to only certain members of your organization, available only to the board etc.

You should also implement a least privilege access system so that people are only able to access what they need. However most importantly, you should encrypt everything where possible, whether it be at rest or in transit.

In AWS the following practices help to protect your data...

- AWS customers maintain full control over their data
- AWS makes it easier for you to encrypt your data and manage keys, including regular key rotation, which can be easily automated natively by AWS or maintained by a customer.

- Detailed logging is available that contains important content, such as file access and changes.
- AWS has designed storage systems for exceptional resiliency. As an example, Amazon S3 is designed for 11, 9's of durability. (if you store 10,000 objects with AWS S3, you can on average expect to incur a loss of a single object once every 10,000,000 years)
- Versioning, which can be part of a larger data lifecycle-management process, can protect against accidental overwrites, deletes and similar harm
- AWS never initiates the movement of data between regions. Content placed in a region will remain in that region unless the customer explicitly enables a feature or leverages a service that provides that functionality

What questions should you be asking yourself?

- How are you encrypting your data at rest?
- How are you encrypting your data in transit (SSL)?

Privilege Management

Privilege Management ensures that only authorized and authenticated users are able to access your resources, and only in a manner that is intended.

This can include

- Access Control Lists (ACLs)
- Role Based Access Controls
- Password Management (such as password rotation policies)

What questions should you be asking yourself?

- How are you protecting access to and use the AWS root account credentials?
- How are you defining roles and responsibilities of system users to control human access to the AWS Management Console and APIs?
- How are you limiting automated access (such as from applications, scripts, or 3rd party tools or services) to AWS resources?
- How are you managing keys and credentials?

Infrastructure Protection

Outside of Cloud, this is how you protect your data center. RFID controls, security, lockable cabinets, CCTV etc. Within AWS they handle this so Infrastructure Protection exists at a VPC level.

What questions should you be asking yourself?

- How are you enforcing network and host-level boundary protection?
- How are you enforcing AWS service level protection?
- How are you protecting the integrity of the OS on your AWS EC2 instances?

Detective Controls

You can use detective controls to detect or identify a security breach. AWS Services to achieve this include

- AWS Cloudtrail
- AWS CloudWatch
- AWS Config
- AWS S3
- AWS Glacier

What questions should you be asking yourself?

- How are you capturing and analyzing your logs?

Key AWS Services

1. Data Protection
   - Encrypt both in transit and at rest using - ELB, EBS, S3 and RDS
2. Privilege Management
   - IAM, MFA
3. Infrastructure Protection
   - VPC
4. Detective Controls
   - AWS Cloud Trail, AWS Config, AWS Cloud Watch

Pillar 2 - Reliability

The reliability pillar covers the ability of a system to recover from service or infrastructure outages/disruptions as well as the ability to dynamically acquire computing resources to meet demand.

- Test recovery procedures
- Automatically recover from failure - Netflix SimianArmy
- Scale horizontally increase aggregate system availability
- Stop guessing capacity

Definition

Reliability in the cloud consists of 3 areas...

1. Foundations
2. Change Management
3. Failure Management

Foundations

With AWS, they handle most of the foundations for you. The cloud is designed to be essentially limitless meaning that AWS handle the networking and compute requirements themselves. However, they do set the service limits to stop customers from accidentally over-provisioning resource

https://docs.AWS.amazon.com/general/latest/gr/AWS_service_limits.html

What questions should you be asking yourself?

- How are you managing AWS service limits for your account?
- How are you planning your network topology on AWS?
- Do you have an escalation path to deal with technical issues?

Change Management

You need to be aware of how change affects a system so that you can plan proactively around it. Monitoring allows you to detect any changes to your environment and react. In traditional systems, change control is done manually and are carefully coordinated with auditing.

With AWS things are a lot easier, you can use CloudWatch to monitor your environment and services such as autoscaling to automate change in response on your production environment

What questions should you be asking yourself?

- How does your system adapt to changes in demand?
- How are you monitoring AWS resources?
- How are you executing change management?

Failure Management

With cloud, you should always architect your systems with the assumptions that failure will occur. You should become aware of these failures, how they occurred, how to respond to them and then plan on how to prevent these from happening again.

What questions should you be asking yourself?

- How are you backing up your data?
- How does your system withstand component failures?
- How are you planning for recovery?

Key AWS Services

1. Foundations
   - IAM, VPC
2. Change Management
   - AWS CloudTrail
3. Failure Management
   - AWS CloudFormation

Pillar 3 - Performance Efficiency

The Performance Efficiency pillar focuses on how to use computing resources efficiency to meet your requirements and how to maintain that efficiency as demand and technology evolves.

Design Principles

- Democratize advanced technologies
- Go global in minutes
- Use server-less architectures
- Experiment more often

Definition

Performance Efficiency in the cloud consists of 4 areas...

Compute

When architecting your system, it is important to choose the right kind of server!!

Some applications require heavy CPU utilization, some require heavy memory utilization etc.

With AWS servers are virtualized and at the click of a button (or API call) you can change the type of server in which your environment is running on. You can even switch to running with no servers at all and use AWS Lambda.

What questions should you be asking yourself?

- How do you select the appropriate instance type for your system?
- How do you ensure that you continue to have the most appropriate instance type as new instance types and features are introduced?
- How do you monitor your instances post launch to ensure they are performing as expected?
- How do you ensure that the quantity of your instances match demand?

Storage

The optimal storage solutions for your environment depends on a number of factors

- Access Methods - Block, File or Object
- Patterns of Access - Random or Sequential
- Throughput Required
- Frequency of Access - Online, Offline or Archival
- Frequency of Update - Worm, Dynamic
- Availability Constraints
- Durability Constraints

At AWS the storage is virtualized. With S3 you can have 11 x 9's durability, Cross Region Replication etc. With EBS you can choose between storage mediums (SSD, Magnetic, PIOPS etc.).
You can also easily move volumes between the different types of storage mediums.

What questions should you be asking yourself?

- How do you select the appropriate storage solution for your system?
- How do you ensure that you continue to have the most appropriate storage solution as new storage solution features are launched?
- How do you monitor your storage solution to ensure it is performing as expected?
- How do you ensure that the capacity and throughput of your storage solutions matches demand?

Database

The optimal database solution depends on a number of factors. Do you need database consistency, do you need high availability, do you need No-SQL, do you need DR etc.

With AWS you get a LOT of options. RDS, DynamoDB, Redshift etc.

What questions should you be asking yourself?

- How do you select the appropriate database solution for your system?
- How do you ensure that you continue to have the most appropriate database solution and features as new database solution and features are launched?
- How do you monitor your databases to ensure performance is as expected?
- How do you ensure the capacity and throughput of your databases matches demand?

Space-time trade-off

Using AWS, you can use services such as RDS to add read replicas, reducing the load on your database and creating multiple copies of the database. This helps to lower latency.

You can use the global infrastructure to have multiple copies of your environment, in regions that is closest to our customer base. You can also use caching services such as ElasticCache or CloudFront to reduce latency.

What questions should you be asking yourself?

- How do you select the appropriate proximity and caching solutions for your system?
- How do you ensure that you continue to have the most appropriate proximity and caching solutions as new solutions are launched?
- How do you monitor your proximity and caching solutions to ensure performance is as expected?
- How do you ensure that the proximity and caching solutions you have matches demand?

Key AWS Services

1. Compute
   - Autoscaling
2. Storage
   - EBS, S3, Glacier
3. Database

  - RDS, DynamoDB, Redshift
4. Space-time Trade-Off
  - Cloudfront, ElasticCache, Direct Connect, RDS Read Replicas etc.

Pillar 4 - Cost Optimization

Use the Cost Optimization pillar to reduce your costs to a minimum and use those savings for other parts of your business. A cost-optimized system allows you to pay the lowest price possible while still achieving your business objectives.

Design Principles

- Transparently attribute expenditure
- Use managed services to reduce cost of ownership
- Trade capital expense for operating expense
- Benefit from economies of scale
- Stop spending money on data center operations

Definition

Cost optimization in the cloud consists of 4 areas...

Matched supply and demand

Try to optimally align supply with demand. Don’t over provision or under provision, instead as demand grows, so should your supply of compute resources. Think of things like Autoscaling which scale with demand.

Similarly, in a server-less context, use services such as Lambda that only execute when a request comes in.

Services such as CloudWatch can also help you keep track as to what your demand is.

What questions should you be asking yourself?

- How do you make sure your capacity matches but does not substantially exceed what you need?
- How are you optimizing your usage of AWS services?

Cost-effective resources

Using the correct instance type can be key to cost savings. For example, you might have a reporting process that is running on a t2-Micro and it takes 7 hours to complete. That same process could be run on a an m4.2xlarge in a manner of minutes. The result remains the same but the t2.micro is more expensive because it ran for longer.

A well architected system will use the most cost-efficient resources to reach the end business goal

What questions should you be asking yourself?

- Have you selected the appropriate resource types to meet your cost targets?
- Have you selected the appropriate pricing model to meet your cost targets?
- Are there managed services (higher level services that Amazon EC2, Amazon EBS) that you can use improve your ROI?

Expenditure Awareness

With cloud you no longer have to go out and get quotes on physical servers, choose a supplier, have those resources delivered, installed, made available etc. You can provision things within seconds, however this comes with its own issues.

Many organizations have different teams, each with their own AWS accounts. Being aware of what each team is spending and where is crucial to any well architected system.

You can use cost allocation tags to track this, billing alerts as well as consolidated billing.

What questions should you be asking yourself?

- What access control and procedures do you have in place to govern AWS costs?
- How are you monitoring usage and spending?
- How do you decommission resources that you no longer need, or stop resources that are temporarily not needed?
- How do you consider data-transfer charges when designing your architecture?

Optimizing Over Time

AWS moves FAST! There are hundreds of new services (and potentially 1000 new services this year). A service that you chose yesterday may not be the best service to be using today.

For example, consider MySQL RDS, Aurora was launched at re:invent 2014 and is now out of preview. Aurora may be a better option now for your business because of its performance and redundancy.

You should keep track of the changes made to AWS and constantly re-evaluate your existing architecture. You can do this by subscribing to AWS blog and by using services such as Trusted Advisor.

What questions should you be asking yourself?

- How do you manage and/or consider the adoption of new services?

Key AWS Services

1. Matched Supply and Demand
   - Autoscaling
2. Cost-effective resources
   - EC2 (reserved instances), AWS Trusted Advisor
3. Expenditure Awareness
   - CloudWatch Alarms, SNS
4. Optimizing Over Time
   - AWS Blog, AWS Trusted Advisor

Pillar 5 - Operational Excellence

The Operational Excellence pillar includes operational practices and procedures used to manage production workloads

This includes how planned changes are executed, as well as responses to unexpected operational events.

Change execution and responses should be automated. All processes and procedures of operational excellence should be documented, tested and regularly reviewed

Design Principles

- Perform operations with code
- Align operations processes to business objectives
- Make regular, small, incremental changes
- Test for responses to unexpected events
- Learn from operational events and failures
- Keep operations procedures current

Definition

There are 3 best practice areas of Operational Excellence in the cloud...

Preparation

Effective preparation is required to drive operational excellence. Operations checklists will ensure that workloads are ready for production operation, and prevent unintentional production promotion without effective preparation.

Workloads should have...

Runbooks - operations guidance that operations teams can refer to so they can perform normal daily tasks.

Playbooks - guidance for responding to unexpected operational events. Playbooks should include response plans, as well as escalation paths and stakeholder notifications.

In AWS there are several methods, services and features that can be used to support operational readiness and the ability to prepare for normal day-to-day operations as well as unexpected operational events.

CloudFormation can be used to ensure that environments contain all required resources when deployed to prod and the configuration of the environment is based on tested best practices, which reduces the opportunity for human error.

Autoscaling or other automated scaling mechanisms will allow workloads to automatically respond when business-related events affect operational needs.

AWS Config with the AWS Config rules feature create mechanisms to automatically track and respond to changes in your AWS workloads and environments

It is also important to use features like tagging to make sure all resources in a workload can be easily identified when needed during operations and response.

What preparation questions should you ask yourself for operational excellence?

- What best practices for cloud operations are your using?
- How are you doing configuration management for your workload?

Be sure that documentation doesn’t become stale or out of date! Documentation should be thorough!

Without application designs, environment configs, resource configs, response plans, and mitigation plans documentation is not complete.

If documentation is not updated and tested regularly, it will not be useful when unexpected operational events occur. If workloads are not reviewed before production, operations will be affected when undetected issues occur.

If resources are not documented, when operational events occur, determining how to respond will be more difficult while the correct resources are identified.

Operation

Operations should be standardized and manageable on a routine basis. The focus should be on automation, small frequent changes, regular QA testing, and defined mechanisms to track, audit, roll back and review changes.

Changes should not be large and infrequent, they should not require scheduled downtime, and they should not require manual execution. A wide range of logs and metrics that are based on key operational indicators for a workload should be collected and reviewed to ensure continuous operations.

What questions should you be asking yourself for operational excellence?

- How are you evolving your workload while minimizing the impact of change?
- How do you monitor your workload to ensure it is operating as expected?

Routine operations, as well as responses to unplanned events, should be automated. Manual processes for deployments, release management, changes and rollbacks should avoid.

Releases should not be large batches that are done infrequently.

Rollbacks are more difficult in large changes, and failing out have a rollback plan or the ability to mitigate failure impacts will prevent continuity of operations.

Align monitoring to business needs, so that the responses are effective at maintaining business continuity. Monitoring that is ad hoc and not centralized, with responses that are manual, will cause more impact to operations during unexpected events.

Response

Responses to unexpected operational events should be automated. This is not just for alerting but also for mitigation, remediation, rollback and recovery.

Alerts should be timely and should invoke escalations when response are not adequate to mitigate the impact of operational events.

QA mechanisms should be in place to automatically roll back failed deployments.

Responses should follow a pre-defined playbook that includes stakeholders, the escalation process and procedures. Escalation paths should be defined and include both functional and hierarchical escalation capabilities. Hierarchical escalation should be automated and escalated priority should result in stakeholder notifications.

What questions should you be asking yourself?

- How do respond to unplanned operational events?
- How is escalation managed when responding to unplanned operational events?

Key AWS Services

1. Preparation
  AWS Config provides a detailed inventory of your AWS resources and configuration, and continuously records configuration changes. AWS Service Catalog helps to create a standardized set of service offerings that are aligned to best practices. Designing workloads that use automation with services like Autoscaling, AWS SQS are good methods to ensure continuous operations in the event of unexpected operational events.
2. Operations
  AWS CodeCommit, AWS CodeDeploy and AWS CodePipeline can be used to manage and automate code changes to AWS workloads. Use AWS SDKs or 3rd party libs to automate operational changes. Use AWS CloudTrail to audit and track changes made to AWS environments
3. Responses
  Take advantage of all of the AWS CloudWatch service features for effective and automated responses. CloudWatch alarms can be used to set thresholds for alerting and notification and CloudWatch events can trigger notifications and automated responses.

Additional Exam Tips

Based on Student Feedback...

Kinesis

- Used to consume Big Data
- Stream large amounts of social media, news feeds, logs etc. to the cloud
- Think Kinesis when approached with big data questions

- Process large amounts of data;
  - Redshift for business intelligence
  - Elastic Map Reduce for Big Data Processing

EC2 - EBS Backed Volumes vs Instance Store Volumes

- EBS backed volumes are persistent
- Instance Store backed volumes are not persistent (ephemeral)
- EBS Volumes can be detached and reattached to other EC2 instances

- Instance store volumes cannot be detached and reattached to other instances - they exist only for the life of that instance.
- EBS volumes can be stopped; data will persist

- Instance store volumes cannot be stopped - if you do this the data will be wiped

- EBS Backed = Store Data Long Term
- Instance Store = Shouldn't be used for long-term data storage

OpsWork

- Orchestration Service that uses Chef
- Chef consists of recipes to maintain a consistent state
- Look for the term "chef" or "recipes" or "cook books" and think OpsWorks

Elastic Transcoder

- Media Transcoder in the cloud
- Convert media files from their original source format in to different formats that will play on smartphones, tablets, PC's, 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
- Pay based on the minutes that you transcode and the resolution at which you transcode

SWF Actors

- Workflow Starter - An application that can initiate (start) a workflow. Could be your e-commerce website when placing an order or a mobile app searching for bus times.
- 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.
- Activity Workers - Carry out the activity tasks

EC2 - Get Public IP Address

- Query the instances metadata:
  - `curl http://169.254.169.254/latest/meta-data`
  - `wget http://169.254.169.254/latest/meta-data`
  - Key thing to remember is that its an instances META-DATA, not user data

Consolidated Billing

AWS Organizations

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage.

Available in 2 feature sets
  - Consolidated Billing
  - All features

General Rules

- Paying account is independent
- Cannot access resources of other accounts
- All linked accounts are independent
- Currently a limit of 20 linked accounts - can add more

Advantages

- One bill per AWS account
- Very easy to track charges and allocate costs
- Volume Pricing

Best Practices

- Always enable MFA on root account
- Always use a strong and complex password on root account
- Paying account should be used for billing purposes only. Do not deploy resources in to paying account

Things to note

- Billing Alerts
  - When monitoring is enabled on the paying account the billing data for all linked accounts is included
  - You can still create billing alerts per individual account

- CloudTrail
  - Per AWS Account and is enabled per region

- Can consolidate logs using an S3 bucket
  1. Turn on CloudTrail in the paying account
  2. Create a bucket policy that allows cross account access
  3. Turn on CloudTrail in the other accounts and use the bucket in the paying account

Tips

- Consolidate billing allows you to get volume discounts on all your accounts.
- Unused reserved instances for EC2 are applied across the group
- CloudTrail is on a per account and per region basis but can be aggregated in to a single bucket in the paying account.

Cross Account Access

Many AWS customers use separate AWS accounts for their development and production resources. This separation allows them to cleanly separate different types of resources and can also provide some security benefits.

Cross account access makes it easier for you to work productively within a multi-account (or multi-role) AWS environment by making it easy for you to switch roles within the AWS Management Console.

You can sign in to the console using you IAM user name then switch the console using your IAM user name then switch the console to manage another account without having to enter (or remember) another user name and password

Resource Groups & Tags

Key Value Pairs attached to AWS resources

Metadata (data about data)

Tags can sometimes be inherited

- Autoscaling, CloudFormation and Elastic Beanstalk can create other resources

Resource groups make it easy to groups your resources using the tags that are assigned to them. You can group that share one or more tags.

Note: Container for resources

Resource groups contain information such as:

- Region
- Name
- Health Checks

Specific Information:
- For EC2 - Public and Private IP Addresses
- For ELB - Port Configurations
- For RDS - Database Engine etc.

VPC Peering

Note: Generally, not tested in Associate exams, only in Professional exams

What is VPC Peering?

VPC Peering is simply a connection between 2 VPCs that enables you to route traffic between them using private IP addresses.

Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a SINGLE REGION

AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on separate piece of physical hardware. There is no single point of failure for communication or bandwidth bottleneck.

VPC Peering Limitations

1. You cannot create a VPC peering connection between VPCs that have not matching or overlapping CIDR blocks i.e.. `10.0.0.0/16 -- X --> 10.0.0.0/24`
2. You cannot create a VPC peering connection between VPCs in different regions
3. VPC peering does not support transitive peering relationships.

Direct Connect

AWS Direct Connect makes it easier to establish a dedicated network connection from your premises to AWS.

Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections.

Main Benefits

- Reduce costs when using large volumes of traffic
- Increase reliability
- Increase bandwidth

How is Direct Connect different from a VPN?

VPN Connections can be configured in minutes and are a good solution if you have an immediate need, have low to moderate bandwidth requirements, and can tolerate the inherent variability in Internet-based connectivity.

AWS Direct Connect does not involve the Internet - instead, it uses dedicated, private network connections between your intranet and AWS VPC

Direct Connect Connections

Available in:
  - 10Gbs
  - 1Gbbs

- Sub 1 Gbps can be purchased through AWS Direct Connect Partners
- Uses Ethernet VLAN trunking (802.1Q)

STS - Security Token Service

Grants users limited and temporary access to AWS resources. Users can come from 3 sources

Federation (typically Active Directory)

- Uses Security Assertion Markup Language (SAML)
- Grants temp access based off the users Active Directory credentials. Does not need to be a user in IAM
- Single sign on allows users to log in to AWS console without assigning IAM credentials
- Federation with mobile apps - Facebook/AWS/Google/OpenID providers
- Cross Account Access - Access resources from one account to another

Key Terms

- Federation
    - Combining or joining a list of users in one domain (such as IAM) with a list of users in another domain (such as Active Directory, Facebook etc.)
- Identity Broker
    - A service that allows you to take an identity from point A and join it (federate it) with point B
- Identity Store
    - Services like Active Directory, Facebook, Google etc.
- Identities
    - A user of a service like Facebook etc.

SCENARIO!

```
You are hosting a company website on some EC2 web servers in your VPC. Users of the website must log in to the site which authenticates against the company’s active directory servers which are based on site at the company’s head quarters

Your VPC is connected to your company HQ via a secure IPSEC VPN. Once logged in the user can only have access to their own S3 bucket. How do you set this up?
```

SOLUTION!

1. Users enter credentials (username and password)

2. Application calls identity broker - broker captures username and passwords
3. Broker checks with LDAP directory server - validates credentials
4. Call to STS (security token service) - getFederationToken function using IAM credentials
5. STS confirms policy and gives permisssion to create new tokens - returns 4 values
    - Access Key
    - Secret Access Key
    - Token
    - Duration (lifetime of token)
6. 4 values are sent back to application via broker
7. Application makes call to S3
8. S3 uses IAM to validate credentials
9. Credentials validated via IAM

In the Exam!

1. Develop and Identity Broker to communicate with LDAP and AWS STS.
2. Identity Broker always authenticates with LDAP first, THEN with AWS STS
3. Application then gets temp access to AWS resources

Active Directory Tips

Exam Questions

QUESITON: _Can you authenticate with Active Directory?

ANSWER: Yes. Using SAML

QUESITON: _In what order do you authenticate to get the security credentials to log into Active Directory?

ANSWER: Authenticate with Active Directory first and then you are assigned the temp security credential.

Workspaces

It's basically a VDI (virtual development infrastructure). A Workspace is a cloud-based replacement for a traditional desktop.

A Workspace is available as a bundle of compute resources, storage space, and software application access that allow a user to perform day-to-day tasks just like using a traditional desktop.

A user can connect to a Workspace from any supported device (PC, Mac, Chromebook, iPad, KindleFire or Android Tablets) using free Amazon Workspaces client application and credentials set up by an administrator, or their existing Active Directory credentials if Amazon Workspaces is integrated with an existing Active Directory domain.

Quick Facts

- Windows 7 experience, provided by Windows Server 2008 R2
- By default, users can personalize their workspaces. This can be locked down by an admin however
- By default, you will be given local admin access, so you can install your own applications
- Workspaces are persistent
- All data on the D:\ is backed up every 12 hours
- You do not need an AWS account to login into workspaces

ECS

- ECS is a regional service that you can use in 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 eliminates the need for you to operate your own cluster management and config management systems, or to worry about scaling your management infrastructure.

- ECS can also be used to create a consistent deployment and build experience, manage and scale batch and ETL workloads, and build sophisticated application architectures on a microservice level.

ECR (Elastic Container Registry)

- Managed AWS Docker registry service that is secure, scalable and reliable.
- Supports private Docker repos with resource-based permissions using AWS IAM so that specific users or EC2 instances can access repos and images.
- Developers can use the Docker CLI to push, pull and manage images.

ECS Task Definitions

- A Task Definition is required to run Docker containers in ECS.
- Task Definitions are text files in JSON format that describe one or more containers that form your application.
- Some of the prams you can specify in a task definition include:
  - Which Docker images to yes with the containers in your task
  - How much CPU and memory to use with each container
  - Whether containers are linked together in a task
- The Docker networking mode to use for the containers in your task
- What (if any) ports from the container are mapped to the host container service
- Whether the task should continue to run if the container finishes or fails
- The command the container should run when it is started
- What (if any) env variables should be passed to the container when it starts.
- Any data volumes that should be used with containers in the task
- What (if any) IAM role your tasks should use for permissions

ECS Services

- An ECS service allows you to run and maintain a specified number (or, the "desired count") of instances of a task definition simultaneously in and ECS cluster
- Think of services like AutoScaling groups for ECS
- If a task should fail or stop, the ECS service scheduler launches another instance of your task definition to replace it and maintain the desired count of tasks in the service.

ECS Clusters

- An 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, but you can create multiple clusters in an account to keep your resources separate.
- 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

- Service Scheduler:
    - Ensures that the specific number of tasks are constantly running and reschedules tasks when a task fails (for example, if the underlying container instance fails for some reason)
    - Can ensure tasks are registered against and ELB
- Custom Scheduler:
    - You can create your own schedulers that meet your business needs.
    - Leverage 3rd party schedulers such as Blox
- The ECS schedulers leverage the same cluster state information provided by the ECS API to make appropriate placement decisions

ECS Container Agent

ECS Container Agent allows container instances to connect to your cluster. ECS Container Agent is included in the ECS optimized AMI, but you can also install it on any EC2 instance that supports ECS specs. ECS Container Agent is only supported on EC2 instances.

- Pre-installed on special ECS AMIs
- Linux based:
    - Works with AWS Linux, Ubuntu, Redhat, CentOS, ets
    - Will not work with Windows

ECS Security

- IAM Roles:
    - EC2 instances use an IAM role to access ECS
    - ECS tasks use an IAM role to access services and resources
- 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 instances in your ECS cluster

ECS Limits

- Soft Limits:
    - Clusters per Region (default = 100)
    - Instances per Cluster (default = 100)
    - Services per Cluster (default = 100)
- Hard Limits:
    - One Load Balancer per Service
    - 1000 Tasks per Service ("desired")
    - Max 10 Containers per Task Definition
    - Max 10 Tasks per Instance (host)