AWS SAA-C02 Study Guide

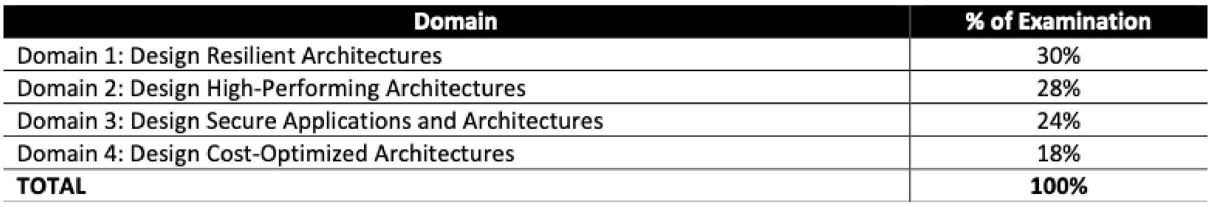
These notes helped me pass the newer AWS Certified Solutions Architect - Associate exam. In order to pass, I took the relevant A Cloud Guru course twice and Andrew Brown's walkthrough once, I kept up-to-date with the discussions in r/AWSCertifications, and I practiced with the fantastic Tutorials Dojo practice exams. Also, Stephane Maarek's course is highly recommended as it is consistently referred to across the web and his YouTube playlist was a great refresher to have just prior to the exam. Finally, I read many of the FAQs for the most critical services which are included in the list of recommended reading in the Introduction section below.

Table of Contents

Introduction

The official AWS Solutions Architect - Associate (SAA-C02) exam guide

Exam Content Breakdown:



Domain	% of Examination
Domain 1: Design Resilient Architectures	30%
Domain 2: Design High-Performing Architectures	28%
Domain 3: Design Secure Applications and Architectures	24%
Domain 4: Design Cost-Optimized Architectures	18%
TOTAL	**100%**

Domain 1: Design Resilient Architectures

1.1 - Design a multi-tier architecture solution

1.2 - Design highly available and/or fault-tolerant architectures

1.3 - Design decoupling mechanisms using AWS services

1.4 - Choose appropriate resilient storage

Domain 2: Design High-Performing Architectures

2.1 - Identify elastic and scalable computesolutions for a workload

2.2 - Select high-performingand scalable storage solutions for a workload

2.3 - Select high-performingnetworking solutions for a workload

2.4 - Choose high-performingdatabase solutions for a workload

Domain 3: Design Secure Applications and Architectures

3.1 - Design secure access to AWS resources

3.2 - Design secure application tiers

3.3 - Select appropriate data security options

Domain 4: Design Cost-Optimized Architectures

4.1 - Identify cost-effective storage solutions

4.2 - Identify cost-effective compute and database services

4.3 - Design cost-optimized network architectures

Recommended Reading:

You can cover a lot of ground by skimming over what you already know or what you can infer to be true. In particular, read the first sentence of each paragraph and if you have no uncertainty about what is being said in that sentence, move on to the first sentence of the next paragraph. Take notes whenever necessary.

1. AWS Well-Architected Framework
2. Amazon VPC FAQs
3. AWS Autoscaling FAQs
4. Amazon EC2 FAQs
5. Amazon EC2 Auto Scaling FAQs
6. Amazon EBS FAQs
7. Elastic network interfaces
8. Amazon S3 FAQs
9. Elastic Load Balancing FAQs
10. Amazon Route 53 FAQs
11. AWS Storage Gateway FAQs
12. Amazon EFS FAQs
13. Amazon FSx for Windows File Server FAQs
14. Amazon FSx for Lustre FAQs
15. AWS Organizations FAQs

Identity Access Management (IAM)

IAM Simplified:

IAM offers a centralized hub of control within AWS and integrates with all other AWS Services. IAM comes with the ability to share access at various levels of permission and it supports the ability to use identity federation (the process of delegating authentication to a trusted external party like Facebook or Google) for temporary or limited access. IAM comes with MFA support and allows you to set up custom password rotation policy across your entire organiation. It is also PCI DSS compliant (passes government mandated credit card security regulations).

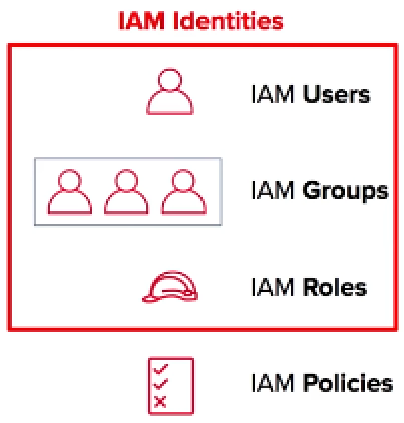
IAM Entities:

Users - any individual end user such as an employee, system architect, CTO, etc.

Groups - any collection of similar people with shared permissions such as system administrators, HR employees, finance teams, etc. Each user within their specified group will inherit the permissions set for the group.

Roles - any software service that needs to be granted permissions to do its job, e.g- AWS Lambda needing write permissions to S3 or a fleet of EC2 instances needing read permissions from a RDS MySQL database.

Policies - the documented rulesets that are applied to grant or limit access. In order for users, groups, or roles to properly set permissions, they use policies. Policies are written in JSON and you can either use custom policies for your specific needs or use the default policies set by AWS.



IAM Policies are separated from the other entities above because they are not an IAM Identity. Instead, they are attached to IAM Identities so that the IAM Identity in question can perform its neccessary function.

IAM Key Details:

- IAM is a global AWS services that is not limited by regions. Any user, group, role or policy is accessible globally.
- The root account with complete admin access is the account used to sign up for AWS. Therefore, the email address used to create the AWS account for use should probably be the official company email address.
- New users have no permissions when their accounts are first created. This is a secure way of delegating access as permissions must be intentionally granted.
- When joining the AWS ecosystem for the first time, new users are supplied an access key ID and a secret access key ID when you grant them programmatic access. These are created just once specifically for the new user to join, so if they are lost simply generate a new access key ID and a new secret access key ID. Access keys are only used for the AWS CLI and SDK so you cannot use them to access the console.
- When creating your AWS account, you may have an existing identity provider internal to your company that offers Single Sign On (SSO). If this is the case, it is useful, efficient, and entirely possible to reuse your existing identities on AWS. To do this, you let an IAM role be assumed by one of the Active Directories. This is because the IAM ID Federation feature allows an external service to have the ability to assume an IAM role.
- IAM Roles can be assigned to a service, such as an EC2 instance, prior to its first use/creation or after its been in used/created. You can change permissions as many times as you need. This can all be done by using both the AWS console and the AWS command line tools.
- You cannot nest IAM Groups. Individual IAM users can belong to multiple groups, but creating subgroups so that one IAM Group is embedded inside of another IAM Group is not possible.
- With IAM Policies, you can easily add tags that help define which resources are accessible by whom. These tags are then used to control access via a particular IAM policy. For example, production and development EC2 instances might be tagged as such. This would ensure that people who should only be able to access development instances cannot access production instances.

Priority Levels in IAM:

- **Explicit Deny**: Denies access to a particular resource and this ruling cannot be overruled.
- **Explicit Allow**: Allows access to a particular resource so long as there is not an associated Explicit Deny.
- **Default Deny (or Implicit Deny)**: IAM identities start off with no resource access. Access instead must be granted.

Simple Storage Service (S3)

S3 Simplified:

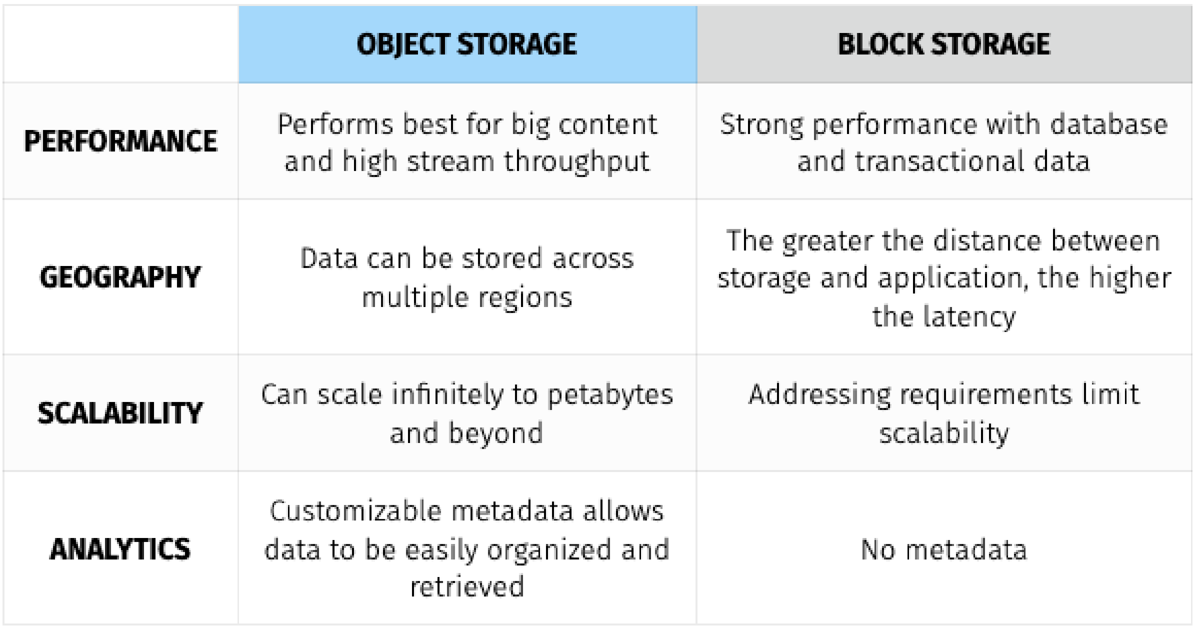
S3 provides developers and IT teams with secure, durable, and highly-scalable object storage. Object storage, as opposed to block storage, is a general term that refers to data composed of three things:

1.) the data that you want to store

2.) an expandable amount of metadata

3.) a unique identifier so that the data can be retrieved

This makes it a perfect candidate to host files or directories and a poor candidate to host databases or operating systems. The following table highlights key differences between object and block storage:



	OBJECT STORAGE	BLOCK STORAGE
PERFORMANCE	Performs best for big content and high stream throughput	Strong performance with database and transactional data
GEOGRAPHY	Data can be stored across multiple regions	The greater the distance between storage and application, the higher the latency
SCALABILITY	Can scale infinitely to petabytes and beyond	Addressing requirements limit scalability
ANALYTICS	Customizable metadata allows data to be easily organized and retrieved	No metadata

Data uploaded into S3 is spread across multiple files and facilities. The files uploaded into S3 have an upper-bound of 5TB per file and the number of files that can be uploaded is virtually limitless. S3 buckets, which contain all files, are named in a universal namespace so uniqueness is required. All successful uploads will return an HTTP 200 response.

S3 Key Details:

- Objects (regular files or directories) are stored in S3 with a key, value, version ID, and metadata. They can also contain torrents and subresources for access control lists which are basically permissions for the object itself.
- The data consistency model for S3 ensures immediate read access for new objects after the initial PUT requests. These new objects are introduced into AWS for the first time and thus do not need to be updated anywhere so they are available immediately.
- The data consistency model for S3 ensures eventual read consistency for PUTS and DELETES of already existing objects. This is because the change takes a little time to propagate across the entire Amazon network.
- Because of the eventual consistency model when updating existing objects in S3, those updates might not be immediately reflected. As object updates are made to the same key, an older version of the object might be provided back to the user when the next read request is made.
- Amazon guarantees 99.999999999% (or 11 9s) durability for all S3 storage classes except its Reduced Redundancy Storage class.
- S3 comes with the following main features:

  1.) tiered storage and pricing variability

  2.) lifecycle management to expire older content

  3.) versioning for version control

  4.) encryption for privacy

  5.) MFA deletes to prevent accidental or malicious removal of content

  6.) access control lists & bucket policies to secure the data

- S3 charges by:

  1.) storage size

  2.) number of requests

  3.) storage management pricing (known as tiers)

  4.) data transfer pricing (objects leaving/entering AWS via the internet)

  5.) transfer acceleration (an optional speed increase for moving objects via Cloudfront)

  6.) cross region replication (more HA than offered by default

- Bucket policies secure data at the bucket level while access control lists secure data at the more granular object level.
- By default, all newly created buckets are private.

- S3 can be configured to create access logs which can be shipped into another bucket in the current account or even a separate account all together. This makes it easy to monitor who accesses what inside S3.
- There are 3 different ways to share S3 buckets across AWS accounts:

  1.) For programmatic access only, use IAM & Bucket Policies to share entire buckets

  2.) For programmatic access only, use ACLs & Bucket Policies to share objects

  3.) For access via the console & the terminal, use cross-account IAM roles

- S3 is a great candidate for static website hosting. When you enable static website hosting for S3 you need both an index.html file and an error.html file. Static website hosting creates a website endpoint that can be accessed via the internet.
- When you upload new files and have versioning enabled, they will not inherit the properties of the previous version.

S3 Storage Classes:

S3 Standard - 99.99% availability and 11 9s durability. Data in this class is stored redundantly across multiple devices in multiple facilities and is designed to withstand the failure of 2 concurrent data centers.

S3 Infrequently Accessed (IA) - For data that is needed less often, but when it is needed the data should be available quickly. The storage fee is cheaper, but you are charged for retrieval.

S3 One Zone Infrequently Accessed (an improvement of the legacy RRS / Reduced Redundancy Storage) - For when you want the lower costs of IA, but do not require high availability. This is even cheaper because of the lack of HA.

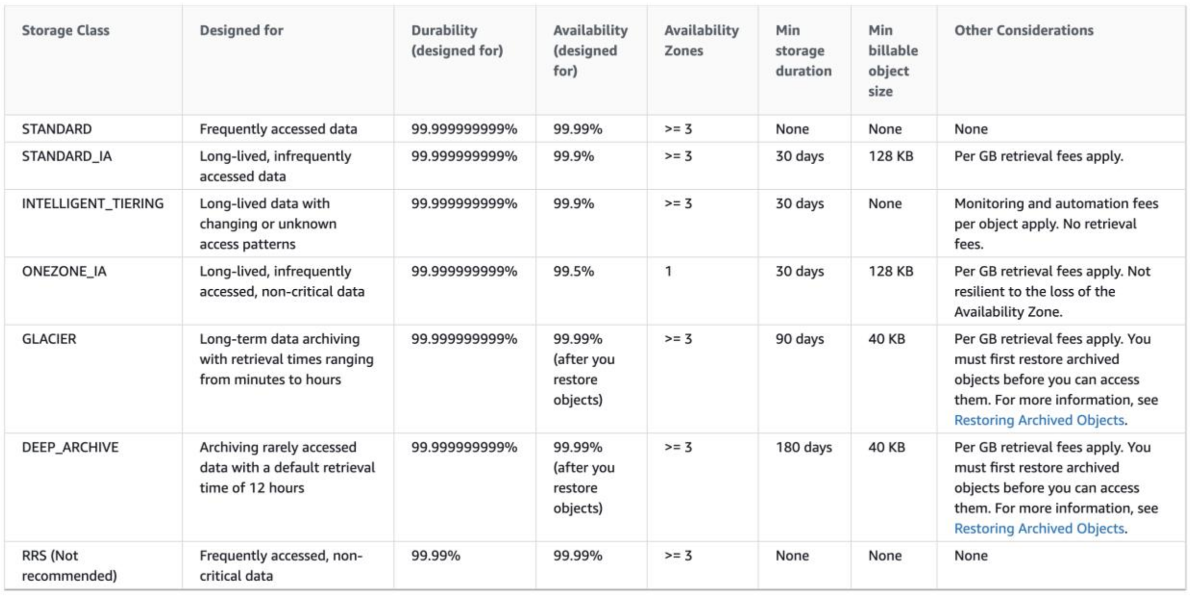
S3 Intelligent Tiering - Uses built-in ML/AI to determine the most cost-effective storage class and then automatically moves your data to the appropriate tier. It does this without operational overhead or performance impact.

S3 Glacier - low-cost storage class for data archiving. This class is for pure storage purposes where retrieval isn’t needed often at all. Retrieval times range from minutes to hours. There are differing retrieval methods depending on how acceptable the default retrieval times are for you:

```
Expedited: 1 - 5 minutes, but this option is the most expensive.
Standard: 3 - 5 hours to restore.
Bulk: 5 - 12 hours. This option has the lowest cost and is good for a large
set of data.
```

The Expedited duration listed above could possibly be longer during rare situations of unusually high demand across all of AWS. If it is absolutely critical to have quick access to your Glacier data under all circumstances, you must purchase *Provisioned Capacity*. Provisioned Capacity guarentees that Expedited retrievals always work within the time constraints of 1 to 5 minutes.

S3 Deep Glacier - The lowest cost S3 storage where retrieval can take 12 hours.



Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	40 KB	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	40 KB	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

S3 Encryption:

S3 data can be encrypted both in transit and at rest.

Encryption In Transit: When the traffic passing between one endpoint to another is indecipherable. Anyone eavesdropping between server A and server B won't be able to make sense of the information passing by. Encryption in transit for S3 is always achieved by SSL/TLS.

Encryption At Rest: When the immobile data sitting inside S3 is encrypted. If someone breaks into a server, they still won't be able to access encrypted info within that server. Encryption at rest can be done either on the server-side or the client-side. The server-side is when S3 encrypts your data as it is being written to disk and decrypts it when you access it. The client-side is when you personally encrypt the object on your own and then upload it into S3 afterwards.

You can encrypted on the AWS supported server-side in the following ways:

- **S3 Managed Keys / SSE - S3 (server side encryption S3)** - when Amazon manages the encryption and decryption keys for you automatically. In this scenario, you concede a little control to Amazon in exchange for ease of use.
- **AWS Key Management Service / SSE - KMS** - when Amazon and you both manage the encryption and decryption keys together.
- **Server Side Encryption w/ customer provided keys / SSE - C** - when I give Amazon my own keys that I manage. In this scenario, you concede ease of use in exchange for more control.

S3 Versioning:

- When versioning is enabled, S3 stores all versions of an object including all writes and even deletes.
- It is a great feature for implictly backing up content and for easy rollbacks in case of human error.
- It can be thought of as analogous to Git.
- Once versioning is enabled on a bucket, it cannot be disabled - only suspended.
- Versioning integrates w/ lifecycle rules so you can set rules to expire or migrate data based on their version.
- Versioning also has MFA delete capability to provide an additional layer of security.

S3 Lifecycle Management:

- Automates the moving of objects between the different storage tiers.
- Can be used in conjunction with versioning.
- Lifecycle rules can be applied to both current and previous versions of an object.

S3 Cross Region Replication:

- Cross region replication only work if versioning is enabled.
- When cross region replication is enabled, no pre-existing data is transferred. Only new uploads into the original bucket are replicated. All subsequent updates are replicated.
- When you replicate the contents of one bucket to another, you can actually change the ownership of the content if you want. You can also change the storage tier of the new bucket with the replicated content.
- When files are deleted in the original bucket (via a delete marker as versioning prevents true deletions), those deletes are not replicated.
- Cross Region Replication Overview
- What is and isn't replicated such as encrypted objects, deletes, items in glacier, etc.

S3 Transfer Acceleration:

- Transfer acceleration makes use of the CloudFront network by sending or receiving data at CDN points of presence (called edge locations) rather than slower uploads or downloads at the origin.
- This is accomplished by uploading to a distinct URL for the edge location instead of the bucket itself. This is then transferred over the AWS network backbone at a much faster speed.
- You can test transfer acceleration speed directly in comparison to regular uploads.

S3 Event Notications:

The Amazon S3 notification feature enables you to receive and send notifications when certain events happen in your bucket. To enable notifications, you must first configure the events you want Amazon S3 to publish (new object added, old object deleted, etc.) and the destinations where you want Amazon S3 to send the event notifications. Amazon S3 supports the following destinations where it can publish events:

- **Amazon Simple Notification Service (Amazon SNS)** - A web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients.
- **Amazon Simple Queue Service (Amazon SQS)** - SQS offers reliable and scalable hosted queues for storing messages as they travel between computers.
- **AWS Lambda** - AWS Lambda is a compute service where you can upload your code and the service can run the code on your behalf using the AWS infrastructure. You package up and upload your custom code to AWS Lambda when you create a Lambda function. The S3 event triggering the Lambda function also can serve as the code's input.

S3 and ElasticSearch:

- If you are using S3 to store log files, ElasticSearch provides full search capabilities for logs and can be used to search through data stored in an S3 bucket.
- You can integrate your ElasticSearch domain with S3 and Lambda. In this setup, any new logs received by S3 will trigger an event notification to Lambda, which in turn will then run your application code on the new log data. After your code finishes processing, the data will be streamed into your ElasticSearch domain and be available for observation.

Maximizing S3 Read/Write Performance:

- If the request rate for reading and writing objects to S3 is extremely high, then you can use hash keys or random strings to prefix the object's name. In such cases, the partitions used to store the objects will be better distributed and therefore will allow better read/write performance on your objects.
- If your S3 data is receiving a high number of GET requests from users, you should consider using Amazon CloudFront for performance optimization. By integrating CloudFront with S3, you can distribute content via CloudFront's cache to your users for lower latency and a higher data transfer rate. This also has the added bonus of sending fewer direct requests to S3 which will reduce costs. For example, suppose that you have a few objects that are very popular. CloudFront fetches those objects from S3 and caches them. CloudFront can then serve future requests for the objects from its cache, reducing the total number of GET requests it sends to Amazon S3.
- More information on how to ensure high performance in S3

S3 Server Access Logging:

- Server access logging provides detailed records for the requests that are made to a bucket. Server access logs are useful for many applications. For example, access log information can be useful in security and access audits. It can also help you learn about your customer base and better understand your Amazon S3 bill.
- By default, logging is disabled. When logging is enabled, logs are saved to a bucket in the same AWS Region as the source bucket.
- Each access log record provides details about a single access request, such as the requester, bucket name, request time, request action, response status, and an error code, if relevant.
- It works in the following way:

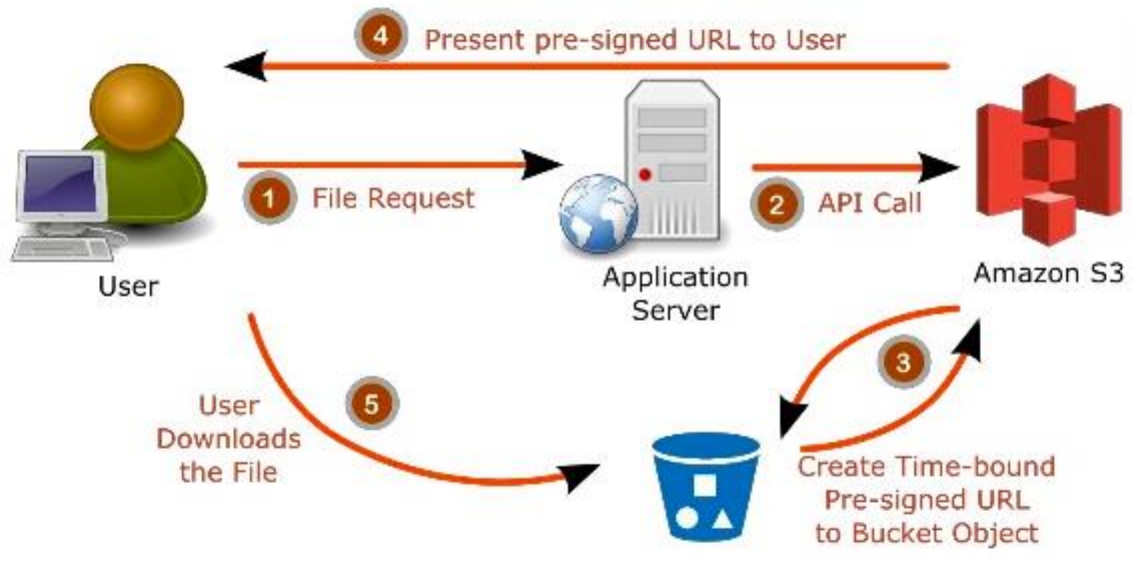
    - S3 periodically collecting access log records of the bucket you want to monitor
    - S3 then consolidates those records into log files
    - S3 finally uploads the log files to your secondary monitoring bucket as log objects

S3 Multipart Upload:

- Multipart upload allows you to upload a single object as a set of parts. Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order.
- If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object.
- Possible reasons for why you would want to use Multipart upload:
    - Multipart upload delivers the ability to begin an upload before you know the final object size.
    - Multipart upload delivers improved throughput.
    - Multipart upload delivers the ability to pause and resume object uploads.
    - Multipart upload delivers quick recovery from network issues.
- You can use an AWS SDK to upload an object in parts. Alternatively, you can perform the same action via the AWS CLI.

S3 Pre-signed URLs:

- All S3 objects are private by default, however the object owner of a private bucket with private objects can optionally share those objects with without having to change the permissions of the bucket to be public.
- This is done by creating a pre-signed URL. Using your own security credentials, you can grant time-limited permission to download or view your private S3 objects.
- When you create a pre-signed URL for your S3 object, you must do the following:
    - provide your security credentials
    - specify a bucket
    - specify an object key
    - specify the HTTP method (GET to download the object)
    - specift the expiration date and time.
- The pre-signed URLs are valid only for the specified duration and anyone who receives the pre-signed URL within that duration can then access the object.
- The following diagram highlights how Pre-signed URLs work:


4 Present pre-signed URL to User
1 File Request
2 API Call
User
Application Server
Amazon S3
3 Create Time-bound Pre-signed URL to Bucket Object
5 User Downloads the File

S3 Select:

- S3 Select is an Amazon S3 feature that is designed to pull out only the data you need from an object, which can dramatically improve the performance and reduce the cost of applications that need to access data in S3.
- Most applications have to retrieve the entire object and then filter out only the required data for further analysis. S3 Select enables applications to offload the heavy lifting of filtering and accessing data inside objects to the Amazon S3 service.
- As an example, let's imagine you're a developer at a large retailer and you need to analyze the weekly sales data from a single store, but the data for all 200 stores is saved in a new GZIP-ed CSV every day.
  - Without S3 Select, you would need to download, decompress and process the entire CSV to get the data you needed.
  - With S3 Select, you can use a simple SQL expression to return only the data from the store you're interested in, instead of retrieving the entire object.
- By reducing the volume of data that has to be loaded and processed by your applications, S3 Select can improve the performance of most applications that frequently access data from S3 by up to 400% because you're dealing with significantly less data.

CloudFront

CloudFront Simplified:

The AWS CDN service is called CloudFront. It serves up cached content and assets for the increased global performance of your application. The main components of CloudFront are the edge locations (cache endpoints), the origin (original source of truth to be cached such as an EC2 instance, an S3 bucket, an Elastic Load Balancer or a Route 53 config), and the distribution (the arrangement of edge locations from the origin or basically the network itself). More info on CloudFront's features

CloudFront Key Details:

- When content is cached, it is done for a certain time limit called the Time To Live, or TTL, which is always in seconds
- If needed, CloudFront can serve up entire websites including dynamic, static, streaming and interactive content.
- Requests are always routed and cached in the nearest edge location for the user, thus propagating the CDN nodes and guaranteeing best performance for future requests.
- There are two different types of distributions:
    - **Web Distribution**: web sites, normal cached items, etc
    - **RTMP**: streaming content, adobe, etc
- Edge locations are not just read only. They can be written to which will then return the write value back to the origin.
- Cached content can be manually invalidated or cleared beyond the TTL, but this does incur a cost.
- You can invalidate the distribution of certain objects or entire directories so that content is loaded directly from the origin everytime. Invalidating content is also helpful when debugging if content pulled from the origin seems correct, but pulling that same content from an edge location seems incorrect.
- You can set up a failover for the origin by creating an origin group with two origins inside. One origin will act as the primary and the other as the secondary. CloudFront will automatically switch between the two when the primary origin fails.
- Amazon CloudFront delivers your content from each edge location and offers a Dedicated IP Custom SSL feature. SNI Custom SSL works with most modern browsers.
- If you run PCI or HIPAA-compliant workloads and need to log usage data, you can do the following:
    - Enable CloudFront access logs.
    - Capture requests that are sent to the CloudFront API.
- An Origin Access Identity (OAI) is used for sharing private content via CloudFront. The OAI is a virtual user that will be used to give your CloudFront distribution permission to fetch a private object from your origin (e.g. S3 bucket).

CloudFront Signed URLs and Signed Cookies:

- CloudFront signed URLs and signed cookies provide the same basic functionality: they allow you to control who can access your content. These features exist because many companies that distribute content via the internet want to restrict access to documents, business data, media streams, or content that is intended for selected users. As an example, users who have paid a fee should be able to access private content that users on the free tier shouldn't.
- If you want to serve private content through CloudFront and you're trying to decide whether to use signed URLs or signed cookies, consider the following:
    - Use signed URLs for the following cases:
        - You want to use an RTMP distribution. Signed cookies aren't supported for RTMP distributions.
        - You want to restrict access to individual files, for example, an installation download for your application.
        - Your users are using a client (for example, a custom HTTP client) that doesn't support cookies.
    - Use signed cookies for the following cases:

    - You want to provide access to multiple restricted files. For example, all of the files for a video in HLS format or all of the files in the paid users' area of a website.
    - You don't want to change your current URLs.

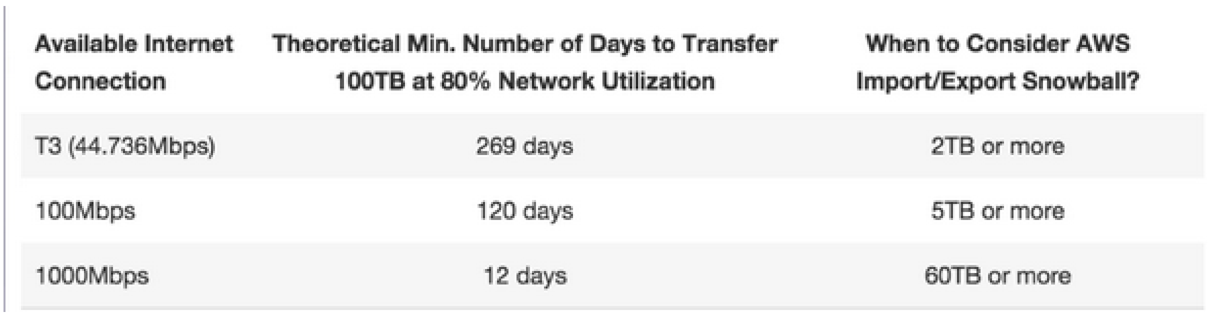
Snowball

Snowball Simplified:

Snowball is a giant physical disk that is used for migrating high quantities of data into AWS. It is a peta-byte scale data transport solution. Using a large disk like Snowball helps to circumvent common large scale data transfer problems such as high network costs, long transfer times, and security concerns. Snowballs are extremely secure by design and once the data transfer is complete, the snowballs are wiped clean of your data.

Snowball Key Details:

- Snowball is a strong choice for a data transfer job if you need a secure and quick data transfer ranging in the terabytes to many petabytes into AWS.
- Snowball can also be the right choice if you don’t want to make expensive upgrades to your existing network infrastructure, if you frequently experience large backlogs of data, if you're located in a physically isolated environment, or if you're in an area where high-speed internet connections are not available or cost-prohibitive.
- As a rule of thumb, if it takes more than one week to upload your data to AWS using the spare capacity of your existing internet connection, then you should consider using Snowball.
- For example, if you have a 100 Mb connection that you can solely dedicate to transferring your data and you need to transfer 100 TB of data in total, it will take more than 100 days for the transfer to complete over that connection. You can make the same transfer in about a week by using multiple Snowballs.
- Here is a reference for when Snowball should be considered based on the number of days it would take to make the same transfer over an internet connection:



Available Internet Connection	Theoretical Min. Number of Days to Transfer 100TB at 80% Network Utilization	When to Consider AWS Import/Export Snowball?
T3 (44.736Mbps)	269 days	2TB or more
100Mbps	120 days	5TB or more
1000Mbps	12 days	60TB or more

Snowball Edge and Snowmobile:

- Snowball Edge is a specific type of Snowball that comes with both compute *and* storage capabilities via AWS Lambda and specific EC2 instance types. This means you can run code within your snowball while your data is en route to an Amazon data center. This enables support of local workloads in remote or offline locations and as a result, Snowball Edge does not need to be limited to a data transfer service. An interesting use case is with airliners. Planes sometimes fly with snowball edges onboard so they can store large amounts of flight data and compute necessary functions for the plane's own systems. Snowball Edges can also be clustered locally for even better performance.

- Snowmobile is an exabyte-scale data transfer solution. It is a data transport solution for 100 petabytes of data and is contained within a 45-foot shipping container hauled by a semi-truck. This massive transfer makes sense if you want to move your entire data center with years of data into the cloud.

Storage Gateway

Storage Gateway Simplified:

Storage Gateway is a service that connects on-premise environments with cloud-based storage in order to seamlessly and securely integrate an on-prem application with a cloud storage backend. and Volume Gateway as a way of storing virtual hard disk drives in the cloud.

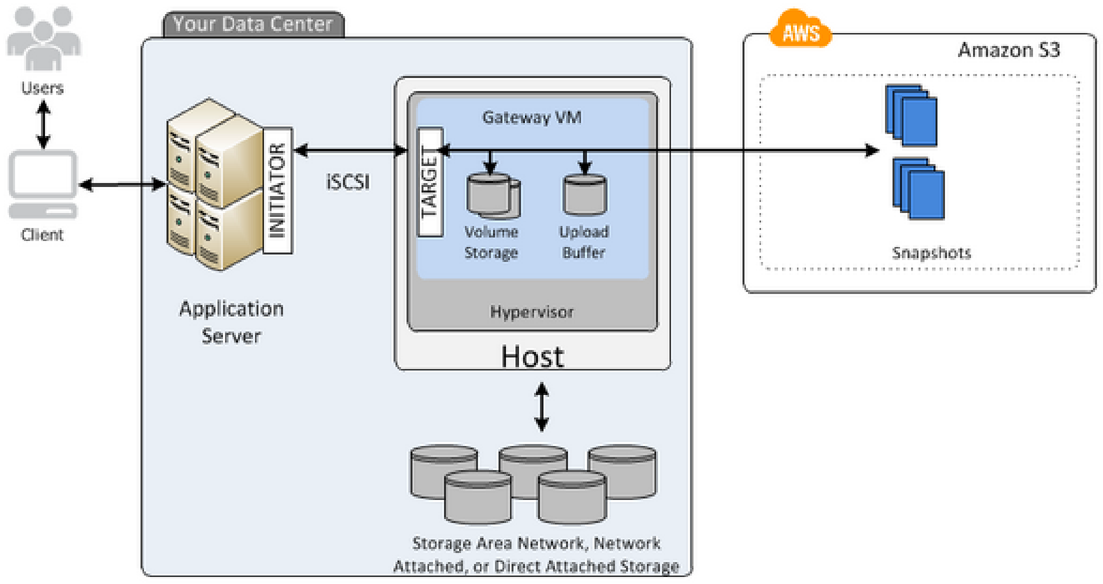
Storage Gateway Key Details:

- The Storage Gateway service can either be a physical device or a VM image downloaded onto a host in an on-prem data center. It acts as a bridge to send or receive data from AWS.
- Storage Gateway can sit on top of VMWare's ESXi hypervisor for Linux machines and Microsoft's Hyper-V hypervisor for Windows machines.
- The three types of Storage Gateways are below:
  - **File Gateway** - Operates via NFS or SMB and is used to store files in S3 over a network filesystem mount point in the supplied virtual machine. Simply put, you can think of a File Gateway as a file system mount on S3.
  - **Volume Gateway** - Operates via iSCSI and is used to store copies of hard disk drives or virtual hard disk drives in S3. These can be achieved via *Stored Volumes* or *Cached Volumes*. Simply put, you can think of Volume Gateway as a way of storing virtual hard disk drives in the cloud.
  - **Tape Gateway** - Operates as a Virtual Tape Library
- Relevant file information passing through Storage Gateway like file ownership, permissions, timestamps, etc. are stored as metadata for the objects that they belong to. Once these file details are stored in S3, they can be managed natively. This mean all S3 features like versioning, lifecycle management, bucket policies, cross region replication, etc. can be applied as a part of Storage Gateway.
- Applications interfacing with AWS over the Volume Gateway is done over the iSCSI block protocol. Data written to these volumes can be asynchronously backed up into AWS Elastic Block Store (EBS) as point-in-time snapshots of the volumes' content. These kind of snapshots act as incremental backups that capture only changed state similar to a pull request in Git. Further, all snapshots are compressed to reduce storage costs.
- Tape Gateway offers a durable, cost-effective way of archiving and replicating data into S3 while getting rid of tapes (old-school data storage). The Virtual Tape Library, or VTL, leverages existing tape-based backup infrastructure to store data on virtual

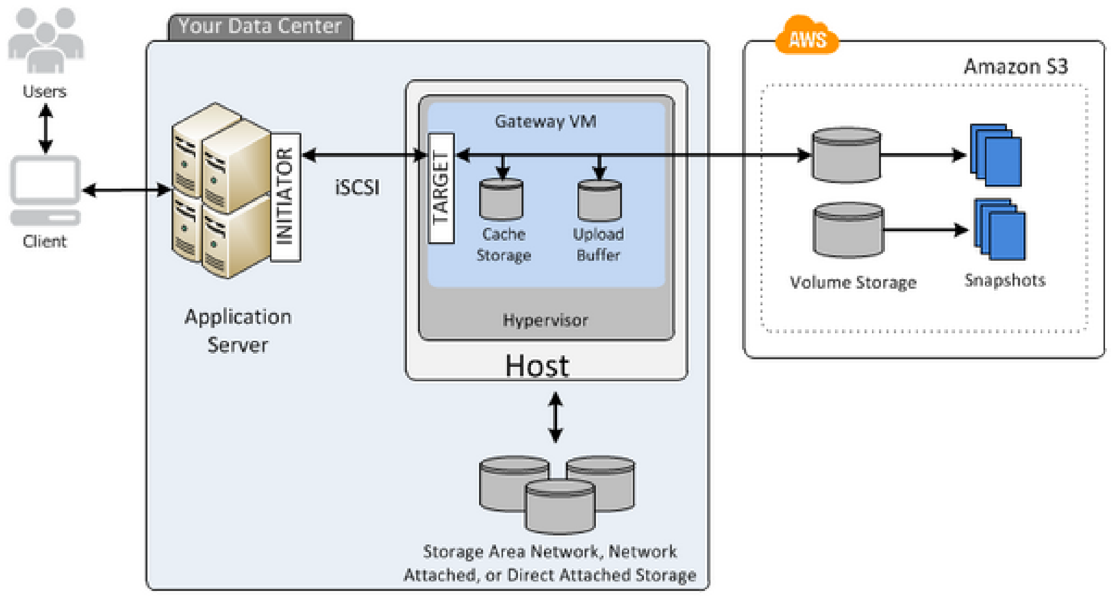
tape cartridges that you create on the Tape Gateway. It’s a great way to modernize and move backups into the cloud.

Stored Volumes vs. Cached Volumes:

- Volume Gateway's **Stored Volumes** let you store data locally on-prem and backs the data up to AWS as a secondary data source. Stored Volumes allow low-latency access to entire datasets, while providing high availability over a hybrid cloud solution. Further, you can mount Stored Volumes on application infrastructure as iSCSI drives so when data is written to these volumes, the data is both written onto the on-prem hardware and asynchronously backed up as snapshots in AWS EBS or S3.
    - In the following diagram of a Stored Volume architecture, data is served to the user from the Storage Area Network, Network Attached, or Direct Attached Storage within your data center. S3 exists just as a secure and reliable backup.



    -
- Volume Gateway's **Cached Volumes** differ as they do not store the entire dataset locally like Stored Volumes. Instead, AWS is used as the primary datasource and the local hardware is used as a caching layer. Only the most frequently used components are retained onto the on-prem infrastructure while the remaining data is served from AWS. This minimizes the need to scale on-prem infrastructure while still maintaining low-latency access to the most referenced data.
    - In the following diagram of a Cached Volume architecture, the most frequently accessed data is served to the user from the Storage Area Network, Network Attached, or Direct Attached Storage within your data center. S3 serves the rest of the data from AWS.



○

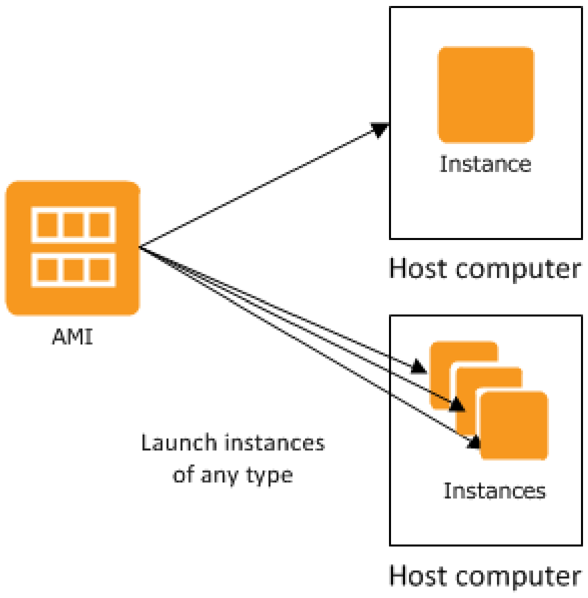
Elastic Compute Cloud (EC2)

EC2 Simplified:

EC2 spins up resizeable server instances that can scale up and down quickly. An instance is a virtual server in the cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance. Its configuration at launch is a live copy of the *Application Machine Image (AMI)* that you specify when you launched the instance. EC2 has an extremely reduced timeframe for provisioning and booting new instances and EC2 ensures that you pay as you go, pay for what you use, pay less as you use more, and pay even less when you reserve capacity. When your EC2 instance is running, you are charged on CPU, memory, storage, and networking. When it is stopped, you are only charged for EBS storage.

EC2 Key Details:

- You can launch different types of instances from a single AMI. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. You should select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on top of the instance.
- You can launch multiple instances of an AMI, as shown in the following figure:



- You have the option of using dedicated tenancy with your instance. This means that within an AWS datacenter, you have exclusive access to physical hardware. Naturally, this option incurs a high cost, but it makes sense if you work with technology that has a strict licensing policy.
- With EC2 VM Import, you can import existing VMs into AWS as long as those hosts use VMware ESX, VMware Workstation, Microsoft Hyper-V, or Citrix Xen virtualization formats.
- When you launch a new EC2 instance, EC2 attempts to place the instance in such a way that all of your VMs are spread out across different hardware to limit failure to a single location. You can use placement groups to influence the placement of a group of interdependent instances that meet the needs of your workload. There is an explanation about placement groups in a section below.
- When you launch an instance in Amazon EC2, you have the option of passing user data to the instance when the instance starts. This user data can be used to run common automated configuration tasks or scripts. For example, you can pass a bash script that ensures htop is installed on the new EC2 host and is always active.
- By default, the public IP address of an EC2 Instance is released when the instance is stopped even if its stopped temporarily. Therefore, it is best to refer to an instance by its external DNS hostname. If you require a persistent public IP address that can be associated to the same instance, use an Elastic IP address which is basically a static IP address instead.
- If you have requirements to self-manage a SQL database, EC2 can be a solid alternative to RDS. To ensure high availability, remember to have at least one other EC2 Instance in a separate Availability zone so even if a DB instance goes down, the other(s) will still be available.
- A golden image is simply an AMI that you have fully customized to your liking with all necessary software/data/configuration details set and ready to go once. This personal AMI can then be the source from which you launch new instances.
- Instance status checks check the health of the running EC2 server, systems status check monitor the health of the underlying hypervisor. If you ever notice a systems status issue, just stop the instance and start it again (no need to reboot) as the VM will start up again on a new hypervisor.

EC2 Instance Pricing:

- **On-Demand instances** are based on a fixed rate by the hour or second. As the name implies, you can start an On-Demand instance whenever you need one and can stop it when you no longer need it. There is no requirement for a long-term commitment.
- **Reserved instances** ensure that you keep exclusive use of an instance on 1 or 3 year contract terms. The long-term commitment provides significantly reduced discounts at the hourly rate.
- **Spot instances** take advantage of Amazon's excess capacity and work in an interesting manner. In order to use them, you must financially bid for access. Because Spot instances are only available when Amazon has excess capacity, this option makes sense only if your app has flexible start and end times. You won't be charged if your instance stops due to a price change (e.g., someone else just bid a higher price for the access) and so consequently your workload doesn't complete. However, if you terminate the instance yourself you will be charged for any hour the instance ran. Spot instances are normally used in batch processing jobs.

Standard Reserved vs. Convertible Reserved vs. Scheduled Reserved:

- **Standard Reserved Instances** have inflexible reservations that are discounted at 75% off of On-Demand instances. Standard Reserved Instances cannot be moved between regions. You can choose if a Reserved Instance applies to either a specific Availability Zone, or an Entire Region, but you cannot change the region.
- **Convertible Reserved Instances** are instances that are discounted at 54% off of On-Demand instances, but you can also modify the instance type at any point. For example, you suspect that after a few months your VM might need to change from general purpose to memory optimized, but you aren't sure just yet. So if you think that in the future you might need to change your VM type or upgrade your VMs capacity, choose Convertible Reserved Instances. There is no downgrading instance type with this option though.
- **Scheduled Reserved Instances** are reserved according to a specified timeline that you set. For example, you might use Scheduled Reserved Instances if you run education software that only needs to be available during school hours. This option allows you to better match your needed capacity with a recurring schedule so that you can save money.

EC2 Instance Lifecycle:

The following table highlights the many instance states that a VM can be in at a given time.

Instance state	Description	Billing
`pending`	The instance is preparing to enter the `running` state. An instance enters the pending state when it launches for the first time, or when it is started after being in the `stopped` state.	Not billed
`running`	The instance is running and ready for use.	Billed
`stopping`	The instance is preparing to be stopped or stop-hibernated.	Not billed if preparing to stop. Billed if

Instance state	Description	Billing
		preparing to hibernate
`stopped`	The instance is shut down and cannot be used. The instance can be started at any time.	Not billed
`shutting-down`	The instance is preparing to be terminated.	Not billed
`terminated`	The instance has been permanently deleted and cannot be started.	Not billed

Note: Reserved Instances that are terminated are billed until the end of their term.

EC2 Security:

- When you deploy an Amazon EC2 instance, you are responsible for management of the guest operating system (including updates and security patches), any application software or utilities installed on the instances, and the configuration of the AWS-provided firewall (called a security group) on each instance.
- With EC2, termination protection of the instance is disabled by default. This means that you do not have a safe-guard in place from accidentally terminating your instance. You must turn this feature on if you want that extra bit of protection.
- Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, and the recipient uses their private key to decrypt the data. The public and private keys are known as a key pair.
- You can encrypt your root device volume which is where you install the underlying OS. You can do this during creation time of the instance or with third-party tools like bit locker. Of course, additional or secondary EBS volumes are also encryptable as well.
- By default, an EC2 instance with an attached AWS Elastic Block Store (EBS) root volume will be deleted together when the instance is terminated. However, any additional or secondary EBS volume that is also attached to the same instance will be preserved. This is because the root EBS volume is for OS installations and other low-level settings. This rule can be modified, but it is usually easier to boot a new instance with a fresh root device volume than make use of an old one.

EC2 Placement Groups:

- Placement groups balance the tradeoff between risk tolerance and network performance when it comes to your fleet of EC2 instances. The more you care about risk, the more isolated you want your instances to be from each other. The more you care about performance, the more conjoined you want your instances to be with each other.
- There are three different types of EC2 placement groups:

  1.) Clustered Placement Groups

    - Clustered Placement Grouping is when you put all of your EC2 instances in a single availability zone. This is recommended for applications that need the lowest latency possible and require the highest network throughput.
    - Only certain instances can be launched into this group (compute optimized, GPU optimized, storage optimized, and memory optimized).

  2.) Spread Placement Groups

    - Spread Placement Grouping is when you put each individual EC2 instance on top of its own distinct hardware so that failure is isolated.
    - Your VMs live on separate racks, with separate network inputs and separate power requirements. Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.

  3.) Partitioned Placement Groups

    - Partitioned Placement Grouping is similar to Spread placement grouping, but differs because you can have multiple EC2 instances within a single partition. Failure instead is isolated to a partition (say 3 or 4 instances instead of 1), yet you enjoy the benefits of close proximity for improved network performance.
    - With this placement group, you have multiple instances living together on the same hardware inside of different availability zones across one or more regions.
    - If you would like a balance of risk tolerance and network performance, use Partitioned Placement Groups.
- Each placement group name within your AWS must be unique
- You can move an existing instance into a placement group guaranteed that it is in a stopped state. You can move the instance via the CLI or an AWS SDK, but not the console. You can also take a snapshot of the existing instance, convert it into an AMI, and launch it into the placement group where you desire it to be.

Elastic Block Store (EBS)

EBS Simplified:

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can think of EBS as a cloud-based virtual hard disk. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans.

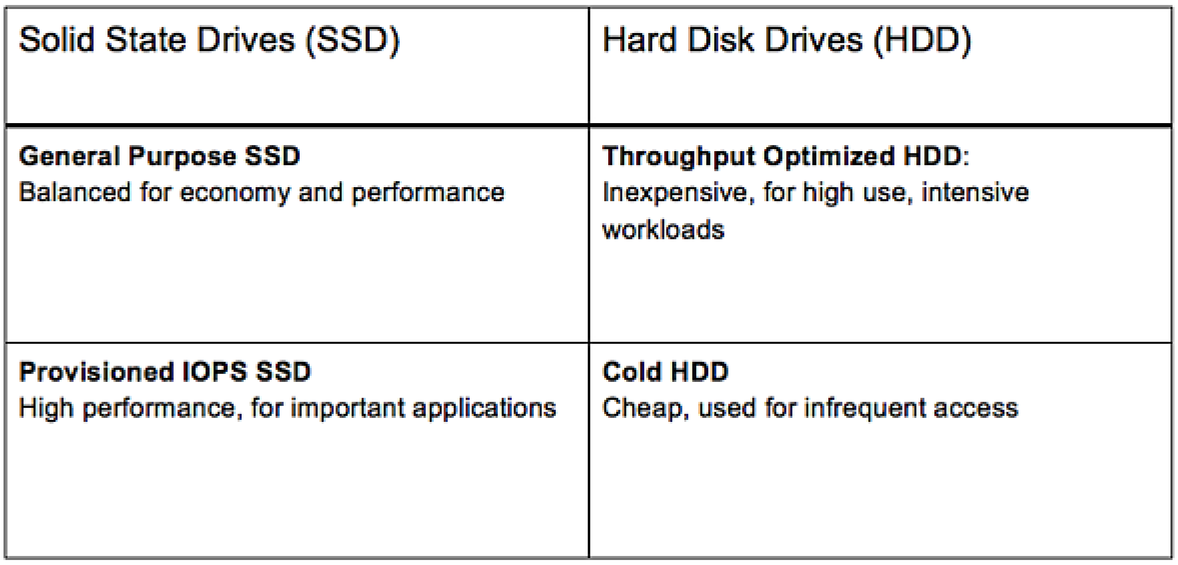
EBS Key Details:

- EBS volumes persist independently from the running life of an EC2 instance.
- Each EBS volume is automatically replicated within its Availability Zone to protect from both component failure and disaster recovery (similar to Standard S3).
- There are five different types of EBS Storage:
    - General Purpose (SSD)

    - Provisioned IOPS (SSD, built for speed)
    - Throughput Optimized Hard Disk Drive (magnetic, built for larger data loads)
    - Cold Hard Disk Drive (magnetic, built for less frequently accessed workloads)
    - Magnetic
- EBS Volumes offer 99.999% SLA.
- Wherever your EC2 instance is, your volume for it is going to be in the same availability zone
- An EBS volume can only be attached to one EC2 instance at a time.
- After you create a volume, you can attach it to any EC2 instance in the same availability zone.
- Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to S3, where it is stored redundantly in multiple Availability Zones
- An EBS snapshot reflects the contents of the volume during a concrete instant in time.
- An image (AMI) is the same thing, but includes an operating system and a boot loader so it can be used to boot an instance.
- AMIs can also be thought of as prebaked, launchable servers. AMIs are always used when launching an instance.
- When you provision an EC2 instance, an AMI is actually the first thing you are asked to specify. You can choose a pre-made AMI or choose your own made from an EBS snapshot.
- You can also use the following criteria to help pick your AMI:
    - Operating System
    - Architecture (32-bit or 64-bit)
    - Region
    - Launch permissions
    - Root Device Storage (more in the relevant section below)
- You can copy AMIs into entirely new regions.
- When copying AMIs to new regions, Amazon won't copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. You must ensure those details are properly set for the instances in the new region.
- You can change EBS volumes on the fly, including the size and storage type

SSD vs. HDD:

- SSD-backed volumes are built for transactional workloads involving frequent read/write operations, where the dominant performance attribute is IOPS. **Rule of thumb**: Will your workload be IOPS heavy? Plan for SSD.
- HDD-backed volumes are built for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS. **Rule of thumb**: Will your workload be throughput heavy? Plan for HDD.



Solid State Drives (SSD)	Hard Disk Drives (HDD)
General Purpose SSD Balanced for economy and performance	**Throughput Optimized HDD**: Inexpensive, for high use, intensive workloads
Provisioned IOPS SSD High performance, for important applications	**Cold HDD** Cheap, used for infrequent access

EBS Snapshots:

- EBS Snapshots are point in time copies of volumes. You can think of Snapshots as photographs of the disk’s current state and the state of everything within it.
- A snapshot is constrained to the region where it was created.
- Snapshots only capture the state of change from when the last snapshot was taken. This is what is recorded in each new snapshot, not the entire state of the server.
- Because of this, it may take some time for your first snapshot to be created. This is because the very first snapshot's change of state is the entire new volume. Only afterwards will the delta be captured because there will then be something previous to compare against.
- EBS snapshots occur asynchronously which means that a volume can be used as normal while a snapshot is taking place.
- When creating a snapshot for a future root device, tt is considered best practices to stop the running instance where the original device is before taking the snapshot.
- The easiest way to move an EC2 instance and a volume to another availability zone is to take a snapshot.
- When creating an image from a snapshot, if you want to deploy a different volume type for the new image (e.g. General Pupose SSD -> Throughput Optimized HDD) then you must make sure that the virtualization for the new image is hardware-assisted.
- A short summary for creating copies of EC2 instances: Old instance -> Snapshot -> Image (AMI) -> New instance
- You cannot delete a snapshot of an EBS Volume that is used as the root device of a registered AMI. If the original snapshot was deleted, then the AMI would not be able to use it as the basis to create new instances. For this reason, AWS protects you from accidentally deleting the EBS Snapshot, since it could be critical to your systems. To delete an EBS Snapshot attached to a registered AMI, first remove the AMI, then the snapshot can be deleted.

EBS Root Device Storage:

- All AMI root volumes (where the EC2's OS is installed) are of two types: EBS-backed or Instance Store-backed
- When you delete an EC2 instance that was using an Instance Store-backed root volume, your root volume will also be deleted. Any additional or secondary volumes will persist however.
- If you use an EBS-backed root volume, the root volume will not be terminated with its EC2 instance when the instance is brought offline. EBS-backed volumes are not temporary storage devices like Instance Store-backed volumes.
- EBS-backed Volumes are launched from an AWS EBS snapshot, as the name implies
- Instance Store-backed Volumes are launched from an AWS S3 stored template. They are ephemeral, so be careful when shutting down an instance!
- Secondary instance stores for an instance-store backed root device must be installed during the original provisioning of the server. You cannot add more after the fact. However, you can add EBS volumes to the same instance after the server's creation.
- With these drawbacks of Instance Store volumes, why pick one? Because they have a very high IOPS rate. So while an Instance Store can't provide data persistence, it can provide much higher IOPS compared to network attached storage like EBS.
- Further, Instance stores are ideal for temporary storage of information that changes frequently such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.
- When to use one over the other?
  - Use EBS for DB data, critical logs, and application configs.
  - Use instance storage for in-process data, noncritical logs, and transient application state.
  - Use S3 for data shared between systems like input datasets and processed results, or for static data needed by each new system when launched.

EBS Encryption:

- EBS encryption offers a straight-forward encryption solution for EBS resources that doesn't require you to build, maintain, and secure your own key management infrastructure.
- It uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and snapshots.
- You can encrypt both the root device and seconary volumes of an EC2 instance. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:
  - Data at rest inside the volume
  - All data moving between the volume and the instance
  - All snapshots created from the volume
  - All volumes created from those snapshots
- EBS encrypts your volume with a data key using the AES-256 algorithm.
- Snapshots of encrypted volumes are naturally encrypted as well. Volumes restored from encrypted snapshots are also encrypted too. You can only share unencrypted snapshots.
- The old way of encrypting a root device was to create a snapshot of a provisioned EC2 instance. While making a copy of that snapshot, you then enabled encryption during the copy's creation. Finally, once the copy was encrypted, you then created an AMI from the encrypted copy and used to have an EC2 instance with encryption on

the root device. Because of how complex this is, you can now simply encrypt root devices as part of the EC2 provisioning options.

Elastic Network Interfaces (ENI)

ENI Simplified:

An elastic network interface is a networking component that represents a virtual network card. When you provision a new instance, there will be an ENI attached automatically and you can create and configure additional network interfaces if desired. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

ENI Key Details:

- ENI is used mainly for low-budget, high-availability network solutions
- However, if you suspect you need high network throughput then you can use Enhanced Networking ENI.
- Enhanced Networking ENI uses single root I/O virtualization to provide high-performance networking capabilities on supported instance types. SR-IOV provides higher I/O and lower throughput and it ensures higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. SR-IOV does this by dedicating the interface to a single instance and effectively bypassing parts of the Hypervisor which allows for better performance.
- Adding more ENIs won’t necessarily speed up your network throughput, but Enhanced Networking ENI will.
- There is no extra charge for using Enhanced Networking ENI and the better network performance it provides. The only downside is that Enhanced Networking ENI is not available on all EC2 instance families and types.
- You can attach a network interface to an EC2 instance in the following ways:
  - When it's running (hot attach)
  - When it's stopped (warm attach)
  - When the instance is being launched (cold attach).
- If an EC2 instance fails with ENI properly configured, you (or more likely,the code running on your behalf) can attach the network interface to a hot standby instance. Because ENI interfaces maintain their own private IP addresses, Elastic IP addresses, and MAC address, network traffic will begin to flow to the standby instance as soon as you attach the network interface on the replacement instance. Users will experience a brief loss of connectivity between the time the instance fails and the time that the network interface is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.
- For instances that work with Machine Learning and High Performance Computing, use EFA (Elastic Fabric Adaptor). EFAs accelerate the work required from the above use cases. EFA provides lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based High Performance Computing systems.
- EFA can also use OS-bypass (on linux only) that will enable ML and HPC applications to interface with the Elastic Fabric Adaptor directly, rather than be normally routed to it through the OS. This gives it a huge performance increase.

- You can enable a VPC flow log on your network interface to capture information about the IP traffic going to and from a network interface.

Security Groups

Security Groups Simplified:

Security Groups are used to control access (SSH, HTTP, RDP, etc.) with EC2. They act as a virtual firewall for your instances to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instance and security groups act at the instance level, not the subnet level.

Security Groups Key Details:

- Security groups control inbound and outbound traffic for your instances (they act as a Firewall for EC2 Instances) while NACLs control inbound and outbound traffic for your subnets (they act as a Firewall for Subnets). Security Groups usually control the list of ports that are allowed to be used by your EC2 instances and the NACLs control which network or list of IP addresses can connect to your whole VPC.
- Everytime you make a change to a security group, that change occurs immediately
- Whenever you create an inbound rule, an outbound rule is created immediately. This is because Security Groups are *stateful*. This means that when you create an ingress rule for a security group, a corresponding egress rule is created to match it. This is in contrast with NACLs which are *stateless* and require manual intervention for creating both inbound and outbound rules.
- Security Group rules are based on ALLOWs and there is no concept of DENY when in comes to Security Groups. This means you cannot explicitly deny or blacklist specific ports via Security Groups, you can only implicitly deny them by excluding them in your ALLOWs list
- Because of the above detail, everything is blocked by default. You must go in and intentionally allow access for certain ports.
- Security groups are specific to a single VPC, so you can't share a Security Group between multiple VPCs. However, you can copy a Security Group to create a new Security Group with the same rules in another VPC for the same AWS Account.
- Security Groups are regional and can span AZs, but can't be cross-regional.
- Outbound rules exist if you need to connect your server to a different service such as an API endpoint or a DB backend. You need to enable the ALLOW rule for the correct port though so that traffic can leave EC2 and enter the other AWS service.
- You can attach multiple security groups to one EC2 instance and you can have multiple EC2 instances under the umbrella of one security group
- You can specify the source of your security group (basically who is allowed to bypass the virtual firewall) to be a single **/32** IP address, an IP range, or even a separate security group.
- You cannot block specific IP addresses with Security Groups (use NACLs instead)
- You can increase your Security Group limit by submitting a request to AWS

Web Application Firewall (WAF)

WAF Simplified:

AWS WAF is a web application that lets you allow or block the HTTP(s) requests that are bound for CloudFront, API Gateway, Application Load Balancers, EC2, and other Layer 7 entrypoints into your AWS environment. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns that you can define. WAF's default rule-set addresses issues like the OWASP Top 10 security risks and is regularly updated whenever new vulnerbilities are discovered.

WAF Key Details:

- As mentioned above, WAF operates as a Layer 7 firewall. This grants it the ability to monitor granular web-based conditions like URL query string parameters. This level of detail helps to detect both foul play and honest issues with the requests getting passed onto your AWS environment.
- With WAF, you can set conditions such as which IP addresses are allowed to make what kind of requests or access what kind of content.
- Based off of these conditions, the corresponding endpoint will either allow the request by serving the requested content or return an HTTP 403 Forbidden status.
- At the simplest level, AWS WAF lets you choose one of the following behaviors:
    - **Allow all requests except the ones that you specify**: This is useful when you want CloudFront or an Application Load Balancer to serve content for a public website, but you also want to block requests from attackers.
    - **Block all requests except the ones that you specify**: This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.
    - **Count the requests that match the properties that you specify**: When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn't accidentally configure AWS WAF to block all the traffic to your website. When you're confident that you specified the correct properties, you can change the behavior to allow or block requests.

WAF Protection Capabilities:

- The different request characteristics that can be used to limit access:
    - The IP address that a request originates from
    - The country that a request originates from
    - The values found in the request headers
    - Any strings that appear in the request (either specific strings or strings that match a regex pattern)
    - The length of the request
    - Any presence of SQL code (likely a SQL injection attempt)
    - Any presence of a script (likely a cross-site scripting attempt)
- You can also use NACLs to block malicious IP addresses, prevent SQL injections / XSS, and block requests from specific countries. However, it is good form to practice defense in depth.
- Denying or blocking malicious users at the WAF level has the added advantage of protecting your AWS ecosystem at its outermost border.

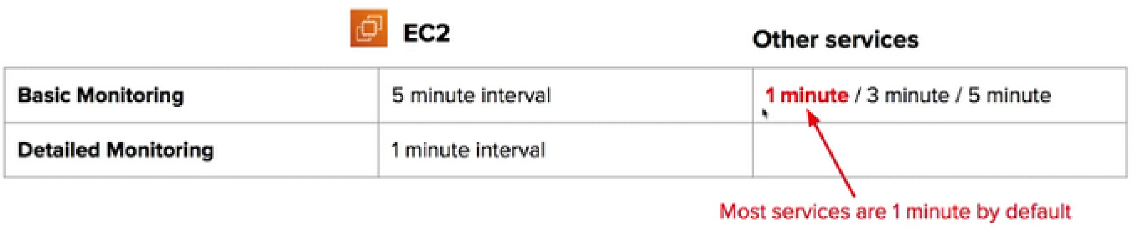
CloudWatch

CloudWatch Simplified:

Amazon CloudWatch is a monitoring and observability service. It provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

CloudWatch Key Details:

- CloudWatch collects monitoring and operational data in the form of logs, metrics, and events.
- You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.
- Within the compute domain, CloudWatch can inform you about the health of EC2 instances, Autoscaling Groups, Elastic Load Balancers, and Route53 Health Checks. Within the storage and content delivery domains, CloudWatch can inform you about the health of EBS Volumes, Storage Gateways, and CloudFront.
- With regards to EC2, CloudWathc can only monitor host level metrics such as CPU, network, disk, and status checks for insights like the health of the underlying hypervisor.
- CloudWatch is *NOT* CloudTrail so it is important to know that only CloudTrail can monitor AWS access for security and auditing reasons. CloudWatch is all about performance. CloudTrail is all about auditing.
- CloudWatch with EC2 will monitor events every 5 minutes by default, but you can have 1 minute intervals if you use Detailed Monitoring.



	EC2	Other services
Basic Monitoring	5 minute interval	1 minute / 3 minute / 5 minute
Detailed Monitoring	1 minute interval	

Most services are 1 minute by default

- You can customize your CloudWatch dashboards for insights.
- There is a multi-platform CloudWatch agent which can be installed on both Linux and Windows-based instances. This agent enables you to select the metrics to be collected, including sub-resource metrics such as per-CPU core. You can use this single agent to collect both system metrics and log files from Amazon EC2 instances and on-premises servers.
- The following metrics are not collected from EC2 instances via CloudWatch:
  - Memory utilization
  - Disk swap utilization
  - Disk space utilization
  - Page file utilization
  - Log collection

- If you need the above information, then you can retrieve it via the official CloudWatch agent or you can create a custom metric and send the data on your own via a custom script.
- CloudWatch's key purpose:
    - Collect metrics
    - Collect logs
    - Collect events
    - Create alarms
    - Create dashboards

CloudWatch Logs:

- You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, Amazon Route 53, and other sources. You can then retrieve the associated log data from CloudWatch Logs.
- It helps you centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service.
- You can create log groups so that you join logical units of CloudWatch Logs together.
- You can stream custom log files for further insights.

CloudWatch Events:

- Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources.
- You can use events to trigger lambdas for example while using alarms to inform you that something went wrong.

CloudWatch Alarms:

- CloudWatch alarms send notifications or automatically make changes to the resources you are monitoring based on rules that you define.
- For example, you can create custom CloudWatch alarms which will trigger notifications such as surpassing a set billing threshold.
- CloudWatch alarms have two states of either `ok` or `alarm`

CloudWatch Metrics:

- CloudWatch Metrics represent a time-ordered set of data points.
- These basically are a variable you can monitor over time to help tell if everything is okay, e.g. Hourly CPU Utilization.
- CloudWatch Metrics allows you to track high resolution metrics at sub-minute intervals all the way down to per second.

CloudWatch Dashboards:

- CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view
- These dashboards integrate with CloudWatch Metrics and CloudWatch Alarms to create customized views of the metrics and alarms for your AWS resources.

CloudTrail

CloudTrail Simplified:

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With it, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, API calls, and other AWS services. It is a regional service, but you can configure CloudTrail to collect trails in all regions.

CloudTrail Key Details:

- CloudTrail Events logs API calls or activities.
- CloudTrail Events stores the last 90 days of events in its Event History. This is enabled by default and is no additional cost.
- This event history simplifies security analysis, resource change tracking, and troubleshooting.
- There are two types of events that can be logged in CloudTrail: management events and data events.
- Management events provide information about management operations that are performed on resources in your AWS account.
- Think of Management events as things normally done by people when they are in AWS. Examples:
    - a user sign in
    - a policy changed
    - a newly created security configuration
    - a logging rule deletion
- Data events provide information about the resource operations performed on or in a resource.
- Think of Data events as things normally done by software when hitting various AWS endpoints. Examples:
    - S3 object-level API activity
    - Lambda function execution activity
- By default, CloudTrail logs management events, but not data events.
- By default, CloudTrail Events log files are encrypted using Amazon S3 server-side encryption (SSE). You can also choose to encrypt your log files with an AWS Key Management Service (AWS KMS) key. As these logs are stored in S3, you can define Amazon S3 lifecycle rules to archive or delete log files automatically. If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.

Elastic File System (EFS)

EFS Simplified:

EFS provides a simple and fully managed elastic NFS file system for use within AWS. EFS automatically and instantly scales your file system storage capacity up or down as you add or remove files without disrupting your application.

EFS Key Details:

- In EFS, storage capacity is elastic (grows and shrinks automatically) and its size changes based on adding or removing files.
- While EBS mounts one EBS volume to one instance, you can attach one EFS volume across multiple EC2 instances.
- The EC2 instances communicate to the remote file system using the NFSv4 protocol. This makes it required to open up the NFS port for our security group (EC2 firewall rules) to allow inbound traffic on that port.
- Within an EFS volume, the mount target state will let you know what instances are available for mounting
- With EFS, you only pay for the storage that you use so you pay as you go. No pre-provisioning required.
- EFS can scale up to the petabytes and can support thousands of concurrent NFS connections.
- Data is stored across multiple AZs in a region and EFS ensures read after write consistency.
- It is best for file storage that is accessed by a fleet of servers rather than just one server

Amazon FSx for Windows

Amazon FSx for Windows Simplified:

Amazon FSx for Windows File Server provides a fully managed native Microsoft File System.

Amazon FSx for Windows Key Details:

- With FSx for Windows, you can easily move your Windows-based applications that require file storage in AWS.
- It is built on Windows Server and exists solely for Microsoft-based applications so if you need SMB-based file storage then choose FSx.
- FSx for Windows also permits connectivity between on-premise servers and AWS so those same on-premise servers can make use of Amazon FSx too.
- You can use Microsoft Active Directory to authenticate into the file system.
- Amazon FSx for Windows provides multiple levels of security and compliance to help ensure your data is protected. Amazon FSx automatically encrypts your data at-rest and in-transit.
- You can access Amazon FSx for Windows from a variety of compute resources, not just EC2.
- You can deploy your Amazon FSx for Windows in a single AZ or in a Multi-AZ configuration.
- You can use SSD or HDD for the storage device depending on your requirements.
- FSx for Windows support daily automated backups and admins take take backups when needed as well.
- FSx for Windows removes duplicated content and compresses common content
- By default, all data is encrypted at rest.

Amazon FSx for Lustre

Amazon FSx for Lustre Simplified:

Amazon FSx for Lustre makes it easy and cost effective to launch and run the open source Lustre file system for high-performance computing applications. With FSx for Lustre, you can launch and run a file system that can proccess massive data sets at up to hundreds of gigabytes per second of throughput, millions of IOPS, and sub-millisecond latencies.

Amazon FSx for Lustre Key Details:

- FSx for Lustre is compatible with the most popular Linux-based AMIs, including Amazon Linux, Amazon Linux 2, Red Hat Enterprise Linux (RHEL), CentOS, SUSE Linux and Ubuntu.
- Since the Lustre file system is designed for high-performance computing workloads that typically run on compute clusters, choose EFS for normal Linux file system if your requirements don't match this use case.
- FSx Lustre has the ability to store and retrieve data directly on S3 on its own.

Relational Database Service (RDS)

RDS Simplified:

RDS is a managed service that makes it easy to set up, operate, and scale a relational database in AWS. It provides cost-efficient and resizable capacity while automating or outsourcing time-consuming administration tasks such as hardware provisioning, database setup, patching and backups.

RDS Key Details:

- RDS comes in six different flavors:
    - SQL Server
    - Oracle
    - MySQL Server
    - PostgreSQL
    - MariaDB
    - Aurora
- Think of RDS as the DB engine in which various DBs sit on top of.
- RDS has two key features when scaling out:
    - Read replication for improved performance
    - Multi-AZ for high availability
- In the database world, *Online Transaction Processing (OLTP)* differs from *Online Analytical Processing (OLAP)* in terms of the type of querying that you would do. OLTP serves up data for business logic that utimately composes the core functioning of your platform or application. OLAP is to gain insights into the data that you have stored in order to make better strategic decisions as a company.
- RDS runs on virtual machines, but you do not have access to those machines. You cannot SSH into an RDS instance so therefore you cannot patch the OS. This means

that AWS isresponsible for the security and maintenance of RDS. You can provision an EC2 instance as a database if you need or want to manage the underlying server yourself, but not with an RDS engine.

- Just because you cannot access the VM directly, it does not mean that RDS is serverless. There is Aurora serverless however (explained below) which serves a niche purpose.
- SQS queues can be used to store pending database writes if your application is struggling under a high write load. These writes can then be added to the database when the database is ready to process them. Adding more IOPS will also help, but this alone will not wholly eliminate the chance of writes being lost. A queue however ensures that writes to the DB do not become lost.

RDS Multi-AZ:

- Disaster recovery in AWS always looks to ensure standby copies of resources are maintained in a separate geographical area. This way, if a disaster (natural disaster, political conflict, etc.) ever struck where your original resources are, the copies would be unaffected.
- When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.
- With a Multi-AZ configuration, EC2 connects to its RDS data store using a DNS address masked as a connection string. If the primary DB fails, Multi-AZ is smart enough to detect that failure and automatically update the DNS address to point at the secondary. No manual intervention is required and AWS takes care of swapping the IP address in DNS.
- Multi-AZ is supported for all DB flavors except aurora. This is because Aurora is completely fault-tolerant on its own.
- Multi-AZ feature allows for high availability across availability zones and not regions.
- During a failover, the recovered former primary becomes the new secondary and the promoted secondary becomes primary. Once the original DB is recovered, there will be a sync process kicked off where the two DBs mirror each other once to sync up on the new data that the failed former primary might have missed out on.
- You can force a failover for a Multi-AZ setup by rebooting the primary instance
- With a Multi-AZ RDS configuration, backups are taken from the standby.

RDS Read Replicas:

- Read Replication is exclusively used for performance enhancement.
- With a Read Replica configuration, EC2 connects to the RDS backend using a DNS address and every write that is received by the master database is also passed onto a DB secondary so that it becomes a perfect copy of the master. This has the overall effect of reducing the number of transactions on the master because the secondary DBs can be queried for the same data.
- However, if the master DB were to fail, there is no automatic failover. You would have to manually create a new connection string to sync with one of the read replicas so that it becomes a master on its own. Then you’d have to update your EC2 instances

to point at the read replica. You can have up to have copies of your master DB with read replication.

- You can promote read replicas to be their very own production database if needed.
- Read replicas are supported for all six flavors of DB on top of RDS.
- Each Read Replica will have its own DNS endpoint.
- Automated backups must be enabled in order to use read replicas.
- You can have read replicas with Multi-AZ turned on or have the read replica in an entirely separate region. You can have even have read replicas of read replicas, but watch out for latency or replication lag. The caveat for Read Replicas is that they are subject to small amounts of replication lag. This is because they might be missing some of the latest transactions as they are not updated as quickly as primaries. Application designers need to consider which queries have tolerance to slightly stale data. Those queries should be executed on the read replica, while those demanding completely up-to-date data should run on the primary node.

RDS Backups:

- When it comes to RDS, there are two kinds of backups:
    - automated backups
    - database snapshots
- **Automated backups** allow you to recover your database to any point in time within a retention period (between one and 35 days). Automated backups will take a full daily snapshot and will also store transaction logs throughout the day. When you perform a DB recovery, RDS will first choose the most recent daily backup and apply the relevant transaction logs from that day. Within the set retention period, this gives you the ability to do a point in time recovery down to the precise second. Automated backups are enabled by default. The backup data is stored freely up to the size of your actual database (so for every GB saved in RDS, that same amount will freely be stored in S3 up until the GB limit of the DB). Backups are taken within a defined window so latency might go up as storage I/O is suspended in order for the data to be backed up.
- **DB snapshots** are done manually by the administrator. A key different from automated backups is that they are retained even after the original RDS instance is terminated. With automated backups, the backed up data in S3 is wiped clean along with the RDS engine. This is why you are asked if you want to take a final snapshot of your DB when you go to delete it.
- When you go to restore a DB via automated backups or DB snapshots, the result is the provisioning of an entirely new RDS instance with its own DB endpoint in order to be reached.

RDS Security:

- You can authenticate to your DB instance using IAM database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.
- An authentication token is a unique string that Amazon RDS generates on request. Authentication tokens have a lifetime of 15 minutes. You don't need to store user credentials in the database because authentication is managed externally using IAM.
- IAM database authentication provides the following benefits:

    - Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
    - You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.
    - For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security
- Encryption at rest is supported for all six flavors of DB for RDS. Encryption is done using the AWS KMS service. Once the RDS instance is encryption enabled, the data in the DB becomes encrypted as well as all backups (automated or snapshots) and read replicas.
- After your data is encrypted, Amazon RDS handles authentication of access and decryption of your data transparently with a minimal impact on performance. You don't need to modify your database client applications to use encryption.
- Amazon RDS encryption is currently available for all database engines and storage types. However, you need to ensure that the underlying instance type supports DB encryption.
- You can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created and DB instances that are encrypted can't be modified to disable encryption.

RDS Enhanced Monitoring:

- RDS comes with an Enhanced Monitoring feature. Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from CloudWatch Logs in a monitoring system of your choice.
- By default, Enhanced Monitoring metrics are stored in the CloudWatch Logs for 30 days. To modify the amount of time the metrics are stored in the CloudWatch Logs, change the retention for the RDSOSMetrics log group in the CloudWatch console.
- Take note that there are key differences between CloudWatch and Enhanced Monitoring Metrics. CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance, and Enhanced Monitoring gathers its metrics from an agent on the instance. As a result, you might find differences between the measurements, because the hypervisor layer performs a small amount of work that can be picked up and interpreted as part of the metric.

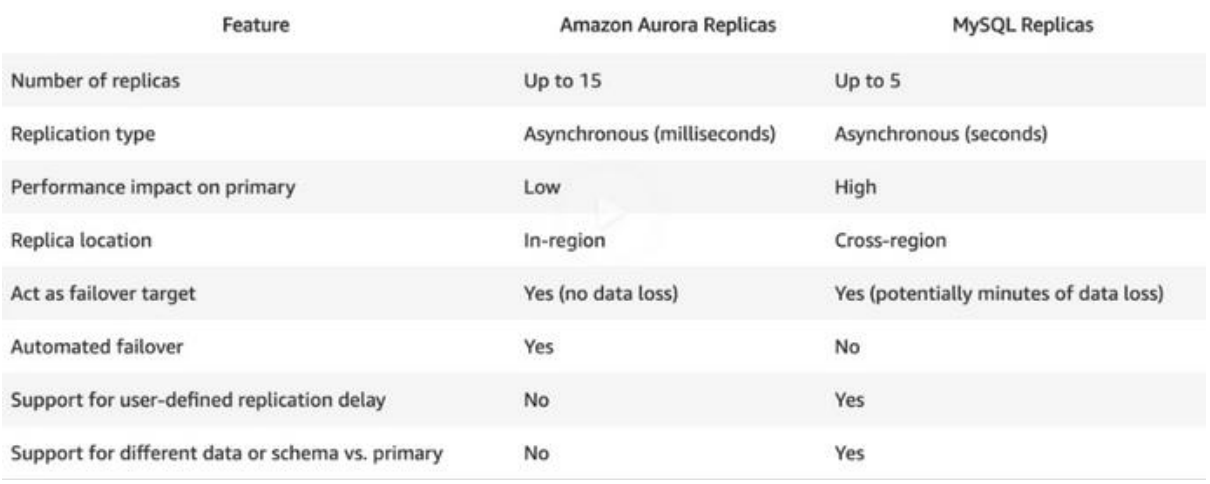
Aurora

Aurora Simplified:

Aurora is the AWS flagship DB known to combine the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. It is a MySQL/PostgreSQL-compatible RDBMS that provides the security, availability, and reliability of commercial databases at 1/10th the cost of competitors. It is far more effective as an AWS database due to the 5x and 3x performance multipliers for MySQL and PostgreSQL respectively.

Aurora Key Details:

- In case of an infrastructure failure, Aurora performs an automatic failover to to a replica of its own.
- Amazon Aurora typically involves a cluster of DB instances instead of a single instance. Each connection is handled by a specific DB instance. When you connect to an Aurora cluster, the host name and port that you specify point to an intermediate handler called an endpoint. Aurora uses the endpoint mechanism to abstract these connections. Thus, you don't have to hardcode all the hostnames or write your own logic for load-balancing and rerouting connections when some DB instances aren't available.
- By default, there are 2 copies in a minimum of 3 availability zones for 6 copies total for all of your Aurora data. This makes it possible for it to handle the potential loss of up to 2 copies of your data without impacting write availability and up to 3 copies of your data without impacting read availability.
- Aurora storage is self-healing and data blocks and disks are continuously scanned for errors. If any are found, those errors are repaired automatically.
- Aurora replication differs from RDS replicas in the sense that it is possible for Aurora's replicas to be be both a standby as part of a multi-AZ configuration as well as a target for read traffic. In RDS, the multi-AZ standby cannot be configured to be a read endpoint and only read replicas can serve that function.
- With Aurora replication, you can have up to fifteen copies. If you want downstream MySQL or PostgreSQL as you replicated copies, then you can only have 5 or 1.
- Automated failover is only possible with Aurora read replication
- For more on the differences between RDS replication and Aurora Replication, please consult the following:



Feature	Amazon Aurora Replicas	MySQL Replicas
Number of replicas	Up to 15	Up to 5
Replication type	Asynchronous (milliseconds)	Asynchronous (seconds)
Performance impact on primary	Low	High
Replica location	In-region	Cross-region
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs. primary	No	Yes

- Automated backups are always enabled on Aurora instances and backups don’t impact DB performance. You can also take snapshots which also don’t impact performance. Your snapshots can be shared across AWS accounts.
- A common tactic for migrating RDS DBs into Aurora RDs is to create a read replica of a RDS MariaDB/MySQL DB as an Aurora DB. Then simply promote the Aurora DB into a production instance and delete the old MariaDB/MySQL DB.
- Aurora starts w/ 10GB and scales per 10GB all the way to 64 TB via storage autoscaling. Aurora's computing power scales up to 32vCPUs and 244GB memory

Aurora Serverless:

- Aurora Serverless is a simple, on-demand, autoscaling configuration for the MySQL/PostgreSQl-compatible editions of Aurora. With Aurora Serveress, your instance automatically scales up or down and starts on or off based on your application usage. The use cases for this service are infrequent, intermittent, and unpredictable workloads.
- This also makes it possible cheaper because you only pay per invocation
- With Aurora Serverless, you simply create a database endpoint, optionally specify the desired database capacity range, and connect your applications.
- It removes the complexity of managing database instances and capacity. The database will automatically start up, shut down, and scale to match your application's needs. It will seamlessly scale compute and memory capacity as needed, with no disruption to client connections.

Aurora Cluster Endpoints:

- Using cluster endpoints, you map each connection to the appropriate instance or group of instances based on your use case.
- You can connect to cluster endpoints associated with different roles or jobs across your Aurora DB. This is because different instances or groups of instances perform different functions.
- For example, to perform DDL statements you can connect to the primary instance. To perform queries, you can connect to the reader endpoint, with Aurora automatically performing load-balancing among all the Aurora Replicas behind the reader endpoint. For diagnosis or tuning, you can connect to a different endpoint to examine details.
- Since the entryway for your DB Instance remains the same after a failover, your application can resume database operation without the need for manual administrative intervention for any of your endpoints.

Aurora Reader Endpoints:

- Aurora Reader endpoints are a subset of the above idea of cluster endpoints. Use the reader endpoint for read operations, such as queries. By processing those statements on the read-only Aurora Replicas, this endpoint reduces the overhead on the primary instance.
- There are up 15 Aurora Read Replicas because a Reader Endpoint to help handle read-only query traffic.
- It also helps the cluster to scale the capacity to handle simultaneous SELECT queries, proportional to the number of Aurora Replicas in the cluster. Each Aurora DB cluster has one reader endpoint.
- If the cluster contains one or more Aurora Replicas, the reader endpoint load-balances each connection request among the Aurora Replicas. In that case, you can only perform read-only statements such as SELECT in that session. If the cluster only contains a primary instance and no Aurora Replicas, the reader endpoint connects to the primary instance directly. In that case, you can perform write operations through the endpoint.

DynamoDB

DynamoDB Simplified:

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster, durable non-SQL database. It comes with built-in security, backup and restore, and in-memory caching for internet-scale applications.

DynamoDB Key Details:

- The main components of DyanmoDB are:
  - a collection which serves as the foundational table
  - a document which is equivalent to a row in a SQL database
  - key-value pairs which are the fields within the document or row
- The convenience of non-relational DBs is that each row can look entirely different based on your use case. There doesn't need to be uniformity. For example, if you need a new column for a particular entry you don't also need to ensure that that column exists for the other entries.
- DyanmoDB supports both document and key-value based models. It is a great fit for mobile, web, gaming, ad-tech, IoT, etc.
- DynamoDB is stored via SSD which is why it is so fast.
- It is spread across 3 geographically distinct data centers.
- The default consistency model is Eventually Consistent Reads, but there are also Strongly Consistent Reads.
- The difference between the two consistency models is the one second rule. With Eventual Consistent Reads, all copies of data are usually reached within one second. A repeated read after a short period of time should return the updated data. However, if you need to read updated data within or less than a second and this needs to be a guarantee, then strongly consistent reads are your best bet.
- If you face a scenario that requires the schema, or the structure of your data, to change frequently, then you have to pick a database which provides a non-rigid and flexible way of adding or removing new types of data. This is a classic example of choosing between a relational database and non-relational (NoSQL) database. In this scenario, pick DynamoDB.
- A relational database system does not scale well for the following reasons:
  - It normalizes data and stores it on multiple tables that require multiple queries to write to disk.
  - It generally incurs the performance costs of an ACID-compliant transaction system.
  - It uses expensive joins to reassemble required views of query results.
- High cardinality is good for DynamoDB I/O performance. The more distinct your partition key values are, the better. It makes it so that the requests sent will be spread across the partitioned space.
- DynamoDB makes use of parallel processing to achieve predictable performance. You can visualise each partition or node as an independent DB server of fixed size with each partition or node responsible for a defined block of data. In SQL terminology, this concept is known as sharding but of course DynamoDB is not a SQL-based DB. With DynamoDB, data is stored on Solid State Drives (SSD).

DynamoDB Accelerator (DAX):

- Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache that can reduce Amazon DynamoDB response times from milliseconds to microseconds, even at millions of requests per second.
- With DAX, your applications remain fast and responsive, even when unprecedented request volumes come your way. There is no tuning required.
- DAX lets you scale on-demand out to a ten-node cluster, giving you millions of requests per second.
- Just like DynamoDB, DAX is fully managed. You no longer need to worry about management tasks such as hardware or software provisioning, setup and configuration, software patching, operating a reliable, distributed cache cluster, or replicating data over multiple instances as you scale.
- DAX enables you to provision one DAX cluster for multiple DynamoDB tables, multiple DAX clusters for a single DynamoDB table or somewhere in between giving you maximal flexibility.

DynamoDB Streams:

- A DynamoDB stream is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.
- Amazon DynamoDB is integrated with AWS Lambda so that you can create triggers—pieces of code that automatically respond to events in DynamoDB Streams.
- Immediately after an item in the table is modified, a new record appears in the table's stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records. The Lambda function can perform any actions you specify, such as sending a notification or initiating a workflow.
- With triggers, you can build applications that react to data modifications in DynamoDB tables.
- Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attribute(s) of the items that were modified. A stream record contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the "before" and "after" images of modified items.

Redshift

Redshift Simplified:

Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. The Amazon Redshift service manages all of the work of setting up, operating, and scaling a data warehouse. These tasks include provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine.

Redshift Key Details:

- An Amazon Redshift cluster is a set of nodes which consists of a leader node and one or more compute nodes. The type and number of compute nodes that you need depends on the size of your data, the number of queries you will execute, and the query execution performance that you need.

- Redshift is used for business intelligence and pulls in very large and complex datasets to perform complex queries in order to gather insights from the data.
- It fits the use case of Online Analytical Processing (OLAP). Redshift is a powerful technology for data discovery including capabilities for almost limitless report viewing, complex analytical calculations, and predictive "what if" scenario (budget, forecast, etc.) planning.
- Depending on your data warehousing needs, you can start with a small, single-node cluster and easily scale up to a larger, multi-node cluster as your requirements change. You can add or remove compute nodes to the cluster without any interruption to the service.
- If you intend to keep your cluster running for a year or longer, you can save money by reserving compute nodes for a one-year or three-year period.
- Snapshots are point-in-time backups of a cluster. These backups are enabled by default with a 1 day retention period. The maximum retention period is 35 days.
- Redshift can also asynchronously replicate your snapshots to a different region if desired.
- A Highly Available Redshift cluster would require 3 copies of your data. One copy would be live in Redshift and the others would be standby in S3.
- Redshift can have up to 128 compute nodes in a multi-node cluster. The leader node always manages client connections and relays queries to the compute nodes which store the actual data and perform the queries.
- Redshift is able to achieve efficiency despite the many parts and pieces in its architecture through using columnar compression of data stores that contain similar data. In addition, Redshift does not require indexes or materialized views which means it can be relatively smaller in size compared to an OLTP database containing the same amount of information. Finally, when loading data into a Redshift table, Redshift will automatically down sample the data and pick the most appropriate compression scheme.
- Redshift also comes with Massive Parallel Processing (MPP) in order to take advantage of all the nodes in your multi-node cluster. This is done by evenly distributing data and query load across all nodes. Because of this, scaling out still retains great performance.
- Redshift is encrypted in transit using SSL and is encrypted at rest using AES-256. By default, Redshift will manage all keys, but you can do so too via AWS CloudHSM or AWS KMS.
- Redshift is billed for:
    - Compute Node Hours (total hours your non-leader nodes spent querying for data)
    - Backups
    - Data transfer within a VPC (but not outside of it)
- Redshift is not multi-AZ, if you want multi-AZ you will need to spin up a separate cluster ingesting the same input. You can also manually restore snapshots to a new AZ in the event of an outage.
- When you provision an Amazon Redshift cluster, it is locked down by default so nobody has access to it. To grant other users inbound access to an Amazon Redshift cluster, you associate the cluster with a security group.
- Amazon Redshift provides free storage for snapshots that is equal to the storage capacity of your cluster until you delete the cluster. After you reach the free snapshot

storage limit, you are charged for any additional storage at the normal rate. Because of this, you should evaluate how many days you need to keep automated snapshots and configure their retention period accordingly, and delete any manual snapshots that you no longer need.

- Regardless of whether you enable automated snapshots, you can take a manual snapshot whenever you want. Amazon Redshift will never automatically delete a manual snapshot. Manual snapshots are retained even after you delete your Redshift cluster. Because manual snapshots accrue storage charges, it’s important that you manually delete them if you no longer need them

Redshift Spectrum:

- Amazon Redshift Spectrum is used to run queries against exabytes of unstructured data in Amazon S3, with no loading or ETL required.
- Redshift Spectrum queries employ massive parallelism to execute very fast against large datasets. Much of the processing occurs in the Redshift Spectrum layer, and most of the data remains in Amazon S3.
- Redshift Spectrum queries use much less of your cluster's processing capacity than other queries.
- The cluster and the data files in Amazon S3 must be in the same AWS Region.
- External S3 tables are read-only. You can't perform insert, update, or delete operations on external tables.

Redshift Enhanced VPC Routing:

- When you use Amazon Redshift Enhanced VPC Routing, Redshift forces all traffic (such as COPY and UNLOAD traffic) between your cluster and your data repositories through your Amazon VPC.
- If Enhanced VPC Routing is not enabled, Amazon Redshift routes traffic through the Internet, including traffic to other services within the AWS network.
- By using Enhanced VPC Routing, you can use standard VPC features, such as VPC security groups, network access control lists (ACLs), VPC endpoints, VPC endpoint policies, internet gateways, and Domain Name System (DNS) servers.

ElastiCache

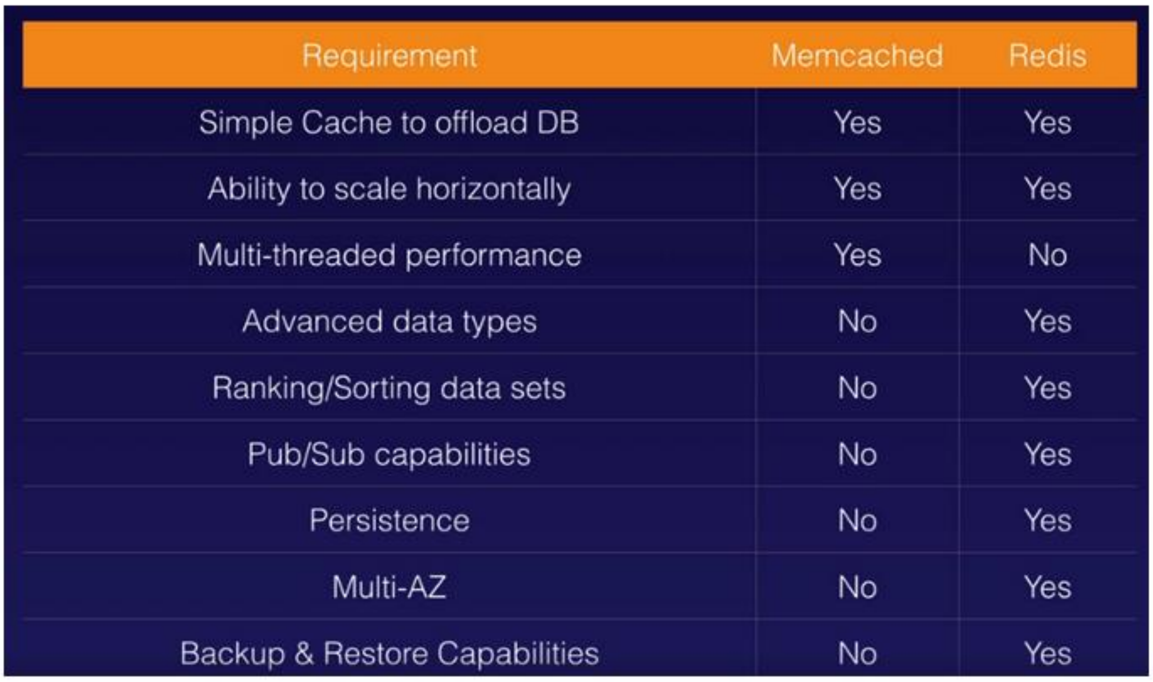
ElastiCache Simplified:

The ElastiCache service makes it easy to deploy, operate, and scale an in-memory cache in the cloud. It helps you boost the performance of your existing databases by retrieving data from high throughput and low latency in-memory data stores.

ElastiCache Key Details:

- The service is great for improving the performance of web applications by allowing you to receive information locally instead of relying solely on relatively distant DBs.
- Amazon ElastiCache offers fully managed Redis and Memcached for the most demanding applications that require sub-millisecond response times.

- For data that doesn’t change frequently and is oftenly asked for, it makes a lot of sense to cache said data rather than querying it from the database.
- Common configurations that improve DB performance include introducing read replicas of a DB primary and inserting a caching layer into the storage architecture.
- MemcacheD is for simple caching purposes with horizontal scaling and multi-threaded performance, but if you require more complexity for your caching environment then choose Redis.
- A further comparison between MemcacheD and Redis for ElastiCache:



Requirement	Memcached	Redis
Simple Cache to offload DB	Yes	Yes
Ability to scale horizontally	Yes	Yes
Multi-threaded performance	Yes	No
Advanced data types	No	Yes
Ranking/Sorting data sets	No	Yes
Pub/Sub capabilities	No	Yes
Persistence	No	Yes
Multi-AZ	No	Yes
Backup & Restore Capabilities	No	Yes

- Another advatnage of using ElastiCache is that by caching query results, you pay the price of the DB query only once without having to re-execute the query unless the data changes.
- Amazon ElastiCache can scale-out, scale-in, and scale-up to meet fluctuating application demands. Write and memory scaling is supported with sharding. Replicas provide read scaling.

Route53

Route53 Simplified:

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.

Route53 Key Details:

- DNS is used to map human-readable domain names into an internet protocol address similarly to how phonebooks map company names with phone numbers.
- AWS has its own domain registrar.

- When you buy a domain name, every DNS address starts with an SOA (Start of Authority) record. The SOA record stores information about the name of the server that kicked off the transfer of ownership, the administrator who will now use the domain, the current metadata available, and the default number of seconds or TTL.
- NS records, or Name Server records, are used by the Top Level Domain hosts (.org, .com, .uk, etc.) to direct traffic to the Content servers. The Content DNS servers contain the authoritative DNS records.
- Browsers talk to the Top Level Domains whenever they are queried and encounter domain name that they do not recognize.
    1. Browsers will ask for the authoritative DNS records associated with the domain.
    2. Because the Top Level Domain contains NS records, the TLD can in turn queries the Name Servers for their own SOA.
    3. Within the SOA, there will be the requested information.
    4. Once this information is collected, it will then be returned all the way back to the original browser asking for it.
- In summary: Browser -> TLD -> NS -> SOA -> DNS record. The pipeline reverses when the correct DNS record is found.
- Authoritative name servers store DNS record information, usually a DNS hosting provider or domain registrar like GoDaddy that offers both DNS registration and hosting.
- There are a multitude of DNS records for Route53. Here are some of the more common ones:
    - **A records**: These are the fundamental type of DNS record. The “A” in A records stands for “address”. These records are used by a computer to directly pair a domain name to an IP address. IPv4 and IPv6 are both supported with "AAAA" refering to the IPv6 version. **A: URL -> IPv4** and **AAAA: URL -> IPv6**.
    - **CName records**: Also refered to as the Canonical Name. These records are used to resolve one domain name to another domain name. For example, the domain of the mobile version of a website may be a CName from the domain of the browser version of that same website rather than a separate IP address. This would allow mobile users who visit the site and to receive the mobile version. **CNAME: URL -> URL**.
    - **Alias records**: These records are used to map your domains to AWS resources such as load balancers, CDN endpoints, and S3 buckets. Alias records function similarly to CNames in the sense that you map one domain to another. The key difference though is that by pointing your Alias record at a service rather than a domain name, you have the ability to freely change your domain names if needed and not have to worry about what records might be mapped to it. Alias records give you dynamic functionality. **Alias: URL -> AWS Resource**.
    - **PTR records**: These records are the opposite of an A record. PTR records map an IP to a domain and they are used in reverse DNS lookups as a way to obtain the domain name of an IP address. **PTR: IPv4 -> URL**.
- One other major difference between CNames and Alias records is that a CName cannot be used for the naked domain name (the apex record in your entire DNS configuration / the primary record to be used). CNames must always be secondary records that can map to another secondary record or the apex record. The primary must always be of type Alias or A Record in order to work.

- Due to the dynamic nature of Alias records, they are often reccomended for most usecases and should be used when it is possible to.
- TTL is the length that a DNS record is cached on either the resolving servers or the users own cache so that a fresher mapping of IP to domain can be retrieved. Time To Live is measured in seconds and the lower the TTL the faster DNS changes propagate across the internet. Most providers, for example, have a TTL that lasts 48 hours.
- You can create health checks to send you a Simple Notification if any issues arise with your DNS setup.
- Further, Route53 health checks can be used for any AWS endpoint that can be accessed via the Internet. This makes it an ideal option for monitoring the health of your AWS endpoints.

Route53 Routing Policies:

- When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to DNS queries. The routing policies available are:
  - Simple Routing
  - Weighted Routing
  - Latency-based Routing
  - Failover Routing
  - Geolocation Routing
  - Geo-proximity Routing
  - Multivalue Answer Routing
- **Simple Routing** is used when you just need a single record in your DNS with either one or more IP addresses behind the record in case you want to balance load. If you specify multiple values in a Simple Routing policy, Route53 returns a random IP from the options available.
- **Weighted Routing** is used when you want to split your traffic based on assigned weights. For example, if you want 80% of your traffic to go to one AZ and the rest to go to another, use Weighted Routing. This policy is very useful for testing feature changes and due to the traffic splitting characteristics, it can double as a means to perform blue-green deployments. When creating Weighted Routing, you need to specify a new record for each IP address. You cannot group the various IPs under one record like with Simple Routing.
- **Latency-based Routing**, as the name implies, is based on setting up routing based on what would be the lowest latency for a given user. To use latency-based routing, you must create a latency resource record set in the same region as the corresponding EC2 or ELB resource receiving the traffic. When Route53 receives a query for your site, it selects the record set that gives the user the quickest speed. When creating Latency-based Routing, you need to specify a new record for each IP.
- **Failover Routing** is used when you want to configure an active-passive failover set up. Route53 will monitor the health of your primary so that it can failover when needed. You can also manually set up health checks to monitor all endpoints if you want more detailed rules.
- **Geolocation Routing** lets you choose where traffic will be sent based on the geographic location of your users.
- **Geo-proximity Routing** lets you choose where traffic will be sent based on the geographic location of your users *and* your resources. You can choose to route more or less traffic based on a specified weight which is referred to as a bias. This bias either expands or shrinks the availability of a geographic region which makes it easy

to shift traffic from resources in one location to resources in another. To use this routing method, you must enable Route53 traffic flow. If you want to control global traffic, use Geo-proximity routing. If you want traffic to stay in a local region, use Geolocation routing.

- **Multivalue Routing** is pretty much the same as Simple Routing, but Multivalue Routing allows you to put health checks on each record set. This ensures then that only a healthy IP will be randomly returned rather than any IP.

Elastic Load Balancers (ELB)

ELB Simplified:

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, Docker containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

ELB Key Details:

- Load balancers can be internet facing or application internal.
- To route domain traffic to an ELB load balancer, use Amazon Route 53 to create an Alias record that points to your load balancer. An Alias record is preferable over a CName, but both can work.
- ELBs do not have predefined IPv4 addresses; you must resolve them with DNS instead. Your load balancer will never have its own IP by default, but you can create a static IP for a network load balancer because network LBs are for high performance purposes.
- Instances behind the ELB are reported as `InService` or `OutofService`. When an EC2 instance behind an ELB fails a health check, the ELB stops sending traffic to that instance.
- A dual stack configuration for a load balancer means load balancing over IPv4 and IPv6
- In AWS, there are three types of LBs:
    - Application LBs
    - Network LBs
    - Classic LBs.
- **Application LBs** are best suited for HTTP(S) traffic and they balance load on layer 7. They are intelligent enough to be application aware and Application Load Balancers also support path-based routing, host-based routing and support for containerized applications. As an example, if you change your web browser’s language into French, an Application LB has visibility of the metadata it receives from your brower which contains details about the language you use. To optimize your browsing experience, it will then route you to the French-language servers on the backend behind the LB. You can also create advanced request routing, moving traffic into specific servers based on rules that you set yourself for specific cases.

- **Network LBs** are best suited for TCP traffic where performance is required and they balance load on layer 4. They are capable of managing millions of requests per second while maintaining extremely low latency.
- **Classic LBs** are the legacy ELB produce and they balance either on HTTP(S) or TCP, but not both. Even though they are the oldest LBs, they still support features like sticky sessions and X-Forwarded-For headers.
- If you need flexible application management and TLS termination then you should use the Application Load Balancer. If extreme performance and a static IP is needed for your application then you should use the Network Load Balancer. If your application is built within the EC2 Classic network then you should use the Classic Load Balancer.
- The lifecycle of a request to view a website behind an ELB:
    1. The browser requests the IP address for the load balancer from DNS.
    2. DNS provides the IP.
    3. With the IP at hand, your browser then makes an HTTP request for an HTML page from the Load Balancer.
    4. AWS perimiter devices checks and verifies your request before passing it onto the LB.
    5. The LB finds an active webserver to pass on the HTTP request.
    6. The webserver returns the requested HTML file.
    7. The browser receives the HTML file it requested and renders the graphical representation of it on the screen.
- Load balancers are a regional service. They do not balance load across different regions. You must provision a new ELB in each region that you operate out of.
- If your application stops responding, you’ll receive a 504 error when hitting your load balancer. This means the application is having issues and the error could have bubbled up to the load balancer from the services behind it. It does not necessarily mean there's a probem with the LB itself.

ELB Advanced Features:

- To enable IPv6 DNS resolution, you need to create a second DNS resource record so that the **ALIAS AAAA** record resolves to the load balancer along with the IPv4 record.
- The X-Forwarded-For header, via the Proxy Protocol, is simply the idea for load balancers to forward the requester's IP address along with the actual request for information from the servers behind the LBs. Normally, the servers behind the LBs only see that the IP sending it traffic belongs to the Load Balancer. They usually have no idea about the true origin of the request as they only know about the computer (the LB) that asks them to do something. But sometimes we may want to route the original IP to the backend servers for specific use cases and have the LB’s IP address ignored. The X-Forwarded-For header makes this possible.
- Sticky Sessions bind a given user to a specific instance throughout the duration of their stay on the application or website. This means all of their interactions with the application will be directed to the same host each time. If you need local disk for your application to work, sticky sessions are great as users are guaranteed consistent access to the same ephemeral storage on a particular instance. The downside of sticky sessions is that, if done improperly, it can defeat the purpose of load balancing. All traffic could hypothetically be bound to the same instance instead of being evenly distributed.

- Path Patterns create a listener with rules to forward requests based on the URL path set within those user requests. This method, known as path-based routing, ensures that traffic can be specifically directed to multiple back-end services. For example, with Path Patterns you can route general requests to one target group and requests to render images to another target group. So the URL, "www.example.com/" will be forwarded to a server that is used for general content while "www.example.com/photos" will be forwarded to another server that renders images.

ELB Cross Zone Load Balancing:

- Cross Zone load balancing guarantees even distribution across AZs rather than just within a single AZ.
- If Cross Zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only.
- Cross Zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances.
- However, it is still recommend that you maintain approximately equivalent numbers of instances in each enabled Availability Zone for higher fault tolerance.
- For environments where clients cache DNS lookups, incoming requests might favor one of the Availability Zones. Using Cross Zone load balancing, this imbalance in the request load is spread across all available instances in the region instead.

ELB Security:

- ELB supports SSL/TLS & HTTPS termination. Termination at load balancer is desired because decryption is resource and CPU intensive. Putting the decryption burden on the load balancer enables the EC2 instances to spend their processing power on application tasks, which helps improve overall performance.
- Elastic Load Balancers (along with CloudFront) support Perfect Forward Secrecy. This is a feature that provides additional safeguards against the eavesdropping of encrypted data in transit through the use of a uniquely random session key. This is done by ensuring that the in-use part of an encryption system automatically and frequently changes the keys it uses to encrypt and decrypt information. So if this latest key is compromised at all, it will only expose a small portion of the user's recent data.
- Classic Load Balancers do not support Server Name Indication (SNI). SNI allows the server (the LB in this case) to safely host multiple TLS Certificates for multiple sites all under a single IP address (the Alias record or CName record in this case). To allow SNI, you have to use an Application Load Balancer instead or use it with a CloudFront web distribution.

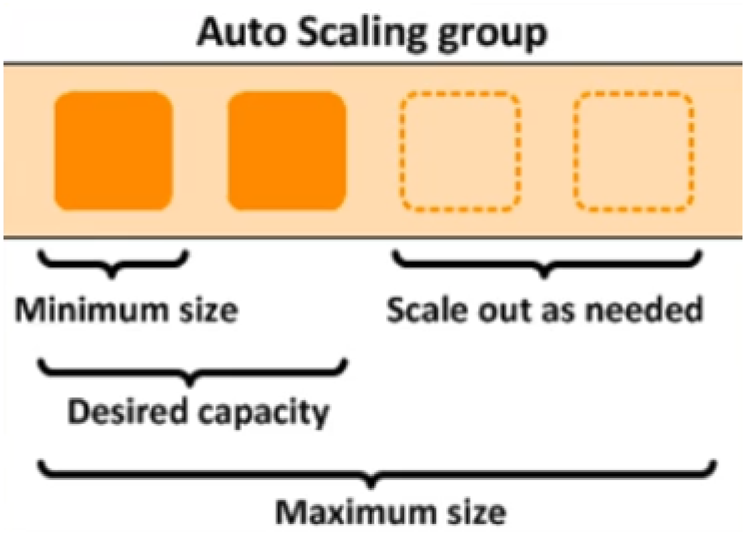
Auto Scaling

Auto Scaling Simplified:

AWS Auto Scaling lets you build scaling plans that automate how groups of different resources respond to changes in demand. You can optimize availability, costs, or a balance of both. AWS Auto Scaling automatically creates all of the scaling policies and sets targets for you based on your preference.

Auto Scaling Key Details:

- Auto Scaling is a major benefit from the cloud's economies of scale so if you ever have a requirement for scaling, automatically think of using the Auto Scaling service.
- Auto Scaling has three components:
    - **Groups**: These are logical components. A webserver group of EC2 instances, a database group of RDS instances, etc.
    - **Configuration Templates**: Groups use a template to configure and launch new instances to better match the scaling needs. You can specify information for the new instances like the AMI to use, the instance type, security groups, block devices to associate with the instances, and more.
    - **Scaling Options**: Scaling Options provides several ways for you to scale your Auto Scaling groups. You can base the scaling trigger on the occurence of a specified condition or on a schedule.
- The following image highlights the state of an Auto scaling group. The orange squares represent active instances. The dotted squares represent potential instances that can and will be spun up whenever necessary. The minimum number, the maximum number, and the desired capacity of instances are all entirely configurable.



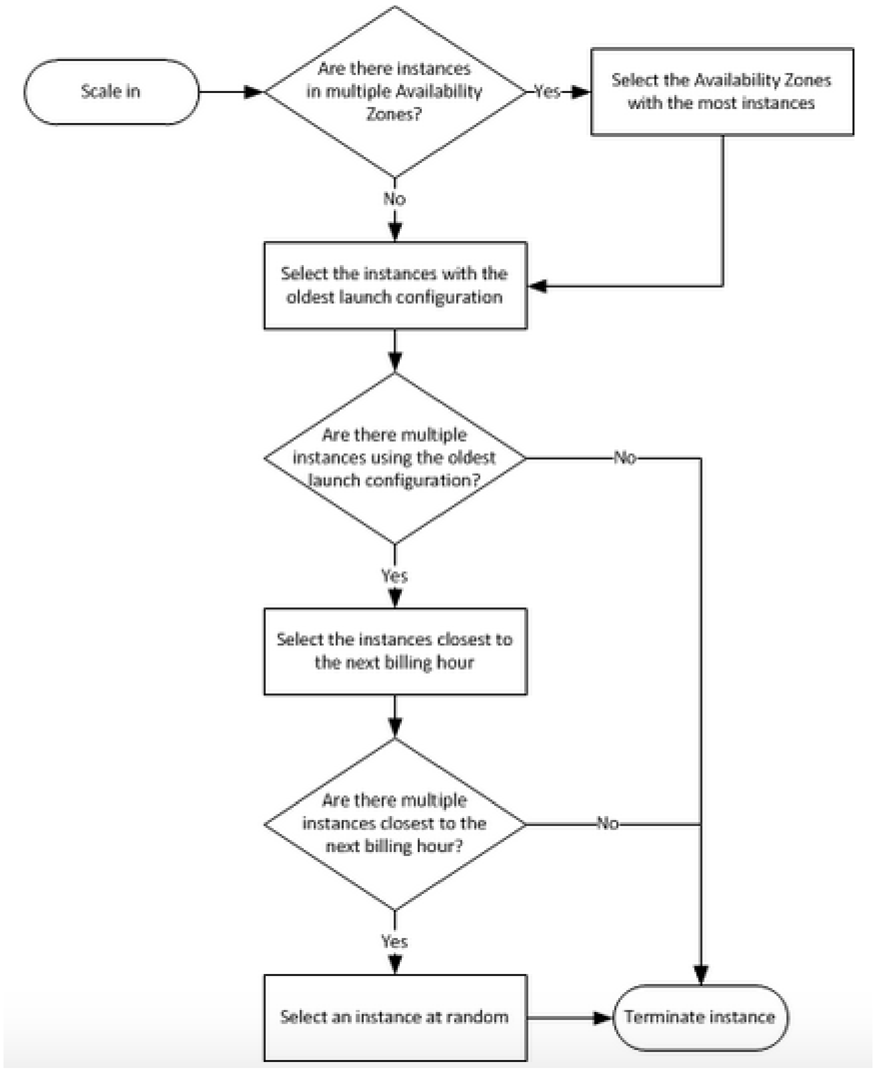
- When you use Auto Scaling, your applications gain the following benefits:
    - **Better fault tolerance**: Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.
    - **Better availability**: Auto Scaling can help you ensure that your application always has the right amount of capacity to handle the current traffic demands.
- When it comes to actualing scaling your instance groups, the Auto Scaling service is flexible and can be done in various ways:
    - Auto Scaling can scale based on the demand placed on your instances. This option automates the scaling process by specifying certain thresholds that, when reached, will trigger the scaling. This is the most popular implementation of Auto Scaling.

    - Auto Scaling can ensure the current number of instances at all times. This option will always maintain the number of servers you want running even when they fail.
    - Auto Scaling can scale only with manual intervention. If want to control all of the scaling yourself, this option makes sense.
    - Auto Scaling can scale based on a schedule. If you can reliably predict spikes in traffic, this option makes sense.
    - Auto Scaling based off of predictive scaling. This option lets AWS AI/ML learn more about your environment in order to predict the best time to scale for both performance improvements and cost-savings.
- In maintaining the current running instance, Auto Scaling will perform occasional health checks on the running instances to ensure that they are all healthy. When the service detects that an instance is unhealthy, it will terminate that instance and then bring up a new one online.
- When designing HA for your Auto Scaling, use multiple AZs and multiple regions wherever you can.
- Auto Scaling allows you to suspend and then resume one or more of the Auto Scaling processes in your Auto Scaling Group. This can be very useful when you want to investigate a problem in your application without triggering the Auto Scaling process when making changes.
- You can specify your launch configuration with multiple Auto Scaling groups. However, you can only specify one launch configuration for an Auto Scaling group at a time.
- You cannot modify a launch configuration after you've created it. If you want to change the launch configuration for an Auto Scaling group, you must create a new launch configuration and update your Auto Scaling group to inherit this new launch configuration.

Auto Scaling Default Termination Policy:

- The default termination policy for an Auto Scaling Group is to automatically terminate a stopped instance, so unless you've configured it to do otherwise, stopping an instance will result in termination regardless if you wanted that to happen or not. A new instance will be spun up in its place.
- The default termination policy will spare instances that you tell it in case some servers are running critical systems or applications. These critical servers are protected from "scale in", which is just the deletion process of instances deemed superfluous to requirements.
- The default termination policy is designed to help ensure that your network architecture spans Availability Zones evenly. With the default termination policy, the behavior of the Auto Scaling group is as follows:
    - If there are instances in multiple Availability Zones, it will terminate an instance from the Availability Zone with the most instances. If there is more than one Availability Zone with the same max number of instances, it will choose the Availability Zone where instances use the oldest launch configuration.
    - It will then determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, it will terminate it.

    - If there are multiple instances to terminate, it will determine which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances and manage your Amazon EC2 usage costs.) If there are some instances that match this criteria, they will be terminated.
- This flow chart can provide futher clarity on how the default Auto Scaling policy decides which instances to delete:


Scale in
Are there instances in multiple Availability Zones?
Yes
Select the Availability Zones with the most instances
No
Select the instances with the oldest launch configuration
Are there multiple instances using the oldest launch configuration?
No
Yes
Select the instances closest to the next billing hour
Are there multiple instances closest to the next billing hour?
No
Yes
Select an instance at random
Terminate instance

Auto Scaling Cooldown Period:

- The cooldown period is a configurable setting for your Auto Scaling Group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect.
- After the Auto Scaling Group scales using a policy, it waits for the cooldown period to complete before resuming further scaling activities if needed.
- The default waiting period is 300 seconds, but this can be modified.

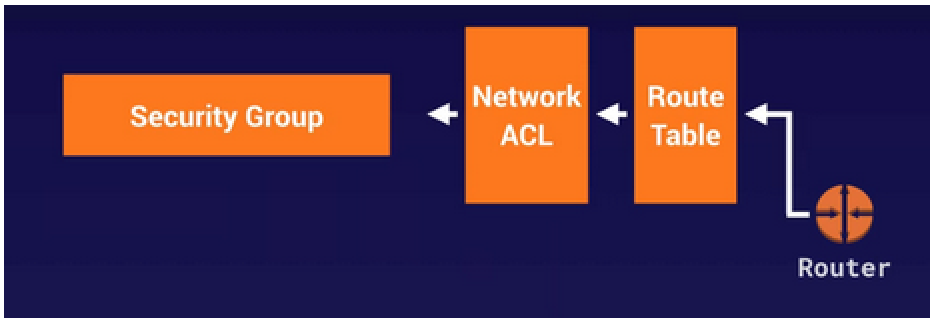
Virtual Private Cloud (VPC)

VPC Simplified:

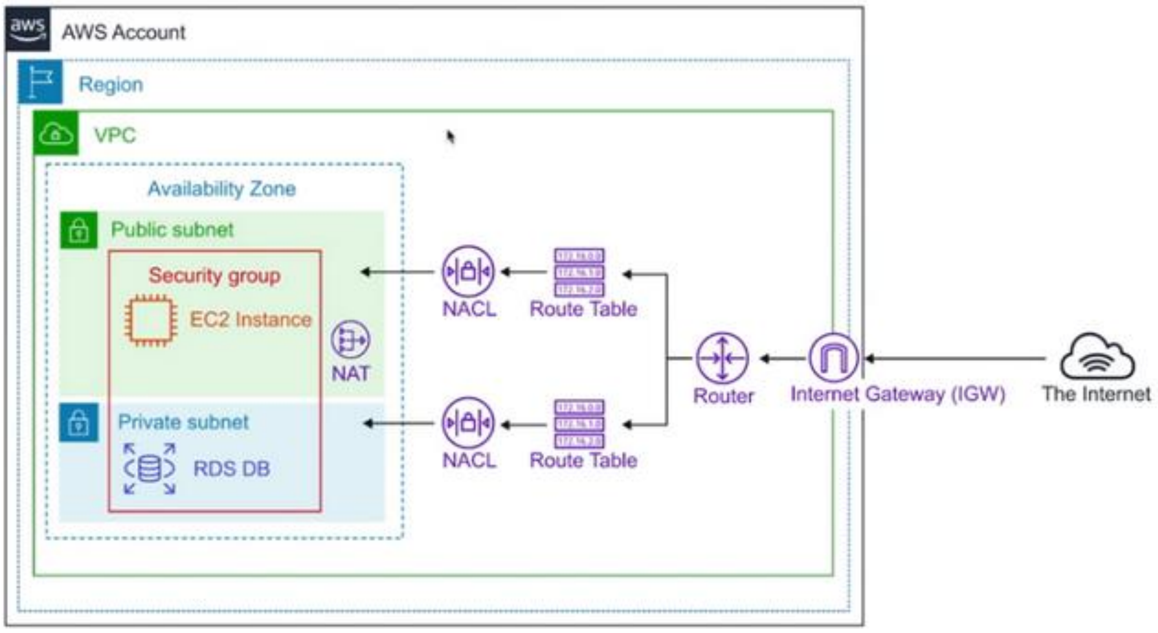
VPC lets you provision a logically isolated section of the AWS cloud where you can launch services and systems within a virtual network that you define. By having the option of selecting which AWS resources are public facing and which are not, VPC provides much more granular control over security.

VPC Key Details:

- You can think of VPC as your own virtual datacenter in the cloud. You have complete control of your own network; including the IP range, the creation of sub-networks (subnets), the configuration of route tables and the network gateways used.
- You can then launch EC2 instances into a subnet of your choosing, select the IPs to be available for the instances, assign security groups for them, and create Network Access Control Lists (NACLs) for the subnets themselves as additional protection.
- This customization gives you much more control to specify and personalize your infrastructure setup. For example, you can have one public-facing subnet for your web servers to receive HTTP traffic and then a different private-facing subnet for your database server where internet access is forbidden.
- You use subnets to efficiently utilize networks that have a large number of hosts
- VPCs come with defense in depth by design. From the sub-network (NACLs) down to the individual server (security group) and further down to the application itself (secure coding practices), you can set up multiple levels of protection against malicious users and programs.
- The default VPC for your AWS environment permits all subnets to have a route out to the internet meaning all subnets in the default VPC are internet accessible. The default setting allows you to immediately deploy instances and each EC2 instance will have both a public and private IP address.
- There is one default VPC per region. However, you can have as many custom VPCs as you want and all are private by default.
- When you create a custom VPC, new subnets are not created by default. You must create them separately. The same is true for an internet gateway. If you want your VPC to have internet access, you need to also create the gateway so that the network can be reached publicly by the world.
- Because of this, when you create an IGW it will initially be in an detached state. You will need to manually assign it to the custom VPC.
- Once you create a custom VPC however, the following are created by default:
    - a route table
    - a NACL
    - a security group



- These components, which will be explained in further depth in case they are not already known, actually correspond to the traffic flow for how data will reach your instances. Whether the traffic originates from outside of the VPC or from within it, it must first go through the route table by way of the router in order to know where the desired destination is. Once that is known, the traffic then passes through subnet level security as described by the NACL. If the NACL deems the traffic as valid, the traffic then passes through to the instance level security as described by the security group. If the traffic hasn't been dropped at this point, only then will it reach its intended instance.
- The VPC Wizard is an automated tool that is useful for creating custom VPCs.
- You can have your VPC on dedicated hardware so that the network is exclusive at the physical level, but this option is extremely expensive. Fortunately, if a VPC is on dedicated hosting it can always be changed back to the default hosting. This can be done via the AWS CLI, SDK or API. However, existing hosts on the dedicated hardware must first be in a `stopped` state.
- When you create a VPC, you must assign it an IPv4 CIDR block. This CIDR block is a range of private IPv4 addresses that will be inherited by your instances when you create them.
- The IP range of a default VPC is always **/16**.
- When creating IP ranges for your subnets, the **/16** CIDR block is the largest range of IPs that can be used. This is because subnets must have just as many IPs or fewers IPs than the VPC it belongs to. A **/28** CIDR block is the smallest IP range available for subnets.
- With CIDR in general, a **/32** denotes a single IP address and **/0** refers to the entire network The higher you go in CIDR, the more narrow the IP range will be.
- The above information about IPs is in regards to both public and private IP addresses.
- Private IP addresses are not reachable over the Internet and instead are used for communication between the instances in your VPC. When you launch an instance into a VPC, a private IP address from the IPv4 address range of the subnet is assigned to the default network interface (eth0) of the instance.
- This means that all instances within a VPC has a private IP, but only those selected to communicate with the external world have a public IP.
- When you launch an instance into a subnet that has public access via an Internet Gateway, both a public IP address and a private IP address are created. The public IP address is instead assigned to the primary network interface (eth0) that's created for the instance. Externally, the public IP address is mapped to the private IP address through network address translation (NAT).
- You can optionally associate an IPv6 CIDR block with your VPC and subnets, and assign IPv6 addresses from that block to the resources in your VPC.
- VPCs are region specific and you can have up to five VPCs per region.
- By default, AWS is configured to have one subnet in each AZ of the regions where your application is.
- In an ideal and secure VPC architecture, you launch the web servers or elastic load balancers in the public subnet and the database servers in the private subnet.
- Here is an example of a hypotherical application sitting behind a typical VPC setup:



- Security groups can span subnets, but do not span VPCs. ICMP ensures that instances from one security group can ping others in a different security group. It is IPv4 and IPv6 compatible.

VPC Subnets:

- If a network has a large number of hosts without logically grouped subdivisions, managing the many hosts can be a tedious job. Therefore you use subnets to divide a network so that management becomes easier.
- When you create a subnet, be sure to specify which VPC you want to place it in. You can assign both IPv4 and IPv6 ranges to your subnets.
- The main benefits of subnets:
    - They improve traffic flow, and thus speed & performance of the entire network. An Internet gateway (IGW) receiving a packet and checking which of 5 subnets the packet should be delivered to is much faster than checking 100 instances individually. And if the destination of a packet is within the subnet from where it originates, the traffic stays inside the subnet and doesn't clutter the rest of the VPC.
    - Subnets function as logical groups to put your entities inside of. It makes it much easier to configure similar resources as a group instead of for every individual instance.
- Amazon always reserves five IP addresses within a subnet. The first four IP addresses and the last IP address of each subnet CIDR block will always be unavailable for use.

Network Access Control Lists:

- Network Access Control Lists (or NACLs) are like security groups but for subnets rather than instances. The main difference between security groups and NACLs is that security groups are *stateless*, meaning you can perform both allow and deny rules that may be divergent, depending if traffic is inbound or outbound, for that rule.

- The following table highlights the differences between NACLs and Subnets.

NACL	Security Group
Operates at the subnet level	Operates at the instance level
Supports allow rules and deny rules	Supports allow rules only
Is stateless: Return traffic must be explicitly allowed by rules	Is stateful: Return traffic is automatically allowed, regardless of any rules
We process rules in order, starting with the lowest numbered rule, when deciding whether to allow traffic	We evaluate all rules before deciding whether to allow traffic
Automatically applies to all instances in the subnets that it's associated with (therefore, it provides an additional layer of defense if the security group rules are too permissive)	Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on

- Because NACLs are stateless, you must also ensure that outbound rules exist alongside the inbound rules so that ingress and egress can flow smoothly.
- The default NACL that comes with a new VPC has a default rule to allow all inbounds and outbounds. This means that it exists, but doesn't do anything as all traffic passes through it freely.
- However, when you create a new NACL (instead of using the default that comes with the VPC) the default rules will deny all inbounds and outbounds.
- If you create a new NACL, you must associate whichever desired subnets to it manually so that they can inherit the NACL’s rule set. If you don’t explicitly assign a subnet to an NACL, AWS will associate it with your default NACL.
- NACLs are evaluated before security groups and you block malicious IPs with NACLs, not security groups.
- A subnet can only follow the rules listed by one NACL at a time. However, a NACL can describe the rules for any number of subnets. The rules will take effect immediately.
- Network ACL rules are evaluated by rule number, from lowest to highest, and executed immediately when a matching allow/deny rule is found. Because of this, order matters with your rule numbers.
- The lower the number of a rule on the list, the more seniority that rule will have. List your rules accordingly.
- If you are using NAT Gateway along with your NACL, you must ensure the availability of the NAT Gateway ephemeral port range within the rules of your NACL. Because NAT Gateway traffic can appear on any of range's ports for the duration of its connection, you must ensure that all possible ports are accounted for and open.
- NACL can have a small impact on how EC2 instances in a private subnet will communicate with any service, including VPC Endpoints.

NAT Instances vs. NAT Gateways:

- Attaching an Internet Gateway to a VPC allows instances with public IPs to directly access the internet. NAT does a similar thing, however it is for instances that do not have a public IP. It serves as an intermediate step which allow private instances to first masked their own private IP as the NAT's public IP before accessing the internet.

- You would want your private instances to access the internet so that they can have normal software updates. NAT prevents any initiating of a connection from the internet.
- **NAT instances** are individual EC2 instances that perform the function of providing private subnets a means to securely access the internet.
- Because they are individual instances, High Availability is not a built-in feature and they can become a choke point in your VPC. They are not fault-tolerant and serve as a single point of failure. While it is possible to use auto-scaling groups, scripts to automate failover, etc. to prevent bottlenecking, it is far better to use the NAT Gateway as an alternative for a scalable solution.
- **NAT Gateway** is a managed service that is composed of multiple instances linked together within an availability zone in order to achieve HA by default.
- To achieve further HA and a zone-independent architecture, create a NAT gateway for each Availability Zone and configure your routing to ensure that resources use the NAT gateway in their corresponding Availability Zone.
- NAT instances are deprecated, but still useable. NAT Gateways are the prefered means to achieve Network Address Translation.
- There is no need to patch NAT Gateways as the service is managed by AWS. You do need to patch NAT Instances though because they’re just individual EC2 instances.
- Because communication must always be initiated from yout private instances, you need a route rule to route traffic from a private subnet to your NAT gateway.
- Your NAT instance/gateway will have to live in a public subnet as your public subnet is the subnet configured to have internet access.
- When creating NAT instances, it is important to remember that EC2 instances have source/destination checks on them by default. What these checks do is ensure that any traffic it comes across must be either generated by the instance or be the intended recipient of that traffic. Otherwise, the traffic is dropped because the EC2 instance is neither the source nor the destination.
- So because NAT instances act as a sort of proxy, you *must* disable source/destination checks when musing a NAT instance.

Bastion Hosts:

- Bastion Hosts are special purpose computers designed and configured to withstand attacks. This server generally runs a single program and is stripped beyond this purpose in order to reduce attack vectors.
- The purpose of Bastion Hosts are to remotely access the instances behind the private subnet for system administration purposes without exposing the host via an internet gateway.
- The best way to implement a Bastion Host is to create a small EC2 instance that only has a security group rule for a single IP address. This ensures maximum security.
- It is perfectly fine to use a small instance rather than a large one because the instance will only be used as a jump server that connects different servers to each other.
- If you are going to be RDPing or SSHing into the instances of your private subnet, use a Bastion Host. If you are going to be providing internet traffic into the instances of your private subnet, use a NAT.
- Similar to NAT Gateways and NAT Instances, Bastion Hosts live within a public-facing subnet.

- There are pre-baked Bastion Host AMIs.

Route Tables:

- Route tables are used to make sure that subnets can communicate with each other and that traffic knows where to go.
- Every subnet that you create is automatically associated with the main route table for the VPC.
- You can have multiple route tables. If you do not want your new subnet to be associated with the default route table, you must specify that you want it associated with a different route table.
- Because of this default behavior, there is a potential security concern to be aware of: if the default route table is public then the new subnets associated with it will also be public.
- The best practice is to ensure that the default route table where new subnets are associated with is private.
- This means you ensure that there is no route out to the internet for the default route table. Then, you can create a custom route table that is public instead. New subnets will automatically have no route out to the internet. If you want a new subnet to be publically accessible, you can simply associate it with the custom route table.
- Route tables can be configured to access endpoints (public services accessed privately) and not just the internet.

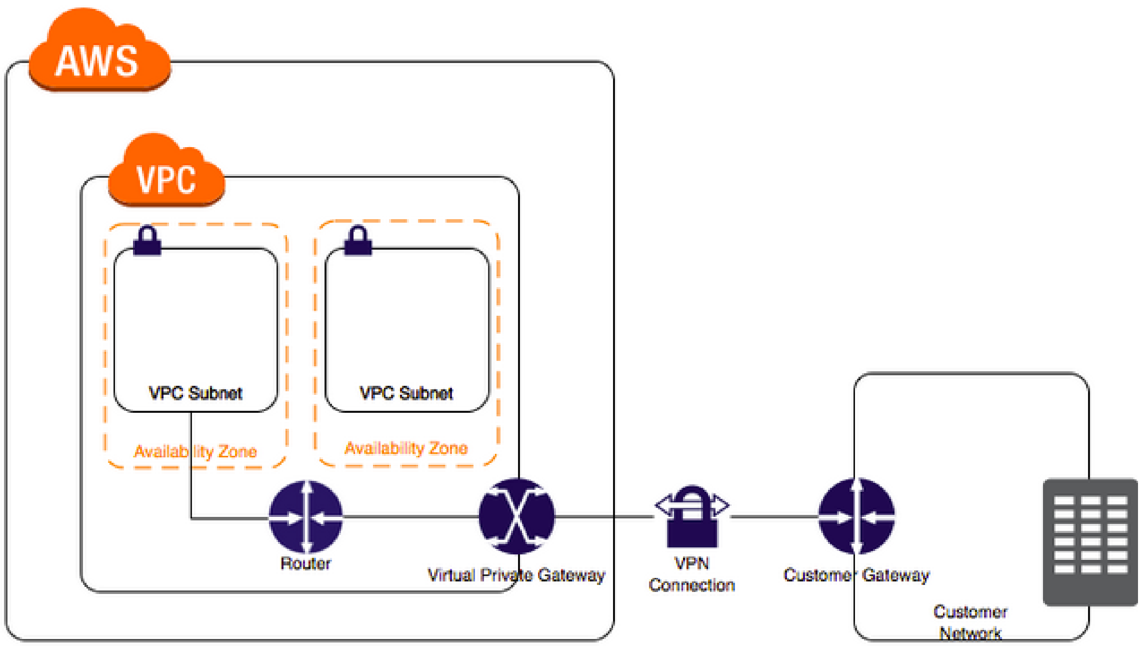
Internet Gateway:

- If the Internet Gateway is not attached to the VPC, which is the prerequisite for instances to be accessed from the internet, then natually instances in your VPC will not be reachable.
- If you want all of your VPC to remain private (and not just some subnets), then do not attach an IGW.
- When a Public IP address is assigned to an EC2 instance, it is effectively registered by the Internet Gateway as a valid public endpoint. However, each instance is only aware of its private IP and not its public IP. Only the IGW knows of the public IPs that belong to instances.
- When an EC2 instance initiates a connection to the public internet, the request is sent using the public IP as its source even though the instance doesn't know a thing about it. This works because the IGW performs its own NAT translation where private IPs are mapped to public IPs and vice versa for traffic flowing into and out of the VPC.
- So when traffic from the internet is destined for an instance's public IP endpoint, the IGW receives it and forwards the traffic onto the EC2 instance using its internal private IP.
- You can only have one IGW per VPC.
- **Summary**: IGW connects *your VPC with the internet.*

Virtual Private Networks (VPNs):

- VPCs can also serve as a bridge between your corporate data center and the AWS cloud. With a VPC Virtual Private Network (VPN), your VPC becomes an extension of your on-prem environment.

- Naturally, your instances that you launch in your VPC can't communicate with your own on-premise servers. You can allow the access by first:
    - attaching a virtual private gateway to the VPC
    - creating a custom route table for the connection
    - updating your security group rules to allow traffic from the connection
    - creating the managed VPN connection itself.
- To bring up VPN connection, you mustalso define a customer gateway resource in AWS, which provides AWS information about your customer gateway device. And you have to set up an Internet-routable IP address of the customer gateway's external interface.
- A customer gateway is a physical device or software application on the on-premise side of the VPN connection.
- Although the term "VPN connection" is a general concept, a VPN connection for AWS always refers to the connection between your VPC and your own network. AWS supports Internet Protocol security (IPsec) VPN connections.
- The following diagram illustrates a single VPN connection.



- The above VPC has an attached virtual private gateway (note: not an internet gateway) and there is a remote network that includes a customer gateway which you must configure to enable the VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private gateway.
- **Summary**: VPNs connect your *on-prem with your VPC* over the internet.

AWS DirectConnect:

- Direct Connect is an AWS service that establishes a dedicated network connection between your premises and AWS. You can create this private connectivity to reduce network costs, increase bandwidth, and provide more consistent network experience compared to regular internet-based connections.

- The use case for Direct Connect is high throughput workloads or if you need a stable or reliable connection
- VPN connects to your on-prem over the internet and DirectConnect connects to your on-prem off through a private tunnel.
- The steps for setting up an AWS DirectConnect connection:
    1. Create a virtual interface in the DirectConnect console. This is a public virtual interface.
    2. Go to the VPC console and then VPN connections. Create a customer gateway for your on-premise.
    3. Create a virtual private gateway and attach it to the desired VPC environment.
    4. Select VPN connections and create a new VPN connection. Select both the customer gateway and the virtual private gateway.
    5. Once the VPN connection is available, set up the VPN either on the customer gateway or the on-prem firewall itself
- Data flow into AWS via DirectConnect looks like the following: On-prem router -> dedicated line -> your own cage / DMZ -> cross connect line -> AWS Direct Connect Router -> AWS backbone -> AWS Cloud
- **Summary**: DirectConnect connects your *on-prem with your VPC* through a non-public tunnel.

VPC Endpoints:

- VPC Endpoints ensure that you can connect your VPC to supported AWS services without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect. Traffic between your VPC and other AWS services stay within the Amazon ecosystem and these Endpoints are virtual devices that are HA and without bandwidth constraints.
- These work basically by attaching an ENI to an EC2 instance that can easily communicate to a wide range of AWS services.
- **Gateway Endpoints** rely on creating entries in a route table and pointing them to private endpoints used for S3 or DynamoDB. Gateway Endpoints are mainly just a target that you set.
- **Interface Endpoints** use AWS PrivateLink and have a private IP address so they are their own entity and not just a target in a route table. Because of this, they cost $.01/hour. Gateway Endpoints are free as they're just a new route in to set.
- Interface Endpoint provisions an Elastic Network interface or ENI (think network card) within your VPC. They serve as an entry and exit for traffic going to and from another supported AWS service. It uses a DNS record to direct your traffic to the private IP address of the interface. Gateway Endpoint uses route prefix in your route table to direct traffic meant for S3 or DynamoDB to the Gateway Endpoint (think 0.0.0.0/0 -> igw).
- To secure your Interface Endpoint, use Security Groups. But to secure Gateway Endpoint, use VPC Endpoint Policies.
- **Summary**: VPC Endpoints connect your *VPC with AWS services* through a non-public tunnel.

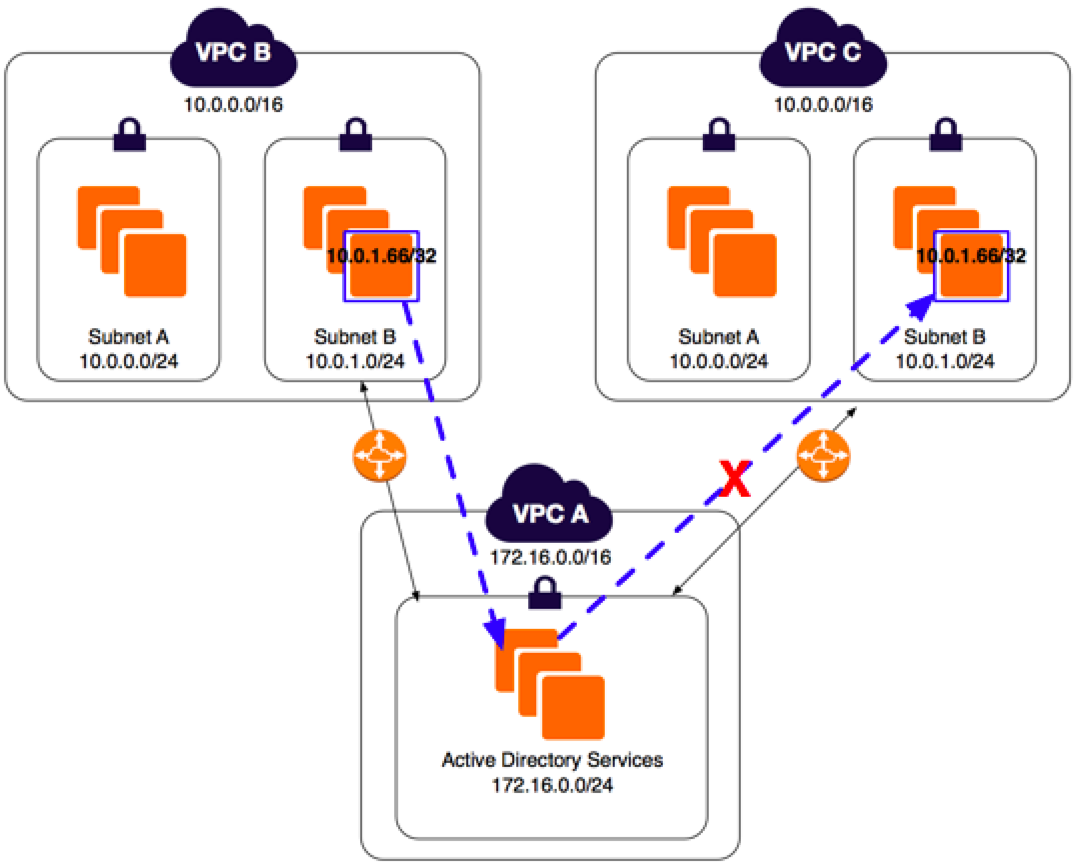
AWS PrivateLink:

- AWS PrivateLink simplifies the security of data shared with cloud-based applications by eliminating the exposure of data to the public Internet. AWS PrivateLink provides private connectivity between different VPCs, AWS services, and on-premises applications, securely on the Amazon network.

- It's similar to the AWS Direct Connect service in that it establishes private connections to the AWS cloud, except Direct Connect links on-premises environments to AWS. PrivateLink, on the other hand, secures traffic from VPC environments which are already in AWS.
- This is useful because different AWS services often talk to each other over the internet. If you do not want that behavior and instead want AWS services to only communicate within the AWS network, use AWS PrivateLink. By not traversing the Internet, PrivateLink reduces the exposure to threat vectors such as brute force and distributed denial-of-service attacks.
- PrivateLink allows you to publish an "endpoint" that others can connect with from their own VPC. It's similar to a normal VPC Endpoint, but instead of connecting to an AWS service, people can connect to your endpoint.
- Further, you'd want to use private IP connectivity and security groups so that your services function as though they were hosted directly on your private network.
- Remember that AWS PrivateLink applies to Applicatiosn/Services communicating with each other within the AWS network. For VPCs to communicate with each other within the AWS network, use VPC Peering.
- **Summary:** AWS PrivateLink connects your *AWS services with other AWS services* through a non-public tunnel.

VPC Peering:

- VPC peering allows you to connect one VPC with another via a direct network route using the Private IPs belonging to both. With VPC peering, instances in different VPCs behave as if they were on the same network.
- You can create a VPC peering connection between your own VPCs, regardless if they are in the same region or not, and with a VPC in an enirely different AWS account.
- VPC Peering is usually done in such a way that there is one central VPC that peers with others. Only the central VPC can talk to the other VPCs.
- You cannot do transitive peering for non-central VPCs. Non-central VPCs cannot go through the central VPC to get to another non-central VPC. You must set up a new portal between non-central nodes if you need them to talk to each other.
- The following diagram highlights the above idea. VPC B is free to communicate with VPC A with VPC Peering enabled between both. However, VPC B cannot continue the conversation with VPC C. Only VPC A can communicate with VPC C.



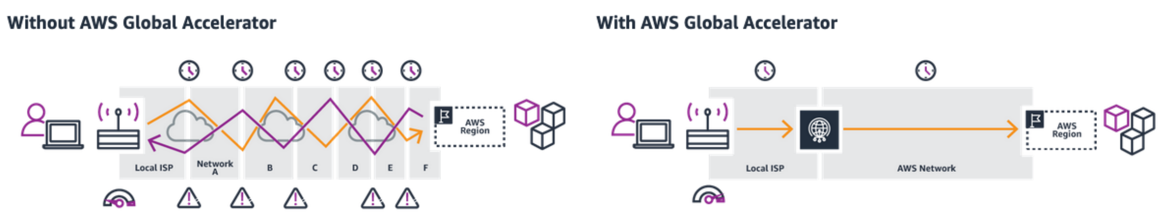
- It is worth knowing what VPC peering configurations are not supported:
    - Overlapping CIDR Blocks
    - Transitive Peering
    - Edge to Edge Routing through a gateway or connection device (VPN connection, Internet Gateway, AWS Direct Connect connection, etc.)
- You can peer across regions, but you cannot have one subnet stretched over multiple availability zones. However, you can have multiple subnets in the same availability zone.
- **Summary**: VPC Peering connects your *VPC to another VPC* through a non-public tunnel.

VPC Flow Logs:

- VPC Flow Logs is a feature that captures the IP information for all traffic flowing into and out of your VPC. Flow log data is sent to an S3 bucket or CloudWatch where you can view, retrieve, and manipulate this data.
- You can capture the traffic flow at various stages through its travel:
    - Traffic flowing into and out of the VPC (like at the IGW)
    - Traffic flowing into and out of the subnet
    - Traffic flowing into and out of the network interface of the EC2 instance (eth0, eth1, etc.)
- VPS Flow Logs capture packet metadata and not packet contents. Things like:
    - The source IP
    - The destination IP

    - The packet size
    - Anything which could be observed from outside of the packet.
- Your flow logs can be configured to log valid traffic, invalid traffic, or both
- You can have flow logs sourced from a different VPC compared to the VPC where your Flow Logs are. However, the other VPC must be peered via VPC Peering and under your account via AWS Organizations.
- You can customize your logs by tagging them.
- Once you create a Flow Log, you cannot change its config. You must make a new one.
- Not all IP traffic is monitored under VPC Flow Logs. The following is a list of things that are ignored by Flow Logs:
    - Query requests for instance metadata
    - DHCP traffic
    - Query requests to the AWS DNS server

- **AWS Global Accelerator:**
- AWS Global Accelerator accelerates connectivity to improve performance and availability for users. Global Accelerator sits on top of the AWS backbone and directs traffic to optimal endpoints worldwide. By default, Global Accelerator provides you two static IP addresses that you can make use of.
- Global Accelerator helps reduce the number of hops to get to your AWS resources. Your users just need to make it to an edge location and once there, everything will remain internal to the AWS global network. Normally, it takes many networks to reach the application in full and paths to and from the application may vary. With each hop, there is risk involved either in security or in failure.



It can take many networks to reach the application. Paths to and from the application may differ. Each hop impacts performance and can introduce risks.

Adding AWS Global Accelerator removes these inefficiencies. It leverages the Global AWS Network, resulting in improved performance.

- In summary, Global Accelerator is a fast/reliable pipeline between user and application.
- It's like going on a trip (web traffic) and stopping to ask for directions in possibly unsafe parts of town (multiple networks are visited which can increase security risks) as opposed to having a GPS (global accelerator) that leads you directly where you want to go (endpoint) without having to make unnecessary stops.
- It can be confused with Cloudfront, but CloudFront is a cache for content stemming from a distant origin server.
- While CloudFront simply caches static content to the closest AWS Point Of Presence (POP) location, Global accelerator will use the same Amazon POP to accept initial requests and routes them directly to the services.
- Route53's latency based routing might also appear similar to Global Accelerator, but Route 53 is for simply helping choose which region for the user to use. Route53 has nothing to do with actually providing a fast network path.
- Global Accelerator also provides fast regional failover.

Simple Queuing Service (SQS)

SQS Simplified:

SQS is a web-based service that gives you access to a message queue that can be used to store messages while waiting for a queue to process them. It helps in the decoupling of systems and the horizontal scaling of AWS resources.

SQS Key Details:

- The point behind SQS is to decouple work across systems. This way, downstream services in a system can perform work when they are ready to rather than when upstream services feed them data.
- In a hypothetical AWS environment running without SQS, *Application A* would pass *Application B* data regardless if Application B was ready to receive the info. With SQS however, there is an intermediary step where the data is stored temporarily in a buffer. It waits there until Application B pulls the temporarily stored data. SQS is not a push-based service so it is necessary for SQS to work in tandem with another service that queries it for information.
- There are two types of SQS queues; **standard** and **FIFO**. Standard queues may be received out of order based on message size or however else the SQS queues decide to optimize. FIFO queues guarantees that the order of messages that went into the queue is the same as the order of messages that leave it.
- Standard SQS queues guarantee that a message is delivered at least once and because of this, it is possible on occasion that a message might be delivered more than once due to the asynchronous and highly distributed architecture. With standard queues, you have a nearly unlimited number of transactions per second.
- FIFO SQS queues guarantee exactly-once processing and is limited to 300 transactions per second.
- Messages in the queue can be kept there from one minute to 14 days and the default retention period is 4 days.
- Visibility timeouts in SQS are the mechanism in which messages marked for delivery from the queue are given a timeframe to be fully received by a reader. This is done by temporarily making them invisible to other readers. If the message is not fully processed within the time limit, the message becomes visible again. This is another way in which messages can be duplicated. If you want to reduce the chance of duplication, increase the visibility timeout.
- The visibility timeout maximum is 12 hours.
- Always remember that the messages in the SQS queue will continue to exist even after the EC2 instance has processed it, until you delete that message. You have to ensure that you delete the message after processing to prevent the message from being received and processed again once the visibility timeout expires.
- An SQS queue can contain an unlimited number of messages.
- You cannot set a priority to the individual items in the SQS queue. If priority of messaging matters, create two separate SQS queues. The SQS queues for the priority message can be polled first by the EC2 Instances and once completed, the messages from the second queue can be processed next.

SQS Polling:

- Polling is the means in which you query SQS for messages or work. Amazon SQS provides short-polling and long-polling to receive messages from a queue. By default, queues use short polling.
- **SQS long-polling**: This polling technique will only return from the queue once a message is there, regardless if the queue is currently full or empty. This way, the reader needs to wait either for the timeout set or for a message to finally arrive. SQS long polling doesn't return a response until a message arrives in the queue, reducing your overall cost over time.
- **SQS short-polling**: This polling technique will return immediately with either a message that’s already stored in the queue or empty-handed.
- The ReceiveMessageWaitTimeSeconds is the queue attribute that determines whether you are using Short or Long polling. By default, its value is zero which means it is using short-polling. If it is set to a value greater than zero, then it is long-polling.
- Everytime you poll the queue, you incur a charge. So thoughtfully deciding on a polling strategy that fits your use case is important.

Simple Workflow Service (SWF)

SWF Simplified:

SWF is a web service that makes it easy to coordinate work across distributed application components. SWF has a range of use cases including media processing, web app backends, business process workflows, and analytical pipelines.

SWF Key Details:

- SWF is a way of coordinating tasks between application and people. It is a service that combines digital and human-oriented workflows.
- An example of a human-oriented workflow is the process in which Amazon warehouse workers find and ship your item as part of your Amazon order.
- SWF provides a task-oriented API and ensures a task is assigned only once and is never duplicated. Using Amazon warehouse workers as an example again, this would make sense. Amazon wouldn’t want to send you the same item twice as they'd lose money.
- The SWF pipeline is composed of three different worker applications that help to bring a job to completion:
    - SWF Actors are workers that trigger the beginning of a workflow.
    - SWF Deciders are workers that control the flow of the workflow once it's been started.
    - SWF Activity Workers are the workers that actually carry out the task to completion.
- With SWF, workflow executions can last up to one year compared to the 14 day maximum retention period for SQS.

Simple Notification Service (SNS)

SNS Simplified:

Simple Notification Service is a pushed-based messaging service that provides a highly scalable, flexible, and cost-effective method to publish a custom messages to subscribers who wish to be informed about a certain topic.

SNS Key Details:

- SNS is mainly used to send alarms or alerts.
- SNS provides topics for high-throughput, push-based, many-to-many messaging.
- Using Amazon SNS topics, your publisher systems can fan out messages to a large number of subscriber endpoints for parallel processing, including Amazon SQS queues, AWS Lambda functions, and HTTP/S webhooks. Additionally, SNS can be used to fan out notifications to end users using mobile push, SMS, and email.
- You can send these push notifications to Apple, Google, Fire OS, and Windows devices.
- 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. When you publish to a topic, SNS appropriately formats copies of that message to send to whichever kind of device.
- To prevent messages being lost, messages are stored redundantly across multiple AZs.
- There is no long or short polling involved with SNS due to the instantaneous pushing of messages
- SNS has flexible message delivery over multiple transport protocols and has a simple API.

Kinesis

Kinesis Simplified:

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information. With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications. Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly instead of having to wait until all your data is collected before the processing can begin.

Kinesis Key Details:

- Amazon Kinesis makes it easy to load and analyze the large volumes of data entering AWS.
- Kinesis is used for processing real-time data streams (data that is generated continuously) from devices constantly sending data into AWS so that said data can be collected and analyzed.
- It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.
- There are three different types of Kinesis:

    - Kinesis Streams
    - Kinesis Firehose
    - Kinesis Analytics
- Kinesis Streams works where the data producers stream their data into Kinesis Streams which can retain the data that enters it from one day up until 7 days. Once inside Kinesis Streams, the data is contained within shards.
- Kinesis Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources such as website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events. For example: purchase requests from a large online store like Amazon, stock prices, Netflix content, Twitch content, online gaming data, Uber positioning and directions, etc.
- Amazon Kinesis Firehose is the easiest way to load streaming data into data stores and analytics tools. When data is streamed into Kinesis Firehose, there is no persistent storage there to hold onto it. The data has to be analyzed as it comes in so it's optional to have Lambda functions inside your Kinesis Firehose. Once processed, you send the data elsewhere.
- Kinesis Firehose can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards you’re already using today.
- Kinesis Analytics works with both Kinesis Streams and Kinesis Firehose and can analyze data on the fly. The data within Kinesis Analytics also gets sent elsewhere once it is finished processing. It analyzes your data inside of the Kinesis service itself.
- Partition keys are used with Kinesis so you can organize data by shard. This way, input from a particular device can be assigned a key that will limit its destination to a specific shard.
- Partition keys are useful if you would like to maintain order within your shard.
- Consumers, or the EC2 instances that read from Kinesis Streams, can go inside the shards to analyze what is in there. Once finished analyzing or parsing the data, the consumers can then pass on the data to a number of places for storage like a DB or S3.
- The total capacity of a Kinesis stream is the sum of data within its constituent shards.
- You can always increase the write capacity assigned to your shard table.

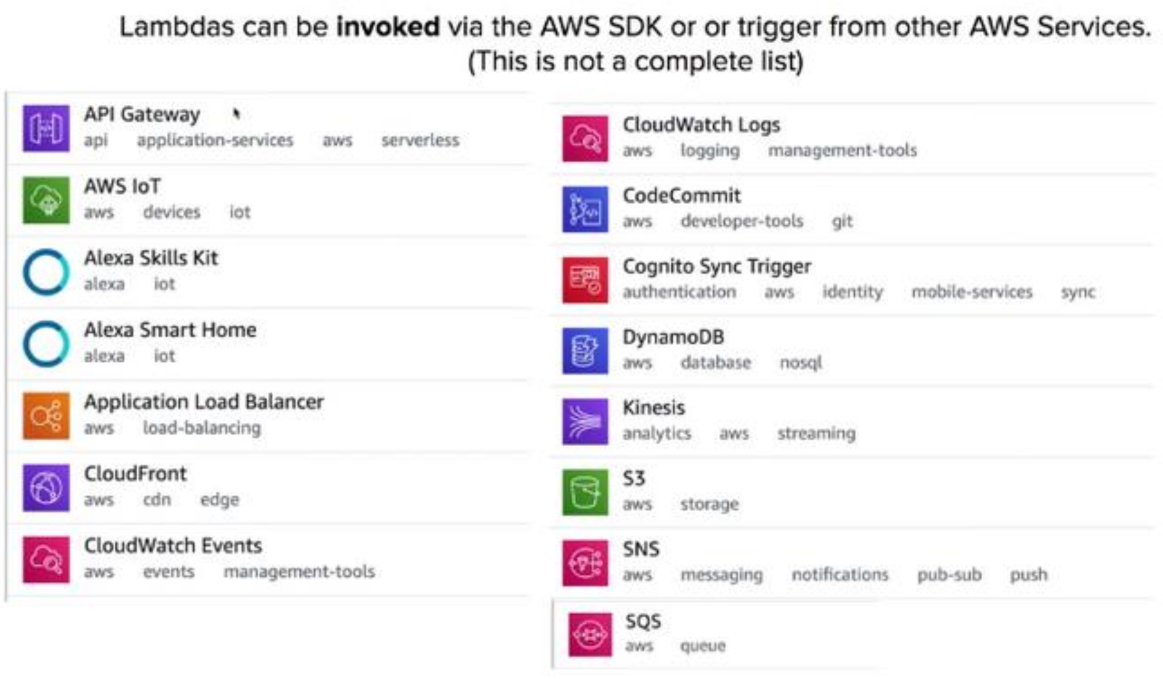
Lambda

Lambda Simplified:

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. You upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to be automatically triggered from other AWS services or be called directly from any web or mobile app.

Lambda Key Details:

- Lambda is a compute service where you upload your code as a function and AWS provisions the necessary details underneath the function so that the function executes successfully.
- AWS Lambda is the ultimate abstraction layer. You only worry about code, AWS does everything else.
- Lambda supports Go, Python, C#, PowerShell, Node.js, and Java
- Each Lambda function maps to one request. Lambda scales horizontally automatically.
- Lambda is priced on the number of requests and the first one million are free. Each million afterwards is $0.20.
- Lambda is also priced on the runtime of your code, rounded up to the nearest 100mb, and the amount of memory your code allocates.
- Lambda works globally.
- Lambda functions can trigger other Lambda functions.
- You can use Lambda as an event-driven service that executes based on changes in your AWS ecosystem.
- You can also use Lambda as a handler in response to HTTP events via API calls over the AWS SDK or API Gateway.



Lambdas can be **invoked** via the AWS SDK or or trigger from other AWS Services.
(This is not a complete list)

API Gateway
api application-services aws serverless

AWS IoT
aws devices iot

Alexa Skills Kit
alexa iot

Alexa Smart Home
alexa iot

Application Load Balancer
aws load-balancing

CloudFront
aws cdn edge

CloudWatch Events
aws events management-tools

CloudWatch Logs
aws logging management-tools

CodeCommit
aws developer-tools git

Cognito Sync Trigger
authentication aws identity mobile-services sync

DynamoDB
aws database nosql

Kinesis
analytics aws streaming

S3
aws storage

SNS
aws messaging notifications pub-sub push

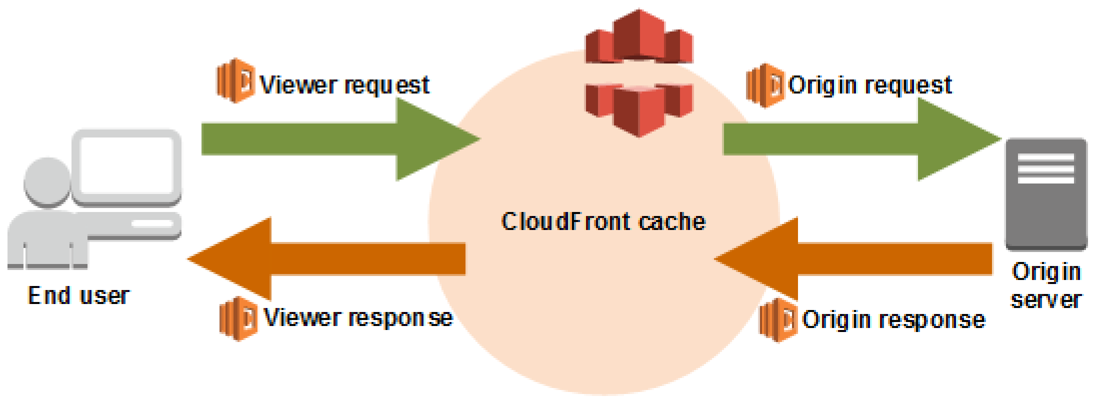
SQS
aws queue

- When you create or update Lambda functions that use environment variables, AWS Lambda encrypts them using the AWS Key Management Service. When your Lambda function is invoked, those values are decrypted and made available to the Lambda code.
- The first time you create or update Lambda functions that use environment variables in a region, a default service key is created for you automatically within AWS KMS. This key is used to encrypt environment variables. However, if you wish to use encryption helpers and use KMS to encrypt environment variables after your Lambda function is created, you must create your own AWS KMS key and choose it instead of the default key.

- To enable your Lambda function to access resources inside a private VPC, you must provide additional VPC-specific configuration information that includes VPC subnet IDs and security group IDs. AWS Lambda uses this information to set up elastic network interfaces (ENIs) that enable your function to connect securely to other resources within a private VPC.
- AWS X-Ray allows you to debug your Lambda function in case of unexpected behavior.

Lambda@Edge:

- You can use Lambda@Edge to allow your Lambda functions to customize the content that CloudFront delivers.
- It adds compute capacity to your CloudFront edge locations and allows you to execute the functions in AWS locations closer to your application's viewers. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests and responses at the following points:
    - After CloudFront receives a request from a viewer (viewer request)
    - Before CloudFront forwards the request to the origin (origin request)
    - After CloudFront receives the response from the origin (origin response)
    - Before CloudFront forwards the response to the viewer (viewer response)



- You'd use Lambda@Edge to simplify and reduce origin infrastructure.

API Gateway

API Gateway Simplified:

API Gateway is a fully managed service for developers that makes it easy to build, publish, manage, and secure entire APIs. With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on EC2) code running on AWS Lambda, or any web application.

API Gateway Key Details:

- Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management.
- Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.
- API Gateway does the following for your APIs:
  - Exposes HTTP(S) endpoints for RESTful functionality
  - Uses serverless functionality to connect to Lambda & DynamoDB
  - Can send each API endpoint to a different target
  - Runs cheaply and efficiently
  - Scales readily and effortlessly
  - Can throttle requests to prevent attacks
  - Track and control usage via an API key
  - Can be version controlled
  - Can be connected to CloudWatch for monitoring and observability
- Since API Gateway can function with AWS Lambda, you can run your APIs and code without needing to maintain servers.
- Amazon API Gateway provides throttling at multiple levels including global and by a service call.
  - In software, a throttling process, or a throttling controller as it is sometimes called, is a process responsible for regulating the rate at which application processing is conducted, either statically or dynamically.
  - Throttling limits can be set for standard rates and bursts. For example, API owners can set a rate limit of 1,000 requests per second for a specific method in their REST APIs, and also configure Amazon API Gateway to handle a burst of 2,000 requests per second for a few seconds.
  - Amazon API Gateway tracks the number of requests per second. Any requests over the limit will receive a 429 HTTP response. The client SDKs generated by Amazon API Gateway retry calls automatically when met with this response.
- You can add caching to API calls by provisioning an Amazon API Gateway cache and specifying its size in gigabytes. The cache is provisioned for a specific stage of your APIs. This improves performance and reduces the traffic sent to your back end. Cache settings allow you to control the way the cache key is built and the time-to-live (TTL) of the data stored for each method. Amazon API Gateway also exposes management APIs that help you invalidate the cache for each stage.
- You can enable API caching for improving latency and reducing I/O for your endpoint.
- When caching for a particular API stage (version controlled version), you cache responses for a particular TTL in seconds.
- API Gateway supports AWS Certificate Manager and can make use of free TLS/SSL certificates.
- With API Gateway, there are two kinds of API calls:
  - Calls to the API Gateway API to create, modify, delete, or deploy REST APIs. These are logged in CloudTrail.
  - API calls set up by the developers to deliver their custom functionality: These are not logged in CloudTrail.

Cross Origin Resource Sharing:

- In computing, the same-origin policy is an important concept where a web browser permits scripts contained in one page to access data from another page, but only if both pages have the same origin.
- This behavior is enforced by browsers, but is ignored by tools like cURL and PostMan.
- Cross-origin resource sharing (CORS) is one way the server at the origin can relax the same-origin policy. CORS allows sharing of restricted resources like fonts to be requested from another domain outside the original domain of where the first resource was shared from.
- CORS defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.
- If you ever come across an error that mentions that an origin policy cannot be read at the remote resource, then you need to enable CORS on API Gateway.
- CORS is enforced on the client (web browser) side.
- A common example of this issue is if you are using a site with Javascript/AJAX for multiple domains under API Gateway. You would need to ensure that CORS is enabled.
- CORS does not prevent XSS attacks, but does protect against CSRF attacks. What it does is controls who can use the data served by your endpoint. So if you have a weather website with callbacks to an API that checks the forecast, you could stop someone from writing a website that serves JavaScript calls into your API when they navigate to your website.
- When someone attempts the malicious calls, your browser will read the CORS headers and it will not allow the request to take place thus protecting you from the attack.

CloudFormation

CloudFormation Simplified:

CloudFormation is an automated tool for provisioning entire cloud-based environments. It is similar to Terraform where you codify the instructions for what you want to have inside your application setup (X many web servers of Y type with a Z type DB on the backend, etc). It makes it a lot easier to just describe what you want in markup and have AWS do the actual provisioning work involved.

CloudFormation Key Details:

- The main use case for CloudFormation is for advanced setups and production environments as it is complex and has many robust features.
- CloudFormation templates can be used to create, update, and delete infrastructure.
- The templates are written in YAML or JSON
- A full CloudFormation setup is called a stack.
- Once a template is created, AWS will make the corresponding stack. This is the living and active representation of said template. One template can create an infinite number of stacks.

- The *Resources* field is the only mandatory field when creating a CloudFormation template
- Rollback triggers allow you to monitor the creation of the stack as it's built. If an error occurs, you can trigger a rollback as the name implies.
- AWS Quick Starts is composed of many high-quality CloudFormation stacks designed by AWS engineers.
- An example template that would spin up an EC2 instance:

```
Resources:
  Instance: ## Logical Resource
    Type: 'AWS::EC2::Instance' ## This is what will be created
    Properties: ## Configure the resources in a particular way
      ImageId: !Ref LatestAmiId
      Instance Type: !Ref Instance Type
      KeyName: !Ref Keyname
```

- For any Logical Resources in the stack, CloudFormation will make a corresponding Physical Resources in your AWS account. It is CloudFormation's job to keep the logical and physical resources in sync.
- A template can be updated and then used to update the same stack.

ElasticBeanstalk

ElasticBeanstalk Simplified:

ElasticBeanstalk is another way to script out your provisioning process by deploying existing applications to the cloud. ElasticBeanstalk is aimed toward developers who know very little about the cloud and want the simplest way of deploying their code.

ElasticBeanstalk Key Details:

- Just upload your application and ElasticBeanstalk will take care of the underlying infrastructure.
- ElasticBeanstalk has capacity provisioning, meaning you can use it with autoscaling from the get-go. ElasticBeanstalk applies updates to your application by having a duplicate ready with the already updated version. This duplicate is then swapped with the original. This is done as a preventative measure in case your updated application fails. If the app does fail, ElasticBeanstalk will switch back to the original copy with the older version and there will be no downtime experienced by the users who are using your application.
- You can use ElasticBeanstalk to even host Docker as Elastic Beanstalk supports the deployment of web applications from containers. With Docker containers, you can define your own runtime environment, your own platform, programming language, and any application dependencies (such as package managers or tools) that aren't supported by other platforms. ElasticBeanstalk makes it easy to deploy Docker as Docker containers are already self-contained and include all the configuration information and software required to run.

AWS Organizations

AWS Organizations Simplified:

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

AWS Organizations Key Details:

- Best practices is to use the root account to manage billing only with separate accounts used to deploy resources.
- The point of AWS Organizations is to deploy permissions to the separate accounts underneath the root account and have those policies trickle down. AWS Organizations helps you centrally govern your environment as you grow and scale your workloads on AWS.
- You can use organizational units (OUs) to group similar accounts together to administer as a single unit. This greatly simplifies the management of your accounts.
- You can attach a policy-based control to an OU, and all accounts within the OU automatically inherit the policy. So if your company's developers all have their own sandbox AWS account, they can be treated as a single unit and be restricted by the same policies.
- With AWS Organizations, we can enable or disable services using Service Control Policies (SCPs) broadly on organizational units or more specifically on individual accounts
- Use SCPs with AWS Organizations to establish access controls so that all IAM principals (users and roles) adhere to them. With SCPs, you can specify *Conditions*, *Resources*, and *NotAction* to deny access across accounts in your organization or organizational unit. For example, you can use SCPs to restrict access to specific AWS Regions, or prevent deleting common resources, such as an IAM role used for your central administrators.

Miscellaneous

The following section includes services, features, and techniques that may appear on the exam. They are also extremely useful to know as an engineer using AWS. If the following items do appear on the exam, they will not be tested in detail. You'll just have to know what the meaning is behind the name. It is a great idea to learn each item in depth for your career's benefit, but it is not necessary for the exam.

What is the Amazon Cognito?

- Before discussing Amazon Cognito, it is first important to understand what Web Identity Federation is. Web Identity Federation lets you give your users access to AWS resources after they have successfully authenticated into a web-based identity provider such as Facebook, Google, Amazon, etc. Following a successful login into these services, the user is provided an auth code from the identity provider which can be used to gain temporary AWS credentials.

- Amazon Cognito is the Amazon service that provides Web Identity Federation. You don’t need to write the code that tells users to sign in for Facebook or sign in for Google on your application. Cognito does that already for you out of the box.
- Once authenticated into an identity provider (say with Facebook as an example), the provider supplies an auth token. This auth token is then supplied to cognito which responds with limited access to your AWS environment. You dictate how limited you would like this access to be in the IAM role.
- Cognito's job is broker between your app and legitimate authenticators.
- *Cognito User Pools* are user directories that are used for sign-up and sign-in functionality on your application. Successful authentication generates a JSON web token. Remember user pools to be user based. It handles registration, recovery, and authentication.
- *Cognito Identity Pools* are used to allow users temp access to direct AWS Services like S3 or DynamoDB. Identity pools actually go in and grant you the IAM role.
- SAML-based authentication can be used to allow AWS Management Console login for non-IAM users.
- In particular, you can use Microsoft Active Directory which implements Security Assertion Markup Language (SAML) as well.
- You can use Amazon Cognito to deliver temporary, limited-privilege credentials to your application so that your users can access AWS resources.
- Amazon Cognito identity pools support both authenticated and unauthenticated identities.
- You can retrieve a unique Amazon Cognito identifier (identity ID) for your end user immediately if you're allowing unauthenticated users or after you've set the login tokens in the credentials provider if you're authenticating users.
- When you need to easily add authentication to your mobile and desktop app, think Amazon Cognito.

What is AWS Resource Access Manager?

- AWS Resource Access Manager (RAM) is a service that enables you to easily and securely share AWS resources with any AWS account or within your AWS Organization. You can share AWS Transit Gateways, Subnets, AWS License Manager configurations, and Amazon Route 53 Resolver rules resources with RAM.
- Many organizations use multiple accounts to create administrative or billing isolation, and to limit the impact of errors as part of the AWS Organizations service.
- RAM eliminates the need to create duplicate resources in multiple accounts, reducing the operational overhead of managing those resources in every single account you own.
- You can create resources centrally in a multi-account environment, and use RAM to share those resources across accounts in three simple steps: create a Resource Share, specify resources, and specify accounts.
- RAM is available at no additional charge.

What is Athena?

- Athena is an interactive query service which allows you to interact and query data from S3 using standard SQL commands. This is beneficial for programmatic querying for the average developer. It is serverless, requires no provisioning, and you pay per query and per TB scanned. You basically turn S3 into a SQL supported database by using Athena.

- Example use cases:
    - Query logs that are dumped into S3 buckets as an alternative or supplement to the ELK stack
    - Setting queries to run business reports based off of the data regularly entering S3
    - Running queries on click-stream data to have further insight of customer behavior

What is AWS Macie?

- To understand Macie, it is important to understand PII or Personally Identifiable Information:
    - Personal data used to establish an individual's identity which can be exploited
    - Examples: Social Security number, phone number, home address, email address, D.O.B, passport number, etc.
- Amazon Macie is an ML-powered security service that helps you prevent data loss by automatically discovering, classifying, and protecting sensitive data stored in Amazon S3. Amazon Macie uses machine learning to recognize sensitive data such as personally identifiable information (PII) or intellectual property, assigns a business value, and provides visibility into where this data is stored and how it is being used in your organization.
- You can be informed of detections via the Macie dashboards, alerts, or reporting.
- Macie can also analyze CloudTrail logs to see who might have interacted with sensitive data.
- Macie continuously monitors data access activity for anomalies, and delivers alerts when it detects risk of unauthorized access or inadvertent data leaks.
- Macie has ability to detect global access permissions inadvertently being set on sensitive data, detect uploading of API keys inside source code, and verify sensitive customer data is being stored and accessed in a manner that meets their compliance standards.

What is AWS KMS?

- AWS Key Management Service (AWS KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. The master keys that you create in AWS KMS are protected by FIPS 140-2 validated cryptographic modules.
- AWS KMS is integrated with most other AWS services that encrypt your data with encryption keys that you manage. AWS KMS is also integrated with AWS CloudTrail to provide encryption key usage logs to help meet your auditing, regulatory and compliance needs.
- You can configure your application to use the KMS API to encrypt all data before saving it to disk.

What is AWS Secrets Manager?

- AWS Secrets Manager is an AWS service that makes it easier for you to manage secrets.

- Secrets can be database credentials, passwords, third-party API keys, and even arbitrary text. You can store and control access to these secrets centrally by using the Secrets Manager console, the Secrets Manager command line interface (CLI), or the Secrets Manager API and SDKs.
- In the past, when you created a custom application that retrieves information from a database, you typically had to embed the credentials (the secret) for accessing the database directly in the application. When it came time to rotate the credentials, you had to do much more than just create new credentials. You had to invest time to update the application to use the new credentials. Then you had to distribute the updated application. If you had multiple applications that shared credentials and you missed updating one of them, the application would break.
- Because of this risk, many customers have chosen not to regularly rotate their credentials, which effectively substitutes one risk for another (functionality vs. security).
- Secrets Manager enables you to replace hard-coded credentials in your code (including passwords), with an API call to Secrets Manager to retrieve the secret programmatically.
- This helps ensure that the secret can't be compromised by someone examining your code, because the secret simply isn't there.
- Also, you can configure Secrets Manager to automatically rotate the secret for you according to a schedule that you specify. This enables you to replace long-term secrets with short-term ones, which helps to significantly reduce the risk of compromise.

What is AWS STS?

- AWS Security Token Service (AWS STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources.
- Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.
- Temporary security credentials are short-term, as the name implies. They can be configured to last for anywhere from a few minutes to several hours. After the credentials expire, AWS no longer recognizes them or allows any kind of access from API requests made with them.

What is OpsWorks?

- AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers.
- OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments.
- OpsWorks has three offerings - AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.
- AWS OpsWorks Stacks lets you manage applications and servers on AWS and on-premises. With OpsWorks Stacks, you can model your application as a stack containing different layers, such as load balancing, database, and application server.

- OpsWorks Stacks is complex enough for you to deploy and configure Amazon EC2 instances in each layer or connect to other resources such as Amazon RDS databases.

What is Elastic Transcoder?

- A media transcoder in the cloud. Basically, it is a service that converts media files from their original format to the media format specified whether for phones, tablets, PCs, etc.
- Because of the built-in support for different media types, you can trust that the resulting quality will be good.
- With Elastic Transcoder, you pay per minute of the transcode job and the resolution of the finished work.

What is AWS Directory Service?

- AWS Directory Service provides multiple ways to use Amazon Cloud Directory and Microsoft Active Directory (AD) with other AWS services.
- Directories store information about users, groups, and devices, and administrators use them to manage access to information and resources.
- AWS Directory Service provides multiple directory choices for customers who want to use existing Microsoft AD or Lightweight Directory Access Protocol (LDAP)–aware applications in the cloud. It also offers those same choices to developers who need a directory to manage users, groups, devices, and access.

What is IoT Core?

- AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices.
- AWS IoT Core provides secure communication and data processing across different kinds of connected devices and locations so you can easily build IoT applications.

What is AWS WorkSpaces?

- Amazon WorkSpaces is a managed, secure Desktop-as-a-Service (DaaS) solution. You can use Amazon WorkSpaces to provision either Windows or Linux desktops in just a few minutes and quickly scale to provide thousands of desktops to workers across the globe.
- Amazon WorkSpaces helps you eliminate the complexity in managing hardware inventory, OS versions and patches, and Virtual Desktop Infrastructure (VDI), which helps simplify your desktop delivery strategy.
- With Amazon WorkSpaces, your users get a fast, responsive desktop of their choice that they can access anywhere, anytime, from any supported device.

What is AWS Fargate?

- AWS Fargate is a serverless compute engine for containers.
- The Fargate launch type allows you to run your containerized applications without the need to provision and manage the backend infrastructure. Just register your task definition and Fargate launches the container for you.

- It works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).
- Fargate makes it easy for you to focus on building your applications. It removes the need to provision and manage servers, lets you specify and pay for resources per application, and improves security through application isolation by design.

What is Amazon Elastic Container Service?

- Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service.
- Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure. With simple API calls, you can launch and stop container-enabled applications, query the complete state of your cluster, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes and IAM roles.
- You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements. You can also integrate your own scheduler or third-party schedulers to meet business or application specific requirements.
- You can choose to run your ECS clusters using AWS Fargate, which is serverless compute for containers. Fargate removes the need to provision and manage servers, lets you specify and pay for resources per application, and improves security through application isolation by design.

What is Amazon Elastic Kubernetes Service?

- Amazon Elastic Kubernetes Service (Amazon EKS) is a fully managed Kubernetes service. EKS runs upstream Kubernetes and is certified Kubernetes conformant so you can leverage all benefits of open source tooling from the community. You can also easily migrate any standard Kubernetes application to EKS without needing to refactor your code.
- Kubernetes is open source software that allows you to deploy and manage containerized applications at scale. Kubernetes groups containers into logical groupings for management and discoverability, then launches them onto clusters of EC2 instances. Using Kubernetes you can run containerized applications including microservices, batch processing workers, and platforms as a service (PaaS) using the same toolset on premises and in the cloud.
- Amazon EKS provisions and scales the Kubernetes control plane, including the API servers and backend persistence layer, across multiple AWS availability zones for high availability and fault tolerance. Amazon EKS automatically detects and replaces unhealthy control plane nodes and provides patching for the control plane.
- Without Amazon EKS, you have to run both the Kubernetes control plane and the cluster of worker nodes yourself. With Amazon EKS, you provision your worker nodes using a single command in the EKS console, CLI, or API, and AWS handles provisioning, scaling, and managing the Kubernetes control plane in a highly available and secure configuration. This removes a significant operational burden for running Kubernetes and allows you to focus on building applications instead of managing AWS infrastructure.
- You can run EKS using AWS Fargate, which is serverless compute for containers. Fargate removes the need to provision and manage servers, lets you specify and pay

for resources per application, and improves security through application isolation by design.

- Amazon EKS is integrated with many AWS services to provide scalability and security for your applications. These services include Elastic Load Balancing for load distribution, IAM for authentication, Amazon VPC for isolation, and AWS CloudTrail for logging.

What does pilot light mean?

- The term pilot light is often used to describe a disaster recovery scenario in which a minimal version of an environment is always running in the cloud.
- The idea of the pilot light is an analogy that comes from the gas heater. In a gas heater, a small flame that's always on and can quickly ignite the entire furnace to heat up a house. This scenario is similar to a backup-and-restore scenario.
- For example, with AWS you can maintain a pilot light by configuring and running the most critical core elements of your system in AWS. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core that has always been running.

What are Blue-Green deployments?

- One of the challenges with automating deployments is the cut-over from the final stage of testing to live production. You usually need to do this quickly in order to minimize downtime.
- The Blue-Green deployment approach does this by ensuring you have two production environments, as identical as possible. At any time one of them, let's say blue for the example, is live. As you prepare a new release of your software you do your final stage of testing in the green environment. Once the software is working in the green environment, you switch the router so that all incoming requests go to the green environment - the blue one is now idle.
- Blue-green deployment also gives you a rapid way to rollback - if anything goes wrong you switch the router back to your blue environment.
- CloudFormation and CodeDeploy (AWS's version of Jenkins) both support this deployment technique.

What is Amazon Data Lifecycle Manager?

- You can use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes.
- Automating snapshot management helps you to:
    - Protect valuable data by enforcing a regular backup schedule.
    - Retain backups as required by auditors or internal compliance.
    - Reduce storage costs by deleting outdated backups.
- Using Amazon DLM means that you no longer need to remember to take your EBS snapshots, thus reducing cognitive load on engineers.

What is Route Origin Authorization?

- You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. You continue to own the address range, but AWS advertises it on the Internet. After you bring the address range to AWS, it appears in your account as an address pool.
- You can then create an Elastic IP address from your address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers. This is also called "Bring Your Own IP Addresses (BYOIP)".
- To ensure that only you can bring your address range to your AWS account, you must authorize Amazon to advertise the address range and provide proof that you own the address range.
- The benefit of ROA is that you can migrate pre-existing applications to AWS without requiring your partners and customers to change their IP address whitelists.

What is Amazon MQ?

- Amazon MQ is a managed message broker service that makes it easy to set up and operate message brokers in the cloud.
- The service is used when migrating services and apps into the cloud from your on-prem which is how it differs from Amazon SQS.
- Amazon MQ supports durability-optimized brokers backed by Amazon EFS to support high availability and message durability, and throughput-optimized brokers backed by Amazon EBS to support high-volume applications that require low latency and high throughput.
- You can easily move from any message broker to Amazon MQ because you don’t have to rewrite any messaging code in your applications.
- Amazon MQ is suitable for enterprise IT pros, developers, and architects who are managing a message broker themselves–whether on-premises or in the cloud–and want to move to a fully managed cloud service without rewriting the messaging code in their applications.

What is AWS Config?

- AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.
- With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.
- AWS Config allows you to do the following: ·
    - Evaluate your AWS resource configurations for desired settings. ·
    - Get a snapshot of the current configurations of the supported resources that are associated with your AWS account. ·
    - Retrieve configurations of one or more resources that exist in your account. ·
    - Retrieve historical configurations of one or more resources. ·
    - Receive a notification whenever a resource is created, modified, or deleted.
    - View relationships between resources. For example, you might want to find all resources that use a particular security group.