
AWS

Guide

Cloud Computing

What is Cloud Computing?

The on-demand delivery of compute, databases storage, applications and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing.

6 Advantages of Cloud Computing

1. Trade capital expense for variable expense
2. Benefit from massive economies of scale
3. Stop guessing about capacity
4. Increased speed and agility
5. Stop spending money running and maintaining data centres
6. Go global in minutes

CapEx & OpEx

- CapEx (capital expenditure) is defined as business expenses incurred in order to create long-term benefits in the future, such as purchasing fixed assets like a building or equipment. Some examples of IT items that fall under this category would be whole systems and servers, printers and scanners, or air conditioners and generators. You buy these items once and they benefit your business for many, many years. Maintenance of such items is also considered CapEx, as it extends their lifetime and usefulness. Capex can also be defined as Total Cost of Ownership (TCO).
- OpEx (operating expenditure), the expenses to run day-to-day business, like services and consumable items that get used up and are paid for according to use. This includes printer cartridges and paper, electricity, and even yearly services like website hosting or domain registrations. These things are necessary for your business's success but are not considered major long-term investments like CapEx items.
- The cloud allows you to trade high initial CapEx (such as data centers and physical servers) for a variable OpEx model, and only pay for IT as you consume it. Plus, the variable OpEx expenses are much lower than what you would pay to do it yourself because of the massive economies of scale that AWS has created.

Types of Cloud Computing Deployments

1. Public Cloud – AWS, Azure, GCP
2. Private Cloud (On Premise) – You manage it in your datacenter. Openstack or Vmware
3. Hybrid – Mix of public and private cloud

(<http://kayleigholiver.com/aws-cloud-practitioner-cloud-computing-topics/>)

Cloud Migration Approaches

Cloud migration is the process of moving some or all your digital operations to your cloud. There are three main types of cloud migration you can perform:

- on-premises to cloud
- cloud to cloud
- cloud to on-premises

When performing any of these three migration types, there are five methods and strategies you can use. The strategies were first defined in the Gartner “5 Rs” model in 2011. These strategies are:

- Rehosting – moving applications to the cloud as-is. This is also sometimes referred to as ‘Lift and Shift’
- Replatform—moving applications to the cloud without major changes, but taking advantage of benefits of the cloud environment, for example, you may choose to modify the way your application interacts with the database to benefit from automation and a more capable database infrastructure
- Refactor—modifying applications to better support the cloud environment
- Rebuild—rewrite the application from scratch
- Replace—retire the application and replace it with a new cloud-native application

(<https://cloud.netapp.com/blog/cvo-blg-cloud-migration-approach-rehost-refactor-or-replatform>)

Types of Cloud Computing

1. Infrastructure As A Service (IaaS)
 - is a type of cloud computing offering in which a service provider provides the basic building blocks for cloud IT and give access to networking features, computers (virtual or on dedicated hardware), and data storage space.
 - Infrastructure as a Service provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.
 - The service provider leaves the running of server instances to the customer, they do not access to what is on your servers
2. Platform as a Service (PaaS)
 - Platform as a Service – is a type of cloud computing offering in which a service provider removes the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and allow you to focus on the deployment and management of your applications.
 - This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application, enabling them to develop, run, and manage business applications with less distraction.
 - Examples of PaaS include Amazon LightSail and AWS Elastic Beanstalk
3. Software As A Service (SaaS)
 - is a type of cloud computing offering in which a service provider provides you with a completed product that is run and managed by the service provider.
 - In most cases, people referring to Software as a Service are referring to end-user applications.
 - With a SaaS offering you do not have to think about how the service is maintained or how the underlying infrastructure is managed; you only need to think about how you will use that particular piece of software.
 - A common example of a SaaS application is web-based email where you can send and receive email without having to manage feature additions to the email product or maintaining the servers and operating systems that the email program is running on.

Concepts

Components, Workloads and Architecture

- A **component** is the code, configuration and AWS Resources that together deliver against a requirement. A component is often the unit of technical ownership, and is decoupled from other components.
- The term **workload** is used to identify a set of components that together deliver business value. A workload is usually the level of detail that business and technology leaders communicate about.
- We think about **architecture** as being how components work together in a workload. How components communicate and interact is often the focus of architecture diagrams.

Horizontal vs Vertical Scaling

Horizontal scaling means scaling by adding more machines to your pool of resources (also described as “scaling out”), whereas vertical scaling refers to scaling by adding more power (e.g. CPU, RAM) to an existing machine (also described as “scaling up”).

Throughput

Throughput is a measure of how many units of information a system can process in a given amount of time.

High-throughput computing (HTC) involves running many independent tasks that require a large amount of computing power. With HTC, users can run many copies of their software simultaneously across many different computers. What could have taken weeks before on one computer now takes mere hours on a HTC cluster.

Loose Coupling

- As application complexity increases, a desirable attribute of an IT system is that it can be broken into smaller, loosely coupled components. This means that IT systems should be designed in a way that reduces interdependencies—a change or a failure in one component should not cascade to other components.
- Your infrastructure also needs to have well defined interfaces that allow the various components to interact with each other only through specific, technology- agnostic interfaces. Modifying any underlying operations without affecting other components should be made possible.
- Subareas of loose coupling include the coupling of classes, interfaces, data, and services.
- Loose coupling is when each of the components of a system has, or makes use of, little or no knowledge of the definitions of other separate components.
- Loose coupling between services can also be done through asynchronous integration, which involves one component that generates events and another that consumes them. The two components do not integrate through direct point-to-point interaction, but usually through an intermediate durable storage layer. This approach decouples the two components and introduces additional resiliency. So, for example, if a process that is reading messages from the queue fails, messages can still be added to the queue to be processed when the system recovers.

Asynchronous Integration

Asynchronous Integration is integration where the data in a system does not have to be moved to another location immediately but can be moved at a later point in time. This means that the system sending a request doesn't have to wait for a reply in order to continue operating. This type of integration is especially useful when we have large volumes of data to process or when we don't expect any immediate response.

Agility

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.

The requirements might need to change. We are not talking about growth here but a change of way of doing things. May be they started with a static webpage and it turned out they now need a database instead. This is not elasticity. They don't need more computing power, they need an agile solution that can change overtime.

Agility is the practice of “building in” the ability to change quickly and inexpensively. The cloud not only makes these other practices practical but provides agility on its own. Infrastructure can be provisioned in minutes instead of months, and de-provisioned or changed just as quickly.

Granularity

The degree to which you have control or access over a given setting in a computer system. For example:

- In CloudWatch certain metrics are available at 1-minute granularity
- AWS Identity and Access Management (IAM) allows customers to provide granular access control to resources in AWS

Data Integrity

Data integrity is the accuracy, completeness, consistency (validity) and reliability of data throughout its lifecycle.

Compromised data, after all, is of little use to enterprises, not to mention the dangers presented by sensitive data loss. For this reason, maintaining data integrity is a core focus of many enterprise security solutions.

Data integrity can be compromised in several ways. Each time data is replicated or transferred, it should remain intact and unaltered between updates. Error checking methods and validation procedures are typically relied on to ensure the integrity of data that is transferred or reproduced without the intention of alteration.

Data integrity is also related to a security best practice of requiring that secret data remains secret (confidentiality) and unmodified (integrity/authenticity). This is related to data encryption which is the customer's responsibility.

Scalability & Elasticity

Scalability

Scalability is the ability of a software system to increase workload size without application service interruption or performance impact.

Elasticity

In cloud computing, elasticity is defined as "the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible".

Some cloud solutions can also be automatically adjusted to meet these needs. This means you can set them up to scale up or down automatically based on certain conditions, like when your cloud solution has too many resources of which some are being under-utilised or if you have too few resources and your solution is running out of processing power.

Scaling vs Elasticity

Scalability is a characteristic of a **software architecture** related to serving higher amount of workload, where elasticity is a characteristic of the **physical layer** below, entirely related to hardware budget optimizations.

Fault Tolerance, Availability & Reliability

Fault Tolerance

Fault tolerance refers to the ability of a system (computer, network, cloud cluster, etc.) to continue operating without interruption when one or more of its components fail.

The objective of creating a fault-tolerant system is to prevent disruptions arising from a single point of failure, ensuring business continuity of mission-critical applications or systems.

A fault tolerant system must be able to handle such failures and seamlessly recover without any long term negative impacts to business operations.

Availability

Availability refers to the percentage of time that the infrastructure, system or a solution remains operational under normal circumstances in order to serve its intended purpose. For cloud infrastructure solutions, availability relates to the time that the datacenter is accessible or delivers the intended IT service as a proportion of the duration for which the service is purchased. The mathematical formula for Availability is as follows:

Percentage of availability = (total elapsed time – sum of downtime)/total elapsed time

An SLA (Service-level agreement) of 99.999 percent availability (the famous five nines), the yearly service downtime could be as much as 5.256 minutes.

The numbers portray a precise image of the system availability, allowing organizations to understand exactly how much service uptime they should expect from IT service providers.

True high availability means that a resource is available from at least three different availability zones, however AWS currently only guarantees that a resource can be reached at two different availability zones.

Fault Tolerance vs Availability

A fault tolerant system is a critical system that requires being operational with zero downtime, while a highly available system can tolerate short interruptions in service. Both high availability and fault tolerance require the ability to detect failure.

A fault tolerant system typically has significantly higher costs than a highly available system. The higher costs are due to having physical redundancy added to the system. Redundancy is achieved by having mirrored components up and running, so failure of a component will not mean failure of the entire system.

Reliability

Reliability refers to the probability that the system will meet certain performance standards in yielding correct output for a desired time duration.

Reliability can be used to understand how well the service will be available in context of different real-world conditions. For instance, a cloud solution may be available with an SLA commitment of 99.999 percent, but vulnerabilities to sophisticated cyber-attacks may cause IT outages beyond the control of the vendor.

A common metric is to calculate the Mean Time Between Failures (MTBF). MTBF represents the time duration between a component failure of the system. Similarly, organizations may also evaluate the Mean Time To Repair (MTTR), a metric that represents the time duration to repair a failed system component such that the overall system is available as per the agreed SLA commitment

Availability vs Reliability

The measurement of Availability is driven by **time loss** whereas the measurement of Reliability is driven by the **frequency and impact** of failures. Mathematically, the Availability of a system can be treated as a function of its Reliability. In other words, Reliability can be considered a subset of Availability.

Durability

A system that is durable is able to perform its responsibilities over time, even when unexpected events may occur. For example, a durable storage system will reliably store data without data loss.

RTO

The recovery time objective (RTO) is the targeted duration of time between the event of failure and the point where operations resume.

RPO

A recovery point objective (RPO) is the maximum length of time permitted that data can be restored from, which may or may not mean data loss. It is the age of the files or data in backup storage required to resume normal operations if a computer system or network failure occurs.

[Virtual Machine](#)

Virtualization is the process of creating a software-based, or "virtual" version of a computer, with dedicated amounts of CPU, memory, and storage that are "borrowed" from a physical host computer—such as your personal computer—and/or a remote server—such as a server in a cloud provider's datacenter. A virtual machine is a computer file, typically called an image, that behaves like an actual computer. It can run in a window as a separate computing environment, often to run a different operating system—or even to function as the user's entire computer experience—as is common on many people's work computers. The virtual machine is partitioned from the rest of the system, meaning that the software inside a VM can't interfere with the host computer's primary operating system.

[Hypervisor](#)

A virtual machine monitor (VMM) that allows many virtual operating systems to run simultaneously on one computer system. These virtual machines are also called guest machines, and they all share the hardware of the physical machine, such as memory, processor, storage, and other related resources.

Managed Service & Serverless

Managed Service

- is a cloud feature that you can use without having to take care of the underlying hardware's administration. For instance, in the Amazon ecosystem, you will find AWS Fargate, AWS Lambda, AWS Aurora, Amazon DynamoDB, and Elastic Beanstalk, among others. What do all those services have in common? The service provider, and not your organization, is responsible for getting deployments up and running on these platforms.
- In managed services common activities are automated and implemented according to best practices, such as change requests, monitoring, patch management, security, and backup services. AWS Managed Services provide full-lifecycle services to provision, run, and support your infrastructure; and thus unburdens you from infrastructure operations so you can direct resources toward differentiating your business.
- AWS Managed Services takes care of all of your patching and backup activities to help keep your resources current and secure. When updates or patches are released by OS vendors, AWS Managed Services applies them in a timely and consistent manner to minimize the impact on your business. Critical security patches are applied immediately, while others are applied based on the patch schedule you request.

Serverless

- It is a way to describe the services, practices, and strategies that enable you to build more agile applications so you can innovate and respond to change faster. With serverless computing, infrastructure management tasks like capacity provisioning and patching are handled by AWS, so you can focus on only writing code that serves your customers. Serverless services like AWS Lambda come with automatic scaling, built-in high availability, and a pay-for-value billing model. Lambda is an event-driven compute service that enables you to run code in response to events from over 150 natively-integrated AWS and SaaS sources - all without managing any servers.
- Benefits:
 - Move from idea to market, faster - By eliminating operational overhead, your teams can release quickly, get feedback, and iterate to get to market faster.
 - Lower your costs - With a pay-for-value billing model, you never pay for over-provisioning and your resource utilization is optimized on your behalf.
 - Adapt at scale - With technologies that automatically scale from zero to peak demands, you can adapt to customer needs faster than ever.
 - Build better applications, easier - Serverless applications have built-in service integrations, so you can focus on building your application instead of configuring it.
 - Is a way to describe the services, practices, and strategies that enable you to build more agile applications so you can innovate and respond to change faster. With serverless computing, infrastructure management tasks like capacity provisioning and patching are handled by AWS, so you can focus on only writing code that serves your customers. Serverless services (like AWS Lambda) come with automatic scaling, built-in high availability, and a pay-for-value billing model

Serverless Services List

- AWS Lambda
- Amazon Fargate
- Amazon EventBridge
- AWS Step Functions
- Amazon SQS
- Amazon SNS
- Amazon API Gateway
- AWS AppSync
- Amazon S3
- Amazon DynamoDB
- Amazon RDS Proxy
- Amazon Aurora Serverless

What is the Difference between Managed Service and Serverless?

- If a service or product is "Serverless", that means that it is also "Managed". But not all managed services are serverless; serverless is a special kind of managed service.
- What is it about serverless that makes it special? You completely stop thinking about the different kinds of "servers" in your architectures.
 - You stop thinking about asking the file "server" for something; you instead ask the data storage service to get it for you
 - You stop thinking about talking with the database "server"; you instead ask the query service to process your query.
 - You stop thinking about running your application "server" on a "server" instance; you instead have the service run your processing code whenever it's needed
- You can still have lots of great managed services that are not serverless, where you still have to choose the right size for your servers, however the cloud provider runs and manages those servers for you.

Stateful / Stateless

- A stateful web service will keep track of the "state" of a client's connection and data over several requests. So for example, the client might login, select a user's account data, update their address, attach a photo, and change the status flag, then disconnect.
- In a stateless web service, the server doesn't keep any information from one request to the next. The client needs to do its work in a series of simple transactions, and the client has to keep track of what happens between requests. So in the above example, the client needs to do each operation separately: connect and update the address, disconnect. Connect and attach the photo, disconnect. Connect and change the status flag, disconnect.
- To handle the removal of instances without impacting your service, you need to ensure that your application instances are stateless. This means that all system and application state is stored and managed outside of the instances themselves.
- The essence of a stateless installation is that the scalable components are disposable, and configuration is stored away from the disposable components. A stateless web service is much simpler to implement, and can handle greater volume of clients.

API

An application programming interface (API) is a way for two or more computer programs to communicate with each other. In building applications, an API simplifies programming by abstracting the underlying implementation and only exposing objects or actions the developer needs. It allows for vastly different programs to interact much more seamlessly than would otherwise be possible.

Another purpose of APIs is to hide the internal details of how a system works, exposing only those parts a programmer will find useful and keeping them consistent even if the internal details later change.

An API may be custom-built for a particular pair of systems, or it may be a shared standard allowing interoperability among many systems.

Microservices

- are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs. Services are built for business capabilities and each service performs a single function. Because they are independently run, each service can be updated, deployed, and scaled to meet demand for specific functions of an application. Microservices architectures are typically faster to develop, enabling innovation and accelerating time-to-market for new features.
- Microservices contrast with monolithic architectures, where all processes are tightly coupled and run as a single service, meaning that if one process of the application experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features becomes more complex as the code base grows. This complexity limits experimentation and makes it difficult to implement new ideas. Monolithic architectures add risk for application availability because many dependent and tightly coupled processes increase the impact of a single process failure.

Event-Driven Computing

- Event-driven computing is a model in which subscriber services automatically perform work in response to events triggered by publisher services. This paradigm can be applied to automate workflows while decoupling the services that collectively and independently work to fulfil these workflows. Amazon SNS is an event-driven computing hub, in the AWS Cloud, that has native integration with several AWS publisher and subscriber services.

[Graceful Exit](#)

A graceful exit is a programming concept wherein a program detects a serious error condition and "exits gracefully" in a controlled manner as a result. Often the program prints a descriptive error message to a terminal or log as part of the graceful exit. Usually, code for a graceful exit exists when the alternative — allowing the error to go undetected and unhandled — would produce spurious errors or later anomalous behavior that would be more difficult for the programmer to repair and debug. The code associated with a graceful exit may also take additional steps, such as closing files, to ensure that the program leaves data in a consistent, recoverable state.

[Fallback](#)

- Failback is the process of restoring operations to a primary machine or facility after they have been shifted to a secondary machine or facility during failover.
- In a failback stage, the process uses something called change data, which represents changes made to the system under duress, or in other words, changes made only in the backup system. In failback, only the change data is sent to the original system. There is no need to copy an entire drive or set of drives; failback just adds what was recorded by the backup facility during the duration of the crisis.
- One of the implied characteristics of a failback system is that the process is done automatically.

[Failover](#)

- is a backup operational mode in which the functions of a system component (such as a processor, server, network, or database, for example) are assumed by secondary system components when the primary component becomes unavailable through either failure or scheduled down time.
- One of the implied characteristics of a failover system is that the process is done automatically.
- Two main types of failover:
 - Active-active failover - Use this failover configuration when you want all of your resources to be available the majority of the time. When a resource becomes unavailable, Route 53 can detect that it's unhealthy and stop including it when responding to queries. In active-active failover, all the records that have the same name, the same type (such as A or AAAA), and the same routing policy (such as weighted or latency) are active unless Route 53 considers them unhealthy. Route 53 can respond to a DNS query using any healthy record.
 - Active-passive failover - Use an active-passive failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Route 53 includes only the healthy primary resources. If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries.

Data Recovery Methods

- **Backup and Restore:** a simple, straightforward, cost-effective method that backs up and restores data as needed. Keep in mind that because none of your data is on standby, this method, while cheap, can be quite time-consuming.
- **Pilot Light:** The idea of the pilot light is an analogy that comes from gas heating. In that scenario, a small flame that's always on can quickly ignite the entire furnace to heat up a house. In this DR approach, you simply replicate part of your IT structure for a limited set of core services so that the pilot light environment seamlessly takes over in the event of a disaster. A small part of your infrastructure is always running simultaneously syncing mutable data (as databases or documents), while other parts of your infrastructure are switched off and used only during testing. Unlike a backup and recovery approach, you must ensure that your most critical core elements are already configured and running in the pilot light environment. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core.
- **Warm Standby:** The term warm standby is used to describe a DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud. A warm standby solution extends the pilot light elements and preparation. It further decreases the recovery time because some services are always running. By identifying your business-critical systems, you can fully duplicate these systems on in the warm standby environment and have them always on.
- **Hot Standby:** Also known as a Multi-Site Solution, this method fully replicates your company's data/applications between two or more active locations and splits your traffic/usage between them. If a disaster strikes, everything is simply rerouted to the unaffected area, which means you'll suffer almost zero downtime. However, by running two separate environments simultaneously, you will obviously incur much higher costs.

Defence in Depth

Defence in Depth is a concept which means having multiple layers of security controls placed throughout an IT system. Its intent is to provide redundancy in the event a security control fails or a vulnerability is exploited that can cover aspects of personnel, procedural, technical and physical security for the duration of the system's life cycle.

What is the Principle of Least Privilege?

The Principle of Least Privilege states that a subject should be given only those privileges needed for it to complete its task. If a subject does not need an access right, the subject should not have that access right.

Determine what users (and roles) need to do and then craft policies that allow them to perform only those tasks.

Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later.

This principle limits the damage that can result from an accident or error. It also reduces the number of potential interactions among privileged programs to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur.

Cyber Attacks

A cyberattack is any offensive maneuver that targets computer information systems, computer networks, infrastructures, or personal computer devices. An attacker is a person or process that attempts to access data, functions, or other restricted areas of the system without authorization, potentially with malicious intent. Below are a selection of different kinds of cyber attacks:

- DoS (Denial-of-Service)
 - A DoS attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash
- DDoS (Distributed Denial-of-Service)
 - occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers.[14] A DDoS attack uses more than one unique IP address or machines, often from thousands of hosts infected with malware
- HTTP Flood
 - a type of volumetric DDoS attack designed to overwhelm a targeted server with HTTP requests. The request can be either "GET" or "POST". The aim of the attack is to compel the server to allocate as many resources as possible to serving the attack, thus denying legitimate users access to the server's resources and thus impact the operation of web servers and any applications they are running.
- DNS Query Flood
 - DNS flood is a type of Distributed Denial of Service (DDoS) attack in which the attacker targets one or more Domain Name System (DNS) servers belonging to a given zone, attempting to hamper legitimate resolution of resource records of that zone and its sub-zones.
 - DNS servers are the "roadmap" of the Internet, helping requestors find the servers they seek. A DNS zone is a distinct portion of the domain name space in the Domain Name System (DNS). For each zone, administrative responsibility is delegated to a single server cluster.
- Reflection attacks
 - The attacker spoofs the victim's IP address and sends a request for information via UDP to servers known to respond to that type of request. The server answers the request and sends the response to the victim's IP address. From the servers' perspective, it was the victim who sent the original request. All the data from those servers pile up, congesting the target's Internet connectivity. With the maximized bandwidth, normal traffic cannot be serviced and clients cannot connect.
- Amplification attacks
 - is a reflection attack where the reply is larger than the request. It is any attack where an attacker is able to use an amplification factor to multiply its power, amplification attacks are "asymmetric", meaning that a relatively small number or low level of resources is required by an attacker to cause a significantly greater number or higher level of target resources to malfunction or fail.
 - DNS amplification attacks, for example, use DNS requests with a spoofed source address as the target, the DNS request is not sent back to the computer that issued the request, but instead to the victim. Through this method the attacker uses a modest number of machines with little bandwidth to send fairly substantial attacks.

- IP address spoofing
 - is the creation of Internet Protocol (IP) packets with a false source IP address, for the purpose of impersonating another computing system
- SYN (synchronise) floods
 - is a form of denial-of-service attack in which an attacker rapidly initiates a connection to a server without finalizing the connection. The server has to spend resources waiting for half-opened connections, which can consume enough resources to make the system unresponsive to legitimate traffic. Also known as a half-open attack
- UDP (User Datagram Protocol) floods
 - is a type of Denial of Service (DoS) attack in which the attacker overwhelms random ports on the targeted host with IP packets containing UDP packets. The receiving host checks for applications associated with these datagrams and—finding none—sends back a “Destination Unreachable” packet. As more and more UDP packets are received and answered, the system becomes overwhelmed and unresponsive to other clients.
 - In the framework of a UDP flood attack, the attacker may also spoof the IP address of the packets, both to make sure that the return “Destination Unreachable” packets don't reach their host, and to anonymize the attack.
- Packet Sniffing
 - is a process of monitoring and capturing all data packets passing through given network.
 - Sniffers can be used legitimately by network/system administrators to monitor and troubleshoot network traffic.
 - Attackers use sniffers to capture data packets containing sensitive information such as password, account information etc. Sniffers can be hardware or software installed in the system. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyse all of the network traffic.

Block Storage vs. Object Storage

What is Block Storage?

Block storage is the oldest and simplest form of data storage. Block storage stores data in fixed-sized chunks called — you guessed it — ‘blocks’. By itself, a block typically only houses a portion of the data. The application makes SCSI (Small Computer System Interface - set of standards for physically connecting and transferring data between computers and peripheral devices) calls to find the correct address of the blocks, then organizes them to form the complete file. Because the data is piecemeal, the address is the only identifying part of a block — there is no metadata associated with blocks. This structure leads to faster performance when the application and storage are local, but can lead to more latency when they are farther apart. The granular control that block storage offers makes it an ideal fit for applications that require high performance, such as transactional or database applications.

What is Object Storage?

Compared to block storage, object storage is much newer. With object storage, data is bundled with customizable metadata tags and a unique identifier to form objects. Objects are stored in a flat address space and there is no limit to the number of objects stored, making it much easier to scale out.

Each object has data, a key, and metadata.

- The object key uniquely identifies the object in the storage area.
- Object metadata is a set of name-value pairs. The metadata tags are a key advantage with object storage — they allow for much better identification and classification of data. You can think of objects as self-describing: They have descriptive labels assigned by the user or application that writes the object. Using a search application you can easily search for a specific object, even if the data itself is not easily searched (such as an image, or media clip, or data set).

For storing unstructured data, block storage vs object storage is no contest. Search capabilities and unlimited scale make object storage ideal for unstructured data, a classification that is currently expected to hit 44 zettabytes by 2020. Object storage is the only option that can effectively store this data at scale. Block storage has many uses within enterprises, but object storage is best equipped to handle the explosive growth of unstructured data. For a clearer side-by-side comparison of block storage vs object storage, take a look at the table below:

	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

(<https://cloudian.com/blog/object-storage-vs-block-storage/>)

websocket

is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. This contrasts with HTTP which is unidirectional where the client sends the request and the server then sends the response

AWS

General

How can you access the AWS platform?

You can access the AWS platform in 3 ways:

1. Using the Console - Graphical interface to access AWS features
2. Using the CLI (command line interface) - Lets you control AWS services programmatically from command line
3. Using the SDK - Enable you to access AWS using a variety of popular programming languages

Regions, Availability Zones, Edge Locations & Regional Edge Caches

- A Region is a collection of AZs that are geographically located closely to each other. Each Region is completely independent. Regions are distributed across the globe to allow for customers worldwide to access AWS resources with low latency. Large organizations can utilise multiple AWS regions. (As of 07 November 2022 there are 27 AWS regions worldwide). Regions contain at least 3 AZs.
- An Availability Zone (AZ) is an area with either one or more discrete Data Centres (where resources exist to power AWS services such as compute, storage and network resources), each with redundant power, networking, and connectivity, housed in separate facilities. Each AZ is geographically isolated from each other (to enhance resiliency against localised events e.g. floods). The AZs in a Region are connected through highly resilient very low-latency private fibre optic connections. Each AZ is connected to at least 2 other AZs. (As of 07 November 2022 there are 87 Availability Zones worldwide)
- Edge Locations are endpoints located in most of the major cities and highly populated areas around the world. They are specifically used as part of a global Content Delivery Network (CDN) by AWS services such as AWS CloudFront and AWS Lambda@Edge to cache data allowing for low latency end-user access. They are not used for the main AWS infrastructure. (As of 07 November 2022 there are 400+ Edge Locations worldwide)
- Regional Edge Cache - These sit between your Origin servers and the Edge Locations. A Regional Edge Cache has a larger cache-width than each of the individual Edge Locations, and because data expires from the cache at the Edge Locations, the data is retained at the Regional Edge Caches. Therefore, when data is requested at the Edge Location that is no longer available, the Edge Location can retrieve the cached data from the Regional Edge Cache instead of the Origin servers, which would have a higher latency.

Local Zone

Local Zone - an AWS infrastructure deployment that places select services closer to your end users. A Local Zone is an extension of a Region that is in a different location from your Region. It provides a

high-bandwidth backbone to the AWS infrastructure and is ideal for latency-sensitive applications, for example machine learning.

Why bother with multi-region architectures?

There are three reasons why you would want to have a multi-region architecture:

- Improve latency for end-users - The closer your backend origin is to end-users, the better the experience. Content Delivery Networks (CDN) like Amazon CloudFront have successfully been used to speed up the delivery of content, especially static content (e.g., images, videos, JavaScript libraries, etc.) to end-users across the globe. Using a globally-distributed network of caching servers, static content is served as if it was local to consumers, thus improving the delivery of that static content. However, even if CloudFront solves the problem for much of your content, some more dynamic calls still need to be done on the backend, and it could be far away, adding precious milliseconds to the request. By using a multi-region architecture you reduce the physical distance between your most distant users and the resources they are trying to access, thus improving latency.
- Large scale disaster recovery using AWS regions - Most organizations try to implement High Availability (HA) instead of Disaster Recovery (DR) to guard them against any downtime of services. In case of HA, we ensure there exists a fallback mechanism for our services. The service that runs in HA is handled by hosts running in different availability zones but in the same geographical region. This approach, however, does not guarantee that our business will be up and running in case the entire region goes down. DR takes things to a completely new level, wherein you need to be able to recover from a different region that's separated by over 250 miles. Our DR implementation is an Active/Passive model, meaning that we always have minimum critical services running in different regions, but a major part of the infrastructure is launched and restored when required.
- Business requirements – Perhaps for regional compliance that require data and services to be regionally hosted, or perhaps for an entirely different business reason, companies will opt to rollout in multiple regions.

Which Services are Global, Regional and Availability Zone Based?

- IAM
 - Users, Groups, Roles, Accounts – **Global**
 - Same AWS accounts, users, groups and roles can be used in all regions
 - Key Pairs – **Global or Regional**
 - Amazon EC2 created key pairs are specific to the region
 - RSA key pair can be created and uploaded that can be used in all regions
- Virtual Private Cloud
 - VPC – **Regional**
 - VPC are created within a region
 - Subnet – **Availability Zone**
 - Subnet can span only a single Availability Zone
 - Security groups – **Regional**
 - A security group is tied to a region and can be assigned only to instances in the same region.
 - VPC Endpoints – **Regional**
 - You cannot create an endpoint between a VPC and an AWS service in a different region.
 - VPC Peering – **Regional**
 - VPC Peering can be performed across VPC in the same account of different AWS accounts. VPC Peering can span inter-region.
 - Elastic IP Address – Regional
 - Elastic IP address created within the region can be assigned to instances within the region only
- S3 – Global but Data is **Regional**
 - S3 buckets are created within the selected region
 - Objects stored are replicated across Availability Zones to provide high durability but are not cross region replicated unless done explicitly
- Route53 – **Global**
 - Route53 services are offered at AWS edge locations and are global
- DynamoDb – **Regional**
 - All data objects are stored within the same region and replicated across multiple Availability Zones in the same region
 - Data objects can be explicitly replicated across regions using cross-region replication
- WAF – **Global**
 - Web Application Firewall (WAF) services protects web applications from common web exploits are offered at AWS edge locations and are global
- CloudFront – **Global**
 - CloudFront is the global content delivery network (CDN) services are offered at AWS edge locations
- Storage Gateway – **Regional**
 - AWS Storage Gateway stores volume, snapshot, and tape data in the AWS region in which the gateway is activated

- EC2
 - Resource Identifiers – **Regional**
 - Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its region and can be used only in the region where you created the resource.
 - Instances – **Availability Zone**
 - An instance is tied to the Availability Zones in which you launched it. However, note that its instance ID is tied to the region.
 - EBS Volumes – **Availability Zone**
 - Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
 - EBS Snapshot – **Regional**
 - An EBS snapshot is tied to its region and can only be used to create volumes in the same region and has to be copied from One region to other if needed
 - AMIs – **Regional**
 - AMI provides templates to launch EC2 instances
 - AMI is tied to the Region where its files are located with Amazon S3. For using AMI in different regions, the AMI can be copied to other regions
 - Auto Scaling – **Regional**
 - Auto Scaling spans across multiple Availability Zones within the same region but cannot span across regions
 - Elastic Load Balancer – **Regional**
 - Elastic Load Balancer distributes traffic across instances in multiple Availability Zones in the same region
 - Cluster Placement Groups – **Availability Zone**
 - Cluster Placement groups can be span across Instances within the same Availability Zones

(<https://jayendrapatil.com/aws-global-vs-regional-vs-az-resources>)

ARN (Amazon Resource Name)

- uniquely identify AWS resources. We require an ARN when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.
- Can be either qualified or unqualified ARN. Qualified ARNs contain a version suffix, while unqualified ARNs do not.
 - Qualified ARN: arn:aws:lambda:aws-region:acct-id:function:helloworld:42
 - Unqualified ARN: arn:aws:lambda:aws-region:acct-id:function:helloworld

Endpoint

- The URL of the entry point for an AWS web service. It may include a region code if the service supports regions
- For example: <https://awsexamplebucket/s3-us-west-2.amazonaws.com/docs/hello.txt>

What is an AWS Solutions Architect (SA)?

AWS Solutions Architects are individuals with years of experience architecting solutions across a wide variety of business verticals and use cases. Collectively AWS SAs have helped design and review thousands of customers' architectures on AWS. From this experience, they have identified best practices and core strategies for architecting systems in the cloud

They are responsible for building and integration of computer systems and information for meeting specific needs. Typically, this involves the integration of hardware and software for meeting the customer-defined purpose. Examination of current systems and architecture is also one of their responsibilities. They work with technical and business staff for recommending solutions for more effective systems.

The main duties of an AWS Solutions Architect are:

- Use technology to find a solution to business problems
- Decide which platform, framework or tech-stack should be used for the creation of a solution
- Designing the appearance of the application, what modules to use and the interaction between those modules
- Plan for scaling for future and it's the maintenance of the system
- Determine the risk associated with third-party platforms or frameworks

Well Architected Framework

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS. By using the Framework you will learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. It provides a way for you to consistently measure your architectures against best practices and identify areas for improvement.

The process for reviewing an architecture is a constructive conversation about architectural decisions, and is not an audit mechanism. We believe that having well-architected systems greatly increases the likelihood of business success.

The Five Pillars of a well Architecture Framework

- **Cost optimization**
 - The ability to run systems to deliver business value at the lowest price point.
- **Reliability**
 - The ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle.
- **Operational Excellence**
 - The ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value
- **Performance efficiency**
 - The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.
- **Security**
 - The security pillar encompasses the ability to protect data, systems, and assets to take advantage of cloud technologies to improve your security.

Easily remembered as **CROPS**



Figure 1: Harvesting some best practices

When architecting technology solutions, if you neglect the five pillars it can become challenging to build a system that delivers on your expectations and requirements. Incorporating these pillars into your architecture will help you produce stable and efficient systems. This will allow you to focus on the other aspects of design, such as functional requirements.

The AWS Well-Architected Framework documents a set of foundational questions that allow you to understand if a specific architecture aligns well with cloud best practices. The framework provides a consistent approach to evaluating systems against the qualities you expect from modern cloud-based systems, and the remediation that would be required to achieve those qualities. As AWS continues to evolve, and we continue to learn more from working with our customers, we will continue to refine the definition of well-architected.

(https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf)

Cost Optimization Design Principles & Best Practices

- **Cost-effective resources:** Using the appropriate instances and resources for your workload is key to cost savings. AWS offers a variety of flexible and cost-effective pricing options to acquire services in a way that best fits your needs.
 - Pay only for the computing resources that you require and increase or decrease usage depending on business requirements, not by using elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can also modify the demand, using a throttle, buffer, or queue to smooth the demand and serve it with less resources resulting in a lower cost, or process it at a later time with a batch service. When designing to modify demand and supply resources, actively think about the patterns of usage, the time it takes to provision new resources, and the predictability of the demand pattern.
 - AWS does the heavy lifting of data center operations like racking, stacking, and powering servers. It also removes the operational burden of managing operating systems and applications with managed services. This allows you to focus on your customers and business projects rather than on IT infrastructure.
- **Measure overall efficiency and expenditure:** The ability to align your organization to an agreed set of financial objectives, and provide your organization the mechanisms to meet them. Measure the business output of the workload and the costs associated with delivering it. Use this measure to know the gains you make from increasing output and reducing costs
- **Optimize over time:** As your requirements change, be aggressive in decommissioning resources, entire services, and systems that you no longer require. As AWS releases new services and features, it's a best practice
- to review your existing architectural decisions to ensure they continue to be the most cost effective. Your organization needs to dedicate time and resources to build capability in this new domain
 - The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual workload owners and drives efficient usage behaviour. This helps measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs

Reliability Design Principles and Best Practices

- **Workload Architecture:** A reliable workload starts with upfront design decisions for both software and infrastructure. Understand that distributed systems rely on communications networks to interconnect components, such as servers or services. Your workload must operate reliably despite data loss or latency in these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload, i.e. loose coupling.
 - **Foundations:** Foundational requirements are those whose scope extends beyond a single workload or project. Before architecting any system, foundational requirements that influence reliability should be in place. For example, you must have sufficient network bandwidth, adequate storage space and enough compute capacity.
 - **Scale horizontally:** to increase aggregate workload availability. Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall workload. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.
 - **Stop guessing capacity:** A common cause of failure in on-premises workloads is resource saturation, when the demands placed on a workload exceed the capacity of that workload (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and workload utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over or under-provisioning.
- **Change Management:** Changes to your workload or its environment must be anticipated and accommodated to achieve reliable operation of the workload. Changes include those imposed on your workload, such as spikes in demand, as well as those from within, such as feature deployments and security patches. Using AWS, you can monitor the behaviour of a workload and automate the response to these changes. With monitoring in place, your team will be automatically alerted when KPIs deviate from expected norms. Automatic logging of changes to your environment allows you to audit and identify actions that might have impacted reliability.
 - Changes to your infrastructure should be made using automation. The changes that need to be managed include changes to the automation, which then can be tracked and reviewed.
- **Failure Management / automatically recover from failure:** In any system of reasonable complexity, it is expected that failures will occur. Reliability requires that your workload be aware of failures as they occur and take action to avoid impact on availability. Workloads must be able to both withstand failures and automatically repair issues:
 - With AWS, you can take advantage of automation to react to monitoring data. For example, when a particular metric crosses a threshold, you can trigger an automated action to remedy the problem. Also, rather than trying to diagnose and fix a failed resource that is part of your production environment, you can replace it with a new one and carry out the analysis on the failed resource out of band.
 - Since the cloud enables you to stand up temporary versions of a whole system at low cost, you can use automation to simulate different failures or to recreate scenarios that led to failures before (chaos engineering) and observe the full recovery processes
 - Regularly back up your data and test your backup files to ensure that you can recover from both logical and physical errors.
 - Tracking KPIs will help you identify and mitigate single points of failure.
 - These approaches expose failure pathways that you can test and fix before a real failure scenario occurs, thus reducing risk.

Operational Excellence Design Principles and Best Practices

- **Organization:** Your teams need to have a shared understanding of your entire workload, their role in it, and shared business goals to set the priorities that will enable business success. Well-defined priorities will maximize the benefits of your efforts.
 - Evaluate internal and external customer needs involving key stakeholders, including business, development, and operations teams, to determine where to focus efforts.
 - Ensure that you are aware of guidelines or obligations defined by your organizational governance and external factors, such as regulatory compliance requirements and industry standards that may mandate or emphasize specific focus. Validate that you have mechanisms to identify changes to internal governance and external compliance requirements.
 - Evaluate threats to the business (for example, business risk and liabilities, and information security threats) and maintain this information in a risk registry.
- **Understand your workloads and their expected behaviours.** You will then be able design them to provide insight to their status and build the procedures to support them. Design your workload so that it provides the information necessary for you to understand its internal state (for example, metrics, logs, events, and traces) across all components in support of observability and investigating issues. Iterate to develop the telemetry necessary to monitor the health of your workload, identify when outcomes are at risk, and enable effective responses.
- **Define expected outcomes:** Successful operation of a workload is measured by the achievement of business and customer outcomes. Determine how success will be measured, and identify metrics that will be used in those calculations to determine if your workload and operations are successful.
- **Evolve & experiment:** You must learn, share, and continuously improve to sustain operational excellence. Dedicate work cycles to making continuous incremental improvements. Perform post incident analysis of all customer impacting events. Identify the contributing factors and preventative action to limit or prevent recurrence. Try to accelerate employees learning and keeps team members interested and engaged. Teams must grow their skill sets to adopt new technologies, and to support changes in demand and responsibilities. Set up regular game days to test, review and validate that all procedures are effective and that teams are familiar with them.
- **Make frequent, small, reversible changes:** Design workloads to allow components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).
- **Perform operations as code:** In the cloud, you can apply the same engineering discipline that you use for application code to your entire environment. You can define your entire workload (applications, infrastructure) as code and update it with code. You can implement your operations procedures as code and automate their execution by triggering them in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
- **Anticipate failure and learn from it:** Perform “pre-mortem” exercises to identify potential sources of failure so that they can be removed or mitigated. Test your failure scenarios and validate your understanding of their impact. Test your response procedures to ensure that they are effective, and that teams are familiar with their execution. Set up regular game days to test workloads and team responses to simulated events. Drive improvement through lessons learned from all operational events and failures. Share what is learned across teams and through the entire organization.

Performance Efficiency Design Principles and Best Practices

- **Democratize advanced technologies and use serverless architectures:** Make advanced technology implementation easier for your team by delegating complex tasks to your cloud vendor via serverless architectures. Rather than asking your IT team to learn about hosting and running a new technology, consider consuming the technology as a service. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require specialized expertise. In the cloud, these technologies become services that your team can consume, allowing your team to focus on product development rather than resource provisioning and management.
- **Go global in minutes:** Deploying your workload in multiple AWS Regions around the world allows you to provide lower latency and a better experience for your customers at minimal cost.
- **Selection:** The optimal solution for a particular workload varies, and solutions often combine multiple approaches. Well-architected workloads use multiple solutions and enable different features to improve performance. AWS resources are available in many types and configurations, which makes it easier to find an approach that closely matches your workload needs. Also understand how cloud services are consumed and always use the technology approach that aligns best with your workload goals. For example, consider data access patterns when you select database or storage approaches.
- **Trade-offs:** When you architect solutions, think about trade-offs to ensure an optimal approach. Depending on your situation, you could trade consistency, durability, and space for time or latency, to deliver higher performance. As you make changes to the workload, collect and evaluate metrics to determine the impact of those changes. Measure the impacts to the system and to the end-user to understand how your trade-offs impact your workload. Use a systematic approach, such as load testing, to explore whether the trade-off improves performance.
- **Monitoring:** After you implement your workload, you must monitor its performance so that you can remediate any issues before they impact your customers. Monitoring metrics should be used to raise alarms when thresholds are breached. Ensuring that you do not see false positives is key to an effective monitoring solution. Automated triggers avoid human error and can reduce the time it takes to fix problems. Plan for game days, where simulations are conducted in the production environment, to test your alarm solution and ensure that it correctly recognizes issues
- **Experiment more often:** With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.
- **Review:** Cloud technologies are rapidly evolving and you must ensure that workload components are using the latest technologies and approaches to continually improve performance. You must continually evaluate and consider changes to your workload components to ensure you are meeting its performance and cost objectives. New technologies, such as machine learning and artificial intelligence (AI), can allow you to reimagine customer experiences and innovate across all of your business workloads

Security Design Principles and Best Practices

- **Implement a strong identity foundation:** Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize identity management, and aim to eliminate reliance on long-term static credentials.
- **Keep people away from data:** Use mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data. This reduces the risk of mishandling or modification and human error when handling sensitive data.
- **Data Protection:** Data classification provides a way to categorize organizational data based on levels of sensitivity, and encryption protects data by way of rendering it unintelligible to unauthorized access. AWS provides multiple means for encrypting data at rest and in transit. We build features into our services that make it easier to encrypt your data. Additionally, AWS has designed storage systems for exceptional resiliency. For example, Amazon S3 is designed to provide 99.99999999% durability of objects over a given year.
- **Enable traceability:** Monitor, alert, and audit actions and changes to your environment in real time. Integrate log and metric collection with systems to automatically investigate and take action.
- **Apply security at all layers:** Infrastructure protection encompasses control methodologies (such as defence in depth with multiple security controls), this could include controls, such as, enforcing boundary protection; monitoring points of ingress and egress; implementing stateful and stateless packet inspection; comprehensive logging monitoring, and alerting.
- **Detection:** You can use detective controls to identify a potential security threat or incident. There are different types of detective controls. For example, processing logs, monitoring events and conducting an inventory of assets and their detailed attributes promotes more effective decision making to help establish operational baselines. You can also use internal auditing, an examination of controls related to information systems, to ensure that practices meet policies and requirements and that you have set the correct automated alerting notifications based on defined conditions. These controls are important reactive factors that can help your organization identify and understand the scope of anomalous activity.
- **Automate security best practices:** Automated software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively. Create secure architectures, including the implementation of controls that are defined and managed as code in version-controlled templates.
- **Incident Preparation & Response:** Prepare for an incident by having incident management and investigation policy and processes that align to your organizational requirements. Even with extremely mature preventive and detective controls, your organization should still put processes in place to respond to and mitigate the potential impact of security incidents. The architecture of your workload affects the ability of your teams to operate effectively during an incident, to isolate or contain systems and to restore operations to a known good state. Putting in place the tools and access ahead of a security incident, then routinely practicing incident response through game days, will help you ensure that your architecture can accommodate timely investigation and recovery.
- **Stay up to date:** Staying up to date with AWS and industry recommendations and threat intelligence helps you evolve your threat model and control objectives. Automating security processes, testing, and validation allow you to scale your security operations.

Other Cloud Design Principles

1. Stop guessing your capacity needs
 - If you make a poor capacity decision when deploying a workload, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need, and scale up and down automatically.
2. Test systems at production scale:
 - In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises.
3. Automate to make architectural experimentation easier:
 - Automation allows you to create and replicate your workloads at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary.
4. Allow for evolutionary architectures:
 - Allow for evolutionary architectures. In a traditional environment, architectural decisions are often implemented as static, onetime events, with a few major versions of a system during its lifetime. As a business and its context continue to evolve, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.
5. Drive architectures using data:
 - In the cloud, you can collect data on how your architectural choices affect the behaviour of your workload. This lets you make fact based decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.
6. Improve through game days:
 - A game day simulates a failure or event to test systems, processes, and team responses. The purpose is to actually perform the actions the team would perform as if an exceptional event happened. These should be conducted regularly so that your team builds "muscle memory" on how to respond. Your game days should cover the areas of operations, security, reliability, performance, and cost.
 - In AWS, your game days can be carried out with replicas of your production environment using AWS CloudFormation. This enables you to test in a safe environment that resembles your production environment closely.
 - Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made and can help develop organizational experience in dealing with events.

(https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf)

Accounts & IAM

What is an AWS account alias?

The account alias is a name you define to make it more convenient to identify your account, instead of using your AWS ID which is a twelve digit number. You can have one alias per AWS account. You can create an alias using the IAM APIs, AWS Command Line Tools, or the IAM console.

What is the Root Account?

Your root account is the email address you used to set up your AWS account. It has full admin access. Don't give away these account credentials. You should instead create a user for other individuals. Always secure the root account using multi-factor authentication.

(<http://kayleigholiver.com/aws-cloud-practitioner-iam/>)

What tasks require root user credentials?

- Change your account settings. This includes the account name, email address, root user password, and root user access keys. Other account settings, such as contact information, payment currency preference, and Regions, do not require root user credentials.
- View certain tax invoices. An IAM user with the aws-portal:ViewBilling permission can view and download VAT invoices from AWS Europe, but not AWS Inc or Amazon Internet Services Pvt. Ltd (AISPL).
- Close your AWS account.
- Restore IAM user permissions. If the only IAM administrator accidentally revokes their Own permissions, you can sign in as the root user to edit policies and restore those permissions.
- Change your AWS Support plan or Cancel your AWS Support plan. For more information, see IAM for AWS Support.
- Register as a seller in the Reserved Instance Marketplace.
- Configure an Amazon S3 bucket to enable MFA (multi-factor authentication) Delete.
- Edit or delete an Amazon S3 bucket policy that includes an invalid VPC ID or VPC endpoint ID.
- Sign up for GovCloud.

What is IAM?

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users.

IAM Users

An IAM user has permanent named operator with long-term credentials and is used to directly interact with AWS services, can be a human or machine. Users can be used globally.

IAM Roles

IAM roles allow you to delegate access with defined permissions to trusted entities without having to share long-term access keys, as such an IAM role does not have any credentials.

An IAM role is an AWS Identity and Access Management (IAM) entity with permissions to make AWS service requests. IAM roles cannot make direct requests to AWS services; they are meant to be assumed by authorized entities, such as IAM users, applications, or AWS services such as EC2. Roles can be used globally.

IAM Groups

An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users. All the users in an IAM group inherit the permissions assigned to the group. For example, you could have a group called Admins and give that group the types of permissions that administrators typically need. To set permissions in a group you can change a centrally stored access control policy and all users in the group will immediately inherit any changes.

(<http://kayleigholiver.com/aws-cloud-practitioner-iam/>)

[What is an IAM access control policy?](#)

AWS supports six types of policies: identity-based policies, resource-based policies, permissions boundaries, Organizations SCPs, ACLs, and session policies.

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

By default, IAM users, groups, and roles have no permissions; users with sufficient permissions must use a policy to grant the desired permissions.

Most policies are stored as JSON documents, for example:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:PutObject",  
        "s3:GetObject",  
        "s3:GetObjectVersion",  
        "s3:DeleteObject",  
        "s3:DeleteObjectVersion"  
      ],  
      "Resource": "arn:aws:s3:::example_bucket/example_folder/*"  
    }  
  ]  
}
```

The component parts of an IAM policy (i.e. Effect, Action, and Resource) are defined in a similar manner to an S3 bucket policy.

Managed policies are IAM resources that express permissions using the IAM policy language. You can create, edit, and manage separately from the IAM users, groups, and roles to which they are attached. After you attach a managed policy to multiple IAM users, groups, or roles, you can update that policy in one place and the permissions automatically extend to all attached entities. Managed policies are managed either by you (these are called customer managed policies) or by AWS (these are called AWS managed policies).

[What is a Managed Policy?](#)

An AWS managed policy is a access control policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases.

- Full access AWS managed policies such as AmazonDynamoDBFullAccess and IAMFullAccess define permissions for service administrators by granting full access to a service. Power-user AWS managed policies such as AWSCodeCommitPowerUser and AWSKeyManagementServicePowerUser are designed for power users.
- Partial-access AWS managed policies such as AmazonMobileAnalyticsWriteOnlyAccess and AmazonEC2ReadOnlyAccess provide specific levels of access to AWS services without allowing permissions management access level permissions. AWS managed policies make it easier for you to assign appropriate permissions to users, groups, and roles than if you had to write the policies yourself.
- One particularly useful category of AWS managed policies are those designed for job functions. These policies align closely to commonly used job functions in the IT industry. The intent is to make granting permissions for these common job functions easy. One key advantage of using job function policies is that they are maintained and updated by AWS as new services and API operations are introduced.
- You cannot change the permissions defined in AWS managed policies. AWS occasionally updates the permissions defined in an AWS managed policy.

[What is a Standalone Policy?](#)

You can create standalone policies that you administer in your own AWS account, which we refer to as customer managed policies. Standalone policies have their own Amazon Resource Name (ARN) that includes the policy name. A great way to create a customer managed policy is to start by copying an existing AWS managed policy. That way you know that the policy is correct at the beginning and all you need to do is customize it to your environment.

[What is an Inline Policy?](#)

An inline policy is a policy that's embedded in an IAM identity (a user, group, or role). That is, the policy is an inherent part of the identity. You can create a policy and embed it in an identity, either when you create the identity or later.

For more than one user, group or role to include the same policy, the policy must be copied to that user, group or role. Duplicating it and separating it from the original entirely.

IAM Best Practices

To help secure your AWS resources, follow these recommendations for the AWS Identity and Access Management (IAM) service.

- Lock away your AWS account root user access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Get started using permissions with AWS managed policies
- Use customer managed policies instead of inline policies
- Use access levels to review IAM permissions
- Configure a strong password policy for your users
- Enable MFA – These are not physical MFA tokens typically
- Use roles for applications that run on Amazon EC2 instances
- Use roles to delegate permissions
- Do not share access keys
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

Password Policies

In addition to manually creating individual passwords for your IAM users, you can create a password policy that applies to all IAM user passwords in your AWS account.

You can use a password policy to do these things:

- Set a minimum password length.
- Require specific character types, including uppercase letters, lowercase letters, numbers, and non-alphanumeric characters. Be sure to remind your users that passwords are case sensitive.
- Allow all IAM users to change their own passwords.
- Require IAM users to change their password after a specified period of time (enable password expiration).
- Prevent IAM users from reusing previous passwords.
- Force IAM users to contact an account administrator when the user has allowed his or her password to expire.

What is a federated user?

With identity federation, external identities are granted secure access to resources in your AWS account without having to create IAM users. These external identities can come from your corporate identity provider (such as Microsoft Active Directory or from the AWS Directory Service) or from a web IdP (identity provider), such as Amazon Cognito, Login with Amazon, Facebook, Google, or any OpenID Connect-compatible provider.

Federated users (external identities) are users you manage outside of AWS in your corporate directory, but to whom you grant access to your AWS account using temporary security credentials. They differ from IAM users, which are created and maintained in your AWS account.

AWS IAM Identity Center

IAM Identity Center allows you to centrally manage how users authenticate and access your AWS accounts to utilize resources and applications, from either a single account or from multiple accounts. So Identity Center acts as a central hub to control access for your workforce identities, which are also referred to as your workforce users. To help with the simplification of this management, Identity Center is effectively built on top of AWS IAM.

(successor to AWS Single Sign-On)

Support

AWS Support

Basic Support is included for all AWS customers and includes:

- Customer Service and Communities - 24x7 access to customer service, documentation, whitepapers, and [support forums](#).
- [AWS Trusted Advisor](#) –
 - Guidance to provision your resources following best practices to increase performance and improve security. AWS Trusted Advisor is an online tool that provides you real time guidance to help you provision your resources following AWS best practices. Trusted Advisor checks help optimize your AWS infrastructure, increase security and performance, reduce your overall costs, and monitor service limits. Whether establishing new workflows, developing applications, or as part of ongoing improvement, take advantage of the recommendations provided by Trusted Advisor on a regular basis to help keep your solutions provisioned optimally.
 - AWS Trusted Advisor analyzes your AWS environment and provides best practice recommendations in five categories:
 - **Performance:** AWS Trusted Advisor can improve the performance of your service by checking your service limits, ensuring you take advantage of provisioned throughput, and monitoring for overutilized instances.
 - **Service Quotas:** AWS Trusted Advisor checks for service usage that is more than 80% of the service quota.
 - AWS maintains service quotas (formerly called service limits) for each account to help guarantee the availability of AWS resources and prevent accidental provisioning of more resources than needed.
 - Some service quotas are raised automatically over time as you use AWS. However, most AWS services require that you request quota increases manually. You can use AWS Service Quotas console to view and request increases for most AWS quotas.
 - Values are based on a snapshot, so your current usage might differ. Limit and usage data can take up to 24 hours to reflect any changes.
 - **Cost optimization/Reduction:** AWS Trusted Advisor can save you money on AWS by eliminating unused and idle resources or by making commitments to reserved capacity.
 - **Security:** AWS Trusted Advisor can improve the security of your application by closing gaps, enabling various AWS security features, and examining your permissions.
 - **Fault Tolerance:** AWS Trusted Advisor can increase the availability and redundancy of your AWS application by take advantage of auto scaling, health checks, multi AZ, and backup capabilities.
 - Easily remembered as ...PQRST... (like the alphabet)

- AWS Basic Support and AWS Developer Support customers get access to 6 security checks (listed below) and 50 service limit checks (to see how close you are to exceeding use quotas):
 - S3 Bucket Permissions
 - Security Groups – Specific Ports Unrestricted
 - IAM Use
 - MFA on Root Account
 - EBS Public Snapshots
 - RDS Public Snapshots
- AWS Business Support and AWS Enterprise Support customers get access to all 115 Trusted Advisor checks (14 cost optimization, 17 security, 24 fault tolerance, 10 performance, and 50 service limits) and recommendations.

	Developer	Business	Enterprise
	<small>Recommended if you are experimenting or testing in AWS.</small>	<small>Recommended if you have production workloads in AWS.</small>	<small>Recommended if you have business and/or mission critical workloads in AWS.</small>
AWS Trusted Advisor Best Practice Checks	7 Core checks	Full set of checks	Full set of checks
Enhanced Technical Support	Business hours** email access to Cloud Support Associates Unlimited cases / 1 primary contact	24x7 phone, email, and chat access to Cloud Support Engineers Unlimited cases / unlimited contacts (IAM supported)	24x7 phone, email, and chat access to Cloud Support Engineers Unlimited cases / unlimited contacts (IAM supported)
Case Severity / Response Times*	General guidance: < 24 hours** System impaired: < 12 hours**	General guidance: < 24 hours System impaired: < 12 hours	General guidance: < 24 hours System impaired: < 12 hours Production system impaired: < 4 hours Production system down: < 1 hour Business-critical system down: < 15 minutes
Architectural Guidance	General	Contextual to your use-cases	Consultative review and guidance based on your applications
Programmatic Case Management		AWS Support API	AWS Support API
Third-Party Software Support		Interoperability and configuration guidance and troubleshooting	Interoperability and configuration guidance and troubleshooting
Proactive Programs and Services		Access to Infrastructure Event Management for additional fee	Infrastructure Event Management Well-Architected Reviews Access to proactive reviews, workshops, and deep dives
Technical Account Management			Designated Technical Account Manager (TAM) to proactively monitor your environment and assist with optimization and coordinate access to programs and AWS experts
Training			Access to online self-paced labs
Account Assistance			Concierge Support Team
Pricing	Greater of \$29 / month*** - or - 3% of monthly AWS usage <small>See pricing detail and example.</small>	Greater of \$100 / month*** - or - 10% of monthly AWS usage for the first \$0–\$10K 7% of monthly AWS usage from \$10K–\$80K 5% of monthly AWS usage from \$80K–\$250K 3% of monthly AWS usage over \$250K <small>See pricing detail and example.</small>	Greater of \$15,000 - or - 10% of monthly AWS usage for the first \$0–\$150K 7% of monthly AWS usage from \$150K–\$500K 5% of monthly AWS usage from \$500K–\$1M 3% of monthly AWS usage over \$1M <small>See pricing detail and example.</small>

*We will make every reasonable effort to respond to your initial request within the corresponding timeframes.

**Business hours are generally defined as 8:00 AM to 6:00 PM in the customer country as set in [My Account console](#), excluding holidays and weekends. These times may vary in countries with multiple time zones.

*** Plans are subject to a 30 day minimum term.

Note: if you work with an AWS partner and would like to learn more about Partner-led Support, [click here](#).

(<https://aws.amazon.com/premiumsupport/plans/>)

[AWS Support API](#)

The AWS Support API provides access to some of the features of the AWS Service Catalog. AWS provides this access for AWS customers who have a Business or Enterprise support plan. The service currently provides two different groups of operations:

- Support case management operations to manage the entire life cycle of your AWS support cases, from creating a case to resolving it. You can perform these tasks:
 - Open a support case.
 - Get a list and detailed information about recent support cases.
 - Narrow your search for support cases by dates and case identifiers, including cases that are resolved.
 - Add communications and file attachments to your cases, and add the email recipients for case correspondence.
 - Resolve your cases.
- Trusted Advisor operations to access the checks provided by AWS Trusted Advisor. You can perform these tasks:
 - Get names and identifiers for the checks that Trusted Advisor offers.
 - Request that a Trusted Advisor check be run against your account and resources.
 - Obtain summaries and detailed information for your Trusted Advisor checks.
 - Request that Trusted Advisor checks be refreshed.
 - Obtain the status of each Trusted Advisor check you have requested.
 - Also, AWS Support API supports CloudWatch Events for Trusted Advisor operations

[AWS Concierge](#)

Your AWS Concierge is a senior customer service agent who is assigned to your account when you subscribe to an Enterprise or qualified Reseller Support plan.

This Concierge agent is your primary point of contact for billing or account inquiries; when you don't know whom to call, they will find the right people to help.

In most cases, the AWS Concierge is available during regular business hours in your headquarters' geography. Outside of business hours, the global customer service team can assist you 24x7x365. The best way to contact the AWS Concierge is through the AWS Support Center.

[Technical Account Manager \(TAM\)](#)

Only available for Enterprise support level customers. A Technical Account Manager (TAM) is your designated technical point of contact who helps you onboard, provides advocacy and guidance to help plan and build solutions using best practices, coordinates access to subject matter experts, assists with case management, presents insights and recommendations on your AWS spend, workload optimization, and event management, and proactively keeps your AWS environment healthy.

AWS Account Abuse Team

The AWS Abuse team can assist you when AWS resources are used to engage in the following types of abusive behavior:

- Spam: You are receiving unwanted emails from an AWS-owned IP address, or AWS resources are used to spam websites or forums.
- Port scanning: Your logs show that one or more AWS-owned IP addresses are sending packets to multiple ports on your server, and you believe this is an attempt to discover unsecured ports.
- Denial-of-service (DoS) attacks: Your logs show that one or more AWS-owned IP addresses are used to flood ports on your resources with packets, and you believe that this is an attempt to overwhelm or crash your server or the software running on your server.
- Intrusion attempts: Your logs show that one or more AWS-owned IP addresses are used to attempt to log in to your resources.
- Hosting objectionable or copyrighted content: You have evidence that AWS resources are used to host or distribute illegal content or distribute copyrighted content without the consent of the copyright holder.
- Distributing malware: You have evidence that AWS resources are used to distribute software that was knowingly created to compromise or cause harm to computers or machines on which it is installed.

AWS Marketplace

The AWS Marketplace enables qualified partners to market and sell their software to AWS Customers. AWS Marketplace is an online software store that helps customers find, buy, and immediately start using the software and services that run on AWS.

AWS Marketplace is designed for Independent Software Vendors (ISVs), Value-Added Resellers (VARs), and Systems Integrators (SIs) who have software products they want to offer to customers in the cloud. Partners use AWS Marketplace to be up and running in days and offer their software products to customers around the world.

Customers can quickly launch pre-configured software with just a few clicks, and choose software solutions in Amazon Machine Images (AMIs) and software as a service (SaaS) formats, as well as other formats. Additionally, you can browse and subscribe to data products. Flexible pricing options include free trial, hourly, monthly, annual, multi-year, and BYOL (Bring Your Own License), and get billed from one source. AWS handles billing and payments, and charges appear on customers' AWS bill.

(<https://aws.amazon.com/partners/aws-marketplace>)

(<https://aws.amazon.com/about-aws/whats-new/2019/09/aws-marketplace-easier-to-find-solutions-from-aws-console/>)

[APN](#)

The **AWS Partner Network (APN)** is the global partner program for technology and consulting businesses who leverage Amazon Web Services to build solutions and services for customers. The APN helps companies build, market, and sell their AWS offerings by providing valuable business, technical, and marketing support. There are two main APN partner types:

- **APN Technology Partners** provide hardware, connectivity services, or software solutions that are hosted on, or integrated with, the AWS Cloud.
 - Hardware providers include original equipment manufacturers (OEMs) and semiconductor manufacturers.
 - Connectivity services providers include network carriers.
 - Software solution providers include SaaS providers and independent software vendors (ISVs).
- **AWS Consulting Partners** are professional services firms that help customers of all types and sizes design, architect, build, migrate, and manage their workloads and applications on AWS, accelerating their journey to the cloud. These professional services firms include system integrators, strategic consultancies, agencies, managed service providers (MSPs), and value-added resellers.
 - **The AWS Service Delivery Program** enables AWS customers to identify APN Consulting Partners with experience and a deep understanding of specific AWS services. These APN Partners follow best practices for AWS services and have proven success delivering AWS services to customers.
 - **AWS Managed Service Provider (MSP) Partners** provide customers full lifecycle solutions in cloud infrastructure and application migration. They offer support in four key areas: plan and design; build and migrate; run and operate; and optimize. AWS MSP Partners receive their designation by undergoing an extensive third-party validation audit that demonstrates next-generation managed service practices.

Other AWS Partner Groups & Services

[Infrastructure Event Management \(IEM\)](#)

AWS Infrastructure Event Management (IEM) offers architecture and scaling guidance and operational support during the preparation and execution of planned events, such as shopping holidays, product launches, and migrations. For these events, AWS Infrastructure Event Management will help you assess operational readiness, identify and mitigate risks, and execute your event confidently with AWS experts by your side. The program is included in the Enterprise Support plan and is available to Business Support customers for an additional fee.

[APN Delivery Partners](#)

APN Delivery Partners accelerate customers' cloud migration by providing technical support, personnel, and professional services. Delivery Partners take additional responsibilities for customers' migration implementation and project ownership.

[AWS Professional Services](#)

The **AWS Professional Services** organization is a global team of experts that can help you realize your desired business outcomes when using the AWS Cloud. We work together with your team and your chosen member of the AWS Partner Network (APN) to execute your enterprise cloud computing initiatives. Our team provides assistance through a collection of offerings which help you achieve specific outcomes related to enterprise cloud adoption. We also deliver focused guidance through our global specialty practices, which cover a variety of solutions, technologies, and industries. In addition to working alongside our customers, we share our experience through tech talk webinars, White Papers, and blog posts that are available to anyone.

- **Supplementing your team with specialized skills and experience** AWS Professional Services provides global specialty practices to support your efforts in focused areas of enterprise cloud computing e.g. Blockchain, quantum computing, robotics, space, Internet of Things & AI/Machine Learning. Specialty practices deliver targeted guidance through best practices, frameworks, tools, and services across solution, technology, and industry subject areas. Their deep expertise helps you take advantage of business benefits available with the AWS Cloud.

[Amazon Customer Engagement \(ACE\) Program](#)

The APN Customer Engagements (ACE) Program enables AWS Partners to build, grow, and drive successful customer engagements with AWS Sales. It provides Partners with a platform to collaborate with AWS Sales and Marketing teams, request funding, and technical support to help you co-sell with AWS.

AWS Partner Transformation Program

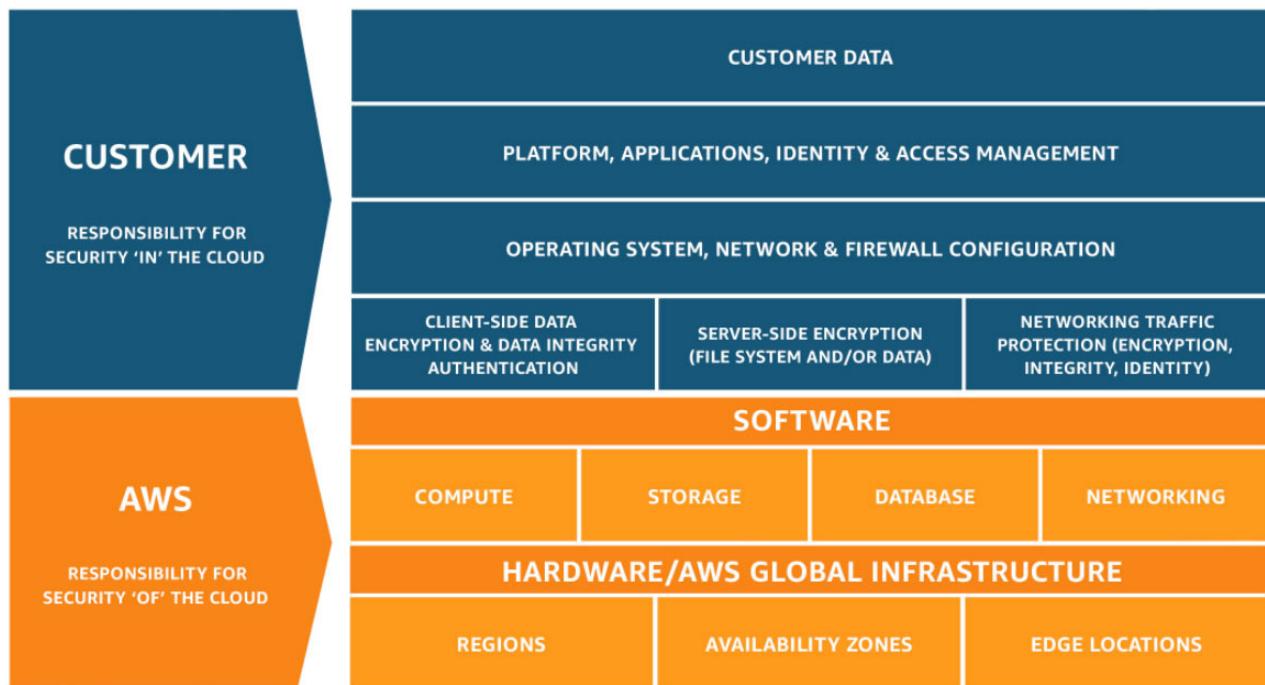
The AWS Partner Transformation Program (PTP) is a comprehensive assessment, training, and enablement program focused on helping you build a successful and profitable AWS Cloud business. Whether you are new to the cloud or in the advanced stages of building your cloud business, this program provides partners with the guidance to accelerate the development of your AWS skills and expertise to better serve your customers' journey to the cloud.

Through the PTP, AWS helps partners expedite cloud readiness in key business areas to help customers migrate to the cloud. The result is partner transformation, building an innovative cloud business for partners to better serve the ultimate customer.

The PTP is open to partners that are either new to cloud and need help with planning for cloud migration, or have started the process but need help in accelerating their journey. Every PTP partner receives a customized Transformation Plan to accelerate their journey to AWS and support in the execution of identified activities.

Security in AWS

Shared Responsibility Model



Security and Compliance is a shared responsibility between AWS and the customer. This shared model can help relieve the customer's operational burden as AWS operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. The nature of this shared responsibility also provides the flexibility and customer control that permits the deployment. As shown in the chart above, this differentiation of responsibility is commonly referred to as Security "of" the Cloud versus Security "in" the Cloud.

Also, note that the customer:

- assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided security group firewall.
- should carefully consider the services they choose as their responsibilities vary depending on the services used, the integration of those services into their IT environment, and applicable laws and regulations.
- is responsible for data configuration (i.e. encrypting data at rest and in transit)

Inherited, Shared and Customer Specific Controls

- Inherited Controls
 - Controls which a customer fully inherits from AWS and so does not have to worry about. Examples include:
 - Physical controls
 - Environmental controls
- Shared Controls
 - Controls which apply to both the infrastructure layer and customer layers, but in completely separate contexts or perspectives. In a shared control, AWS provides the requirements for the infrastructure and the customer must provide their own control implementation within their use of AWS services. Examples include:
 - Patch Management – AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications.
 - Configuration Management – AWS maintains the configuration of its infrastructure devices, but a customer is responsible for configuring their own guest operating systems, databases, and applications.
 - Awareness & Training - AWS trains AWS employees, but a customer must train their own employees.
- Customer Specific
 - Controls which are solely the responsibility of the customer based on the application they are deploying within AWS services. Examples include:
 - Customer data
 - Service and Communications Protection or Zone Security which may require a customer to route or zone data within specific security environments.

Access keys

Access keys are long-term credentials for an IAM user or the AWS account root user. You can use access keys to sign programmatic requests to the AWS CLI (Command Line Interface), SDK (Software Development Kit), and other development tools.

IAM policies don't have access keys. The only way you will ever get an Access key is to create them from an IAM user.

Access keys consist of an **access key ID** and **secret access key**, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them from the AWS Management Console. The only time that you can view or download the secret access key is when you create the keys. You cannot recover them later. However, you can create new access keys at any time.

The AWS CLI requires four pieces of information to be used:

- Access key ID
- Secret access key
- AWS Region
- Output format

Key pairs

Made of public and private keys.

Public key: Encrypt the login information for Linux and Windows EC2 instances. Held by AWS.

Private key: For windows, decrypt the login information allowing you to gain access to the login credentials and hence access to the instance. For Linux, private keys is used to remotely connect to the instance via SSH. Held by AWS users, so it is your responsibility to keep and ensure no lost or compromised.

Can use same key pair on multiple instances.

Once access is gained you can configure less privileged access controls, eg local Windows accounts

Key pairs are needed to direct connect and login into an EC2 instance and not to access AWS services.

Key pairs are not required to use AWS CLI.

Penetration Testing Procedures

AWS customers are welcome to carry out security assessments or penetration tests against their AWS infrastructure without prior approval for 8 services, listed here:

- Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
- Amazon RDS
- Amazon CloudFront
- Amazon Aurora
- Amazon API Gateways
- AWS Lambda and Lambda Edge functions
- Amazon Lightsail resources
- Amazon Elastic Beanstalk environments

Please ensure that these activities are aligned with the policy set out below. Note: Customers are not permitted to conduct any security assessments of AWS infrastructure, or the AWS services themselves. If you discover a security issue within any AWS services in the course of your security assessment, please contact AWS Security immediately.

Prohibited Activities

- DNS zone walking via Amazon Route 53 Hosted Zones
- Denial of Service (DoS), Distributed Denial of Service (DDoS), Simulated DoS, Simulated DDoS
(These are subject to the DDoS Simulation Testing policy)
- Port flooding
- Protocol flooding
- Request flooding (login request flooding, API request flooding)

Things like customized security tests require you to fill out a Simulated Events form telling AWS what it is you want to do. Be sure to include dates, accounts involved, assets involved, and contact information, including phone number and detailed description of planned events. You should expect to receive a non-automated response to your initial contact within 2 business days confirming receipt of your request.

AWS Security Bulletins

No matter how carefully engineered the services are, from time to time it may be necessary to notify customers of security and privacy events with AWS services. We will publish security bulletins online to update our customers of any changes.

[What do I do if I notice unauthorized activity in my AWS account?](#)

If you observe unauthorized activity within your AWS account, or you believe that an unauthorized party has accessed your account, then do the following:

- Change your AWS account root user password.
- Rotate and delete all root and AWS Identity and Access Management (IAM) access keys.
- Delete any potentially unauthorized IAM users, and then change the password for all other IAM users.
- Delete any resources on your account that you didn't create, such as Amazon Elastic Compute Cloud (Amazon EC2) instances and AMIs, Amazon Elastic Block Store (Amazon EBS) volumes and snapshots, and IAM users.
- Respond to the notifications that you received from AWS Support through the AWS Support Center.

AWS Secrets Manager

- AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Users and applications retrieve secrets with a call to Secrets Manager APIs, eliminating the need to hardcode sensitive information in plain text. Secrets Manager offers secret rotation with built-in integration for Amazon RDS, Amazon Redshift, and Amazon DocumentDB. Also, the service is extensible to other types of secrets, including API keys and OAuth tokens. In addition, Secrets Manager enables you to control access to secrets using fine-grained permissions and audit secret rotation centrally for resources in the AWS Cloud, third-party services, and on-premises.

AWS Shield

- a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. There are two tiers of AWS Shield - Standard and Advanced.
- All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge. AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target your web site or applications. When you use AWS Shield Standard with Amazon CloudFront and Amazon Route 53, you receive comprehensive availability protection against all known infrastructure (Layer 3 and 4) attacks.
- For higher levels of protection against attacks targeting your applications running on Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 resources, you can subscribe to AWS Shield Advanced. In addition to the network and transport layer protections that come with Standard, AWS Shield Advanced provides additional detection and mitigation against large and sophisticated DDoS attacks, near real-time visibility into attacks, and **integration with AWS WAF, a web application firewall**. AWS WAF is included with AWS Shield Advanced at no additional cost. AWS Shield Advanced also gives you 24x7 access to the AWS DDoS Response Team (DRT) and protection against DDoS related spikes in your Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 charges.
- AWS Shield Advanced is available globally on all Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 edge locations. You can protect your web applications hosted anywhere in the world by deploying Amazon CloudFront in front of your application. Your origin servers can be Amazon S3, Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), or a custom server outside of AWS. You can also enable AWS Shield Advanced directly on an Elastic IP or Elastic Load Balancing (ELB) in certain regions
 - Shield Advanced is the only tier that can protect EC2 which is not possible in Standard.

AWS Web Application Firewall (WAF)

- AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. 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 you define**. You can monitor many attributes of traffic, such as, IP addresses, URI strings, HTTP headers and HTTP methods ([more details](#)).
- You can get started quickly using Managed Rules for AWS WAF, a pre-configured set of rules managed by AWS or AWS Marketplace Sellers. The Managed Rules for WAF address issues like the OWASP Top 10 security risks. These rules are regularly updated as new issues emerge. AWS WAF includes a full-featured API that you can use to automate the creation, deployment, and maintenance of security rules.
- With AWS WAF, **you pay only for what you use. The pricing is based on how many rules you deploy and how many web requests your application receives**. There are no upfront commitments.
- **You can deploy AWS WAF on Amazon CloudFront** as part of your CDN solution, the Application Load Balancer that fronts your web servers or origin servers running on EC2, or Amazon API Gateway for your APIs.
- AWS WAF is included with AWS Shield Advanced at no additional cost.

Amazon GuardDuty

- Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3. With the cloud, the collection and aggregation of account and network activities is simplified, but it can be time consuming for security teams to continuously analyze event log data for potential threats. With GuardDuty, you now have an intelligent and cost-effective option for continuous threat detection in AWS.
- The service uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats. GuardDuty analyzes tens of billions of events across multiple AWS data sources, such as AWS CloudTrail event logs, Amazon VPC Flow Logs, and DNS logs.
- With a few clicks in the AWS Management Console, GuardDuty can be enabled with no software or hardware to deploy or maintain. By integrating with Amazon CloudWatch Events, GuardDuty alerts are actionable, easy to aggregate across multiple accounts, and straightforward to push into existing event management and workflow systems.
- Threats can include issues like escalations of privileges, uses of exposed credentials, or communication with malicious IPs, URLs, or domains. For example, GuardDuty can detect compromised EC2 instances that serve malware, unauthorized infrastructure deployments such as EC2 instances deployed in a Region that has never been used, or unusual API calls like a password policy change to reduce password strength.
- GuardDuty informs you of the status of your AWS environment by producing security findings that you can view in the GuardDuty console or through Amazon CloudWatch events.

AWS Network Firewall

AWS Network Firewall is a managed firewall service that provides filtering for both inbound and outbound network traffic. It allows you to create rules for traffic inspection and filtering, which can help protect your production VPC.

AWS Firewall Manager

AWS Firewall Manager simplifies your administration and maintenance tasks across multiple accounts and resources for a variety of protections, including AWS WAF, AWS Shield Advanced, Amazon VPC security groups, AWS Network Firewall, and Amazon Route 53 Resolver DNS Firewall. With Firewall Manager, you set up your protections just once and the service automatically applies them across your accounts and resources, even as you add new accounts and resources.

[Using AWS Firewall Manager to centrally configure AWS WAF rules provides the least administrative effort compared to the other options.]

[Firewall Manager allows centralized administration of AWS WAF rules across multiple accounts and Regions. WAF rules can be defined once in Firewall Manager and automatically applied to APIs in all the required Regions and accounts.]

AWS Security Hub

AWS Security Hub is a cloud security posture management (CSPM) service that performs security best practice checks, aggregates alerts, and enables automated remediation.

Amazon Macie

- Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover, classify and protect your sensitive data in AWS.
- Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property. It provides you with dashboards and alerts that give visibility into how this data is being accessed or moved.
- As organizations manage growing volumes of data, identifying and protecting their sensitive data at scale can become increasingly complex, expensive, and time-consuming. Amazon Macie automates the discovery of sensitive data at scale and lowers the cost of protecting your data.
- Macie automatically provides an inventory of Amazon S3 buckets including a list of unencrypted buckets, publicly accessible buckets, and buckets shared with AWS accounts outside those you have defined in AWS Organizations. Then, Macie applies machine learning and pattern matching techniques to the buckets you select to identify and alert you to sensitive data.
- Macie's alerts, or findings, can be searched and filtered in the AWS Management Console and sent to Amazon EventBridge, for easy integration with existing workflow or event

management systems, or to be used in combination with AWS services, such as AWS Step Functions to take automated remediation actions.

All this can help you meet regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) and General Data Privacy Regulation (GDPR).

Amazon Inspector

- Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings prioritized by level of severity. These findings can be reviewed directly or as part of detailed assessment reports which are available via the Amazon Inspector console or API.
- Amazon Inspector security assessments help you check for unintended network accessibility of your Amazon EC2 instances and for vulnerabilities on those EC2 instances. Amazon Inspector assessments are offered to you as pre-defined rules packages mapped to common security best practices and vulnerability definitions. Examples of built-in rules include checking for access to your EC2 instances from the internet, remote root login being enabled, or vulnerable software versions installed. These rules are regularly updated by AWS security researchers.

Compute Services

List of Compute Services

- Amazon EC2
- Amazon EC2 Auto Scaling
- Amazon Elastic Container Registry
- Amazon Elastic Container Service
- Amazon Elastic Kubernetes Service
- Amazon Lightsail
- AWS Batch
- AWS Elastic Beanstalk
- AWS Fargate
- AWS Lambda
- AWS Serverless Application Repository
- AWS Outposts
- VMware Cloud on AWS

Elastic Cloud Compute (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model. We offer the fastest processors in the cloud and we are the only cloud with 400 Gbps ethernet networking. We have the most powerful GPU instances for machine learning training and graphics workloads, as well as the lowest cost-per-inference instances in the cloud. More SAP, HPC, Machine Learning, and Windows workloads run on AWS than any other cloud.

The customer is responsible for managing, support, patching and control of the guest operating system of EC2 instances.

EC2 Instance Type characteristics:

- AES-NI - Advanced Encryption Standard – New Instructions – Shows if instance provides advanced data protection
- AVX – Advanced Vector Extensions – Used for applications focussed on audio and video, scientific calculations and 3D modelling & analysis.
- Turbo – Intel Turbo Boost & AMD Turbo Core technologies

EC2 Storage:

- Persistent – Attaching EBS volumes – Not physically attached to EC2, they are network attached storage devices. Automatically replicated to other EBS volumes in same AZ for resiliency. Can detach EBS and attach to another EC2. Encryption and snapshots to S3 possible. Different sizes and performance capabilities possible.
- Ephemeral – aka instance backed storage - Created by EC2 using local storage, physically attached to EC2. When instance stopped, all data is lost. If you reboot instance the data will remain intact. Unable to detach from instance.

EC2 Instance Families – Groups of similar EC2 instance types:

- Microinstances – Low cost, minimal CPU & memory, for low throughput. For eg low traffic websites
- General Purpose – Balanced mix CPU, memory & storage. Ideal for small to medium DBs, test and dev servers, backend servers
- Computer optimised – Greater focus on compute. High performance front end, video encoding, web servers, science & engineering applications
- GPU instances – For graphics intensive applications
- FPGA instances – Allow you to customise Field Programmable Gate Arrays to create app specific hardware accelerations when used with apps that use massively parallel processing power eg genomics and financial computing
- Memory optimised – For in memory applications eg. Real time processing of unstructured data. Lowest cost per GB from all instance families
- Storage optimised – For enhanced storage, use SSD for low latency and high IO performance, high IOPS. Great for analytic workloads, noSQL DBs, data filesystems, analogue processing applications...

[Amazon Machine Image \(AMI\)](#)

- AMIs common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. AMIs enable you to quickly and easily start new instances that have everything you need.
 - For example, if your application is a website or a web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.
- An AMI provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.
- An AMI includes the following:
 - One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
 - Launch permissions that control which AWS accounts can use the AMI to launch instances.
 - A block device mapping that specifies the volumes to attach to the instance when it's launched.
- You can use a bootstrap action to install additional software, dependencies or customize the configuration of AMI instances.

[User Data](#)

During config of EC2, there is an option to configure ‘User Data’. This section allows you to enter commands that will run on the first boot cycle of that instance. Good way to pull down additional software, get latest OS updates, etc...

[Meta Data](#)

Used to gather and query an instance that is running, such as host name, events, security groups

[Status Checks](#)

Used to check health and status of EC2 instance. Helps to troubleshoot issues. Two types:

System Status Checks - Likely issue with underlying host e.g. loss of power, hardware issues... Mostly out of our control. Stop instance and restart, instance should restart on another host. Rebooting wont do this.

Instance Status Checks – Looks at EC2 instance itself rather than underlying host. Likely will require our intervention to solve. Causes are for example corrupt file systems, incorrect network configuration, incompatible kernel, etc...

EC2 Auto Scaling

Not to be confused with AWS Auto Scaling which deals with DynamoDB, Amazon ECS and Amazon Aurora.

Scaling of your EC2 fleet when required. This includes increase (scale out) and decrease (scale in) based on metric such as CPU utilisation. Helps optimise the costs of EC2 fleet as you only pay for resources as they are running. Leads to better customer satisfaction and less administrative oversight required. Pair with an ELB for flexible and scalable architecture.

Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.

- Create a launch configuration or Launch template (newer) – these define how an auto-scaling group builds EC2 instances, eg. which AMI to use, which instance type, to use spot instances, etc...
- Create an autoscaling group – which will defines (1) the desired capacity and other limitations of the group using scaling policies, (2) where the group should scale resources such as which AZ, which subnet, etc... As an example, launch 2-10 instances depending on these metrics, send these notifications, etc...

General advice is to scale up aggressively to deal with load issues, but scale down slowly. As it can take time for instances to come online but terminating them can be done almost instantly, so to avoid stop-starting it is best to scale in slowly. Best way to do this is to have a long ‘cooldown’ on autoscaling’s instance removal policy.

Autoscaling policies can be:

- Manual scaling – setting upper and lower bounds manually – useful if traffic pattern can be predicted e.g. advertising campaigns will increase traffic. Gives greater control over scaling in and out, ideal for planned events, though it’s not a long-term solution
- Schedule scaling – setting scaling to happen at certain time of day e.g. for batch processing using Spot instances or to turn off all test/dev environments after 8pm and turn back on at 8am. Can be used with other scaling approaches.
- Dynamic scaling – automatically adds removes instances as required. Can be done by:
 - Step scaling – tracks a metric (eg CloudWatch alarms) to determine how to scale. When a ‘trigger point’ is reached an action is taken. Instances take time to get online, so to avoid adding more and more instances during this time, a ‘cooldown policy’ needs to be created. A specified number of new instances can be added at certain levels of the metric using multiple step scaling policies, e.g. CPU>60% add 1 instance, >80% add 2 instances, >95% add 3 instances. Separate step scaling policies can work together by acknowledging each other’s cooldown policy’s and thus avoid over/under-provisioning
 - Target tracking – You set a target for a metric, eg. 40% CPU usage and rest will be automated, such as alarms and scaling mechanisms. Target tracking is harder with few instances as each constitutes a large relative addition to overall instance count
- Predictive scaling – uses machine learning to understand your average loads and provisions instances based on training data. Training data can be provided by CloudWatch. You need at

least 24 hours of historical data. Can find patterns up to 14 days in the past. You can simulate it using ‘forecast only mode’ to show predictions without taking any action. If you are happy with the predictions you can switch to ‘forecast and scale mode’. It only scales instances at the start of every hour, so it does not work in real time, compared to other scaling methods. Can be used with dynamic scaling to alleviate this.

[Purchase Options](#)

On Demand Instances

On-Demand Instances let you pay for compute capacity by the hour or second (depending on the instance type) with no long-term commitments. You have full control over its lifecycle—you decide when to launch, stop, hibernate, start, reboot, or terminate it. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs.

Pricing is per instance-hour consumed for each instance, from the time an instance is launched until it is terminated or stopped. Each partial instance-hour consumed will be billed per-second for Linux Instances (minimum of 60 seconds) and as a full hour for all other instance types.

There is no long-term commitment required when you purchase On-Demand Instances. You pay only for the time that your On-Demand Instances are in the running state. The price per second or hour for a running On-Demand Instance is fixed.

We recommend that you use On-Demand Instances for applications with short-term, irregular workloads that cannot be interrupted.

For significant savings over On-Demand Instances, use AWS Savings Plans, Spot Instances, or Reserved Instances.

Reserved Instances

A Reserved Instance is a reservation of resources and capacity, for either one or three years, for a particular Availability Zone within a region. When you purchase a reservation, you commit to paying for all of the hours of the 1- or 3-year term; in exchange, the hourly rate is lowered significantly.

With RIs, you can choose the type that best fits your applications needs:

- **Standard RIs:** These provide the most significant discount (up to 72% off On-Demand) and are best suited for steady-state usage.
- **Convertible RIs:** These provide a discount (up to 54% off On-Demand) and the capability to change the attributes of the RI (instance family, operating system, and tenancy) as long as the exchange results in the creation of Reserved Instances of equal or greater value (even if this means switching RIs to a different instance family). There are no limits to how many times you perform an exchange. Like Standard RIs, Convertible RIs are best suited for steady-state usage.
- **Scheduled RIs:** These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

(<https://support.cloudability.com/hc/en-us/articles/204307758-AWS-101-Reserved-Instances>)

Reserved Instance Savings Guide

1. Standard one-year - all upfront = up to 72%
2. Standard three-years - all upfront = up to 72%
3. Standard one-year - all no upfront = 40%
4. Standard three-years - all no upfront = 60%
5. Convertible one-year - all upfront = up to 54%
6. Convertible three-years - all upfront = up to 54%
7. Convertible one-year - all no upfront = 31%
8. Convertible three-years - all no upfront = 54%

Amazon EC2 Reserved Instances (RI) provide a significant discount (up to 72%) compared to On-Demand pricing and can provide a capacity reservation when used in a specific Availability Zone. If an Availability Zone is specified, EC2 reserves capacity matching the attributes of the RI. AWS Billing automatically applies your RI's discounted rate when attributes of EC2 instance usage match attributes of an active RI. Alternatively, you can also choose to forego the capacity reservation and purchase an RI that is scoped to a region. RIs that are scoped to a region automatically apply the RI's discount to instance usage across AZs and instance sizes in a region, making it easier for you to take advantage of the RI's discounted rate. In summary, when you purchase a Reserved Instance, you determine the scope of the Reserved Instance. The scope is either regional or zonal.

- Regional: When you purchase a Reserved Instance for a Region, it's referred to as a regional Reserved Instance.
- Zonal: When you purchase a Reserved Instance for a specific Availability Zone, it's referred to as a zonal Reserved Instance.

	Regional Reserved Instances	Zonal Reserved Instances
Availability Zone flexibility	The Reserved Instance discount applies to instance usage in any Availability Zone in the specified Region.	No Availability Zone flexibility—the Reserved Instance discount applies to instance usage in the specified Availability Zone only.
Capacity reservation	A regional Reserved Instance does <i>not</i> reserve capacity.	A zonal Reserved Instance reserves capacity in the specified Availability Zone.
Instance size flexibility	The Reserved Instance discount applies to instance usage within the instance family, regardless of size. Only supported on Amazon Linux/Unix Reserved Instances with default tenancy. For more information, see Instance size flexibility determined by normalization factor.	No instance size flexibility—the Reserved Instance discount applies to instance usage for the specified instance type and size only.

On Demand Capacity Reservations

- On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or regional Reserved Instances.
- By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. You can create Capacity Reservations at any time, without entering into a one-year or three-year term commitment, and the capacity is available immediately. When you no longer need it, cancel the Capacity Reservation to stop incurring charges.
- On-Demand Capacity Reservations are priced exactly the same as their equivalent (On-Demand) instance usage. If a Capacity Reservation is fully utilized, you only pay for instance usage and nothing towards the Capacity Reservation. If a Capacity Reservation is partially utilized, you pay for the instance usage and for the unused portion of the Capacity Reservation.
- When you create a Capacity Reservation, you specify:
 - The Availability Zone in which to reserve the capacity
 - The number of instances for which to reserve capacity
 - The instance attributes, including the instance type, tenancy, and platform/OS
- Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.
- In addition, you can use Savings Plans and regional Reserved Instances with your Capacity Reservations to benefit from billing discounts. AWS automatically applies your discount when the attributes of a Capacity Reservation match the attributes of a Savings Plan or regional Reserved Instance.

Differences between On-Demand Capacity Reservations, Reserved Instances, and Savings Plans

The following table highlights key differences between Capacity Reservations, Reserved Instances, and Savings Plans:

	On-Demand Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
Term	No commitment required. Can be created and cancelled as needed.		Require fixed one-year or three-year commitment	
Capacity benefit	Capacity reserved in a specific Availability Zone.		Do not reserve capacity in an Availability Zone.	
Billing discount	No billing discount. Instances launched into a Capacity Reservation are charged at their standard On-Demand rates. However, you can use Savings Plans or regional Reserved Instances with Capacity Reservations to get a billing discount. Zonal Reserved Instances do not apply to Capacity Reservations.		Provide billing discounts	
Instance Limits	Limited to your On-Demand Instance limits per Region.	Limited to 20 per Availability Zone. A limit increase can be requested.	Limited to 20 per Region. A limit increase can be requested.	No limits.

Spot Instances

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. The Spot prices are determined by 'supply and demand' for Amazon EC2 spare capacity, the spot price is set by AWS. The price per second for a running On-Demand Instance is fixed. Spot Instances are available at up to a 90% discount compared to On-Demand prices. You can use Spot Instances for various stateless, fault-tolerant, or flexible applications such as big data, containerized workloads, CI/CD, web servers, high-performance computing (HPC), and test & development workloads. Because Spot Instances are tightly integrated with AWS services such as Auto Scaling, EMR, ECS, CloudFormation, Data Pipeline and AWS Batch, you can choose how to launch and maintain your applications running on Spot Instances.

Moreover, you can easily combine Spot Instances with On-Demand, RIs and Savings Plans Instances to further optimize workload cost with performance. Due to the operating scale of AWS, Spot Instances can offer the scale and cost savings to run hyper-scale workloads. You also have the option to hibernate, stop or terminate your Spot Instances when EC2 reclaims the capacity back with two-minutes of notice.

Tenancy: Shared Tenancy, Dedicated Instances & Dedicated Hosts

Shared Tenancy

EC2 instance is launched on any available host with the required resources. Same host can be used for multiple customers. AWS security mechanisms prevent one EC2 instance accessing another in the same host.

Dedicated Instances

Dedicated Instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer. Dedicated Instances that belong to different AWS accounts are physically isolated at a hardware level, even if those accounts are linked to a single payer account. However, Dedicated Instances may share hardware with other instances from the same AWS account that are not Dedicated Instances.

Dedicated Hosts

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts give you additional visibility and control over how instances are placed on a physical server, and you can reliably use the same physical server over time. As a result, Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses, including Windows Server, Microsoft SQL Server, SUSE, and Linux Enterprise Server; and address corporate compliance and regulatory requirements.

Host recovery

Host recovery automatically restarts your instances onto a new replacement host if failures are detected on your Dedicated Host. Host recovery reduces the need for manual intervention and lowers the operational burden if there is an unexpected Dedicated Host failure.

Additionally, built-in integration with AWS License Manager automates the tracking and management of your licenses if a host recovery occurs.

Difference between Dedicated Instances and Dedicated Hosts

- Dedicated instances and dedicated hosts are separate offerings.
 - Dedicated Instances are Amazon EC2 instances that run in a VPC on hardware that's dedicated to a single customer.
 - Your Dedicated instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. This means that no other AWS Account will run an instance on the same Host, but other instances (both dedicated and non-dedicated) from the same AWS Account might run on the same Host.
 - A dedicated instance is partitioned under a hypervisor on a shared server
 - A dedicated host is a complete physical machine with a single partition that is dedicated to a single customer.
 - Other important differences between a Dedicated Host and a Dedicated instance is that a Dedicated Host gives you additional visibility and control over how instances are placed on a physical server, you have visibility over physical cores and visibility over socket usage. Also, you can consistently deploy your instances to the same physical server over time.
 - As a result, Dedicated Hosts enable you to use your existing server-bound software licenses (from vendors such as Microsoft and Oracle) and address corporate compliance and regulatory requirements.
 - Amazon EC2 Dedicated Hosts allow you to get the flexibility and cost effectiveness of using your own licenses, but with the resiliency, simplicity and elasticity of AWS.
 - Amazon EC2 Dedicated Host is also integrated with AWS License Manager
- In some cases due to licensing restrictions some software isn't allowed to be run on a shared tenancy model. For instance if you're trying to use Bring Your Own License (BYOL) to AWS, some licenses are based on the Socket model where the number of hosts sockets are used for licensing. In other circumstances, regulatory compliance may dictate that you can't use the shared model.
- Dedicated Hosts and Dedicated Instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use.

There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, there are some differences between the two. The following table highlights some of the key differences between Dedicated Hosts and Dedicated Instances:

	Dedicated Instance	Dedicated Host
Billing	Per-instance billing	Per-host billing
Visibility of sockets, cores, and host ID	No visibility	Provides visibility of the number of sockets and physical cores
Host and instance affinity	Not supported	Allows you to consistently deploy your instances to the same physical server over time
Targeted instance placement	Not supported	Provides additional visibility and control over how instances are placed on a physical server
Automatic instance recovery	Supported	Supported
Bring Your Own License (BYOL)	Not supported	Supported

AWS Lambda

- 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. Just 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 automatically trigger from other AWS services or call it directly from any web or mobile app.
- Continuous scaling - AWS Lambda automatically scales your application by running code in response to each trigger. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.
- With AWS Lambda, you pay only for what you use. You are charged based on the **number of requests** for your functions and the **duration** (to nearest 1ms), the time it takes for your code to execute and the **compute power** provisioned for your function.
 - Lambda counts a request each time it starts executing in response to an event notification or invoke call, including test invokes from the console.
 - Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms*. The price depends on the amount of memory you allocate to your function. In the AWS Lambda resource model, you choose the amount of memory you want for your function, and are allocated proportional CPU power and other resources. An increase in memory size triggers an equivalent increase in CPU available to your function. To learn more, see the Function Configuration documentation.
 - The AWS Lambda free usage tier includes 1M free requests per month and 400,000 GB-seconds of compute time per month.
 - AWS Lambda participates in Compute Savings Plans, a flexible pricing model that offers low prices on EC2, Fargate, and Lambda usage, in exchange for a commitment to a consistent amount of usage (measured in \$/hour) for a 1 or 3 year term. With Compute Savings Plans you can save up to 17% on AWS Lambda. Savings apply to Duration, Provisioned Concurrency, and Duration (Provisioned Concurrency).
- Lambda comprises of functions – comprised of code, permissions, environment variables, power the function needs in MB.
- Can write directly into Lambda or upload a zip to S3.
- Can use a runtime language that Lambda natively supports or import custom runtime
- Functions can be invoked directly via management console, AWS SDK, AWS CLI, function URL (an endpoint for the function), AWS Toolkits, or by trigger (another AWS service or resource running function in response to an event or on a schedule, can pass in the event information). Ultimately all invocations go via the API
- Invocation types:
 - Synchronous (push based invocation). Invoke call to Lambda, response to client. If fails, no automatic retries
 - Asynchronous (event based model). Invoke call to Lambda, no response to client, unless explicitly written into logic. Handles retries if error returned, or if throttled. Uses a built-in queue. If failed can send to 'dead letter queue' or use 'Lambda Destinations' to send a feature rich record of invocation and response to another service.

- Stream (poll based model). When need to retrieve from a stream such as Amazon SQS, Kinesis Stream or DynamoDB stream. Need to create an 'Event Source Mapping' to invoke from these streams. Can be set to run function for only requests that match your specified used case. Can batch requests together to be run all at once.
- Results from Lamda can be output as API calls (e.g to Amazon SQS, DynamoDB and Amazon SNS)
- CloudWatch watches Lambda functions. Can also write custom logging in your code

Amazon LightSail

- is the easiest way to get started with AWS for developers, small businesses, students, and other users who need a solution to build and host their applications on cloud. Lightsail provides developers compute, storage, and networking capacity and capabilities to deploy and manage websites and web applications in the cloud. Lightsail includes everything you need to launch your project quickly – virtual machines, containers, databases, CDN, load balancers, DNS management etc. – for a low, predictable monthly price.
- You can get preconfigured virtual private server plans that include everything to easily deploy and manage your application. Lightsail is best suited to projects that require a few virtual private servers and users who prefer a simple management interface. Common use cases for Lightsail include running websites, web applications, blogs, e-commerce sites, simple software, and more.

Amazon Elastic Container Registry (ECR)

- is a **fully managed** container registry that makes it easy to store, manage, share, and deploy your container images and artifacts anywhere. Amazon ECR eliminates the need to operate your own container repositories or worry about scaling the underlying infrastructure. Amazon ECR hosts your images in a highly available and high-performance architecture, allowing you to reliably deploy images for your container applications.
- You can share container software privately within your organization or publicly worldwide for anyone to discover and download. For example, developers can search the ECR public gallery for an operating system image that is geo-replicated for high availability and faster downloads.
- Amazon ECR works with Amazon Elastic Kubernetes Service (EKS), **Amazon Elastic Container Service (ECS)**, and AWS Lambda, simplifying your development to production workflow, and AWS Fargate for one-click deployments. Or you can use ECR with your own containers environment. Integration with AWS Identity and Access Management (IAM) provides resource-level control of each repository. With ECR, there are no upfront fees or commitments. You pay only for the amount of data you store in your repositories and data transferred to the Internet.
- Components:
 - Registry – allows you to host and store docker images as well as create image repositories. Registry access can be controlled via IAM policies.
 - Authorisation token – To allow docker client to access your registry by authorising it as an AWS user. Use get-login command, producing an authorisation token that can be used for 12 hours.
 - Repository – Objects within registry that allow your to group together and secure different docker images. Multiple repos can be created to organise and manage docker images into different categories. You can use IAM and repo policies to assign set permissions to each repo.
 - Repository policy – Several IAM managed policies exist to help control access to ECR, such as AmazonEC2ContainerRegistryFullAccess, AmazonEC2ContainerRegistryPowerUser & AmazonEC2ContainerRegistryReadOnly. Repo polices are resource based policies, so you need to ensure you add a principal to the policy to determine who has access and what permissions they have. For an AWS user to gain access to the registry they will require access to the ecr:GetAuthorizationToken API call.
 - Image – The docker images stored within ECR. To push use docker push command, to pull use docker pull command.

Amazon Elastic Container Service (ECS)

- Lets you run Docker-enabled applications packaged as containers across a cluster of EC2 instances without requiring you to manage a complex and administratively heavy cluster management system. ‘Fargate Launch’ option which utilises AWS Fargate is used to help in cluster management. Alternatively ‘EC2 Launch’ provides much more configuration.
- ECS cluster is comprised of EC2 instances, and as such all the components of EC2 instances can be applied to ECS clusters, e.g. security groups, Elastic Load Balancers, Auto-Scaling, etc... Can connect to individual EC2 instances if required.
- Cluster acts as a resource pool, aggregating CPU, memory, etc... It is dynamically scalable. Multiple instance types can be used if required. Can span multi-AZs, but only can scale within a single region.
- Each instance will have a docker deamon and an ECS agent installed, allowing ECS commands to be translated into docker commands

Amazon Elastic Kubernetes Service (EKS)

- a fully managed Kubernetes (an open-source container-orchestration system for automating computer application deployment) service. Can grow from 10s to 1,000,000s of containers. Its container-runtime agnostic so you can use it to run Docker and rocket containers.
- AWS provides a managed service allowing you to run EKS without having to take care of provisioning and running the Kubernetes management infrastructure in what is referred to as the control plane. AWS uses multiple-AZs for additional resilience. You only need to provision and maintain the worker nodes.
- You need to configure and create an IAM service role to allow EKS to provision and configure specific resources. You also need to use CloudFormation to run a stack to create an EKS Cluster VPC. You need to install kubectl and AWS-IAM-Authenticator. Further steps are required to configure these components.
- Control plane – comprise of components such as APIs, kubelet process, and Kubernetes master, these determine how Kubernetes and your clusters communicate with each other. Control plane is run across master nodes. Control plane schedules containers onto nodes, depending on their compute requirements. Control plane tracks state of all Kubernetes objects.
- Kubernetes clusters are comprised of nodes. A node is a worker machine in Kubernetes. It runs as an on demand EC2 instance and includes software to run containers. For each node created, a specific AMI is used. Docker, kubelet and AMI authenticator are installed for security controls. Nodes communicate with EKS using an endpoint.
- You can choose to run your EKS 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.
- EKS is deeply integrated with services such as Amazon CloudWatch, Auto Scaling Groups, AWS Identity and Access Management (IAM), and Amazon Virtual Private Cloud (VPC), providing you a seamless experience to monitor, scale, and load-balance your applications.
- EKS integrates with AWS App Mesh and provides a Kubernetes native experience to consume service mesh features and bring rich observability, traffic controls and security features to applications
- EKS provides a scalable and highly-available control plane that runs across multiple availability zones to eliminate a single point of failure.
- Containers are used in PaaS where customer is responsible for app and data and the rest is taken care of by the cloud provider.

AWS Elastic Beanstalk (EBS)

- AWS Elastic Beanstalk is an AWS managed service that takes your uploaded web application code and automatically provisions and deploys the required resources within AWS to make the web application operational. It can be used with applications using Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. Various AWS resources can be utilised such as, EC2, Autoscaling, Application health monitoring, Elastic Load Balancing and Capacity provisioning.
- Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. Within minutes, your application will be ready to use without any infrastructure or resource configuration work on your part. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.
- Elastic Beanstalk provisions and operates the infrastructure and manages the application stack (platform) for you, so you don't have to spend the time or develop the expertise. It will also keep the underlying platform running your application up-to-date with the latest patches and updates. Instead, you can focus on writing code rather than spending time managing and configuring servers, databases, load balancers, firewalls, and networks.
- There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.
- Cannot be used in on-premises situations, can only be used for AWS contexts. For on-premises situations AWS OpsWorks and AWS CodeDeploy are suitable.
- You can configure event notifications for your Elastic Beanstalk environment so that notable events can be automatically published to an SNS topic, then pushed to topic subscribers. As an example, you may use this event-driven architecture to coordinate your continuous integration pipeline (such as Jenkins CI). That way, whenever an environment is created, Elastic Beanstalk publishes this event to an SNS topic, which triggers a subscribing Lambda function, which then kicks off a CI job against your newly created Elastic Beanstalk environment.
- Core components:
 - Application version – a very specific reference to a section of deployable code
 - Environment – an application version that has been deployed on AWS resources, comprises of all the resources EBS has deployed. The application has been deployed as a solution and becomes operational within your environment.
 - Environment configuration – Collection of parameters and settings that dictate how an environment will have its resources provisioned by EBS and how these will behave
 - Environment Tier – How EBS will provision resources based on what the application is designed to do e.g. if the app manages HTML requests it will be run in a web server environment, if it pulls data from an SQS queue then it may be run in a worker environment. Can only be deployed on one tier.
 - Configuration template – template that provides the baseline for creating a new, unique environment configuration
 - Platform – culmination of components in which you can build your application using EBS. These comprise of the OS of the instance, the programming language, the server type (web or application), components of EBS itself.
 - Application – Collection of different elements such as environment, environment configurations and application versions.

- First you need to create the application, you then upload the application version to EBS along with configuration info regarding the application itself, this creates the environment configuration. Environment is created by EBS with appropriate resources to run your code. You simple manage the environment, such as deploying new versions of your application. If a new application version changes the environment requirements, EBS will take care of this.

Amazon EBS fast snapshot restore (FSR) enables you to create a volume from a snapshot that is fully initialized at creation. This eliminates the latency of I/O operations on a block when it is accessed for the first time. Volumes that are created using fast snapshot restore instantly deliver all of their provisioned performance.

AWS Batch

- Used to manage and run batch computing workloads within AWS. Batch computing is used in specialist use cases which require vast amount of compute power across a cluster of compute resources to complete batch processing executing a series of tasks. AWS batch removes constraints, administration activities and maintenance tasks. Great for running multiple jobs in parallel
- Components:
 - Jobs – The unit of work to be run by AWS Batch. Can be an .exe file, an application within an ECS cluster, or a shell script. Run on EC2 instances as containerised application. Can be in different states e.g. submitted, pending, running, failed, etc...
 - Job definitions – Define specific parameters for the jobs themselves and dictate how the job will run and with what config, e.g. how many vCPUs to use, which data volumes to use, which IAM role to use, etc...
 - Job queues – Jobs that are scheduled are placed into a job queue until run. You can have multiple queues for different priorities, e.g. on-demand instance queue and spot instance queue. AWS Batch can bid on your behalf for spot instances.
 - Job scheduling – Takes control of what job should be run and from which compute environment. It will operate on a First-in-first-out basis. Ensures higher priority queues will be run first. Ensures all dependencies for jobs have been met before scheduling.
 - Compute environments – Contain the compute resources to carry out the job. Can be managed (service will handle provisioning, scaling and termination of compute instances – environment created as an ECS cluster) or unmanaged (provisioned and managed by you, gives greater customisation, tho requires greater administration, you must create the ECS cluster).

AWS Fargate

- AWS Fargate is a serverless compute engine for containers that 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. 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. Fargate allocates the right amount of compute power, eliminating the need to choose instances and scale cluster capacity. You only pay for the resources required to run your containers, so there is no over-provisioning and paying for additional servers.
- Fargate runs each task or pod in its own kernel providing the tasks and pods their own isolated compute environment. This enables your application to have workload isolation and improved security by design.
- Containers are used in PaaS where customer is responsible for app and data and the rest is taken care of by the cloud provider.

AWS Outposts

AWS Outposts is a **fully managed** service that extends AWS infrastructure, AWS services, APIs, and tools to virtually any datacenter, co-location space, or on-premises facility for a consistent hybrid experience. AWS Outposts is good for workloads that require low latency access to on-premises systems, local data processing, or local data storage. Can order through AWS Management Console. Can be used for EC2, EBS, RDS, ECS, EKS, Sagemaker and EMR.

Storage Services

Data Storage Categorisation

Data storage categorization

Block Storage

- Data is stored in chunks known as blocks.
- Blocks are stored on a volume and attached to a single instance.
- Provide very low latency data access.
- Comparable to DAS storage used on premises

File Storage

- Data is stored as separate files with a series of directories.
- Data is stored within a file system.
- Shared access is provided for multiple users.
- Comparable to NAS storage used on premises

Object Storage

- Objects are stored across a flat address space
- Objects are referenced by a unique key.
- Each object can also have associated metadata to help categorize and identify the object.

List of Storage Services

- Amazon Simple Storage Service (S3) / Amazon S3 Glacier
- Amazon Elastic Block Store (EBS)
- Amazon Elastic File System (EFS)
- AWS Storage Gateway
- AWS Snow Family (Snowcone, Snowball & Snowmobile)
- Amazon FSx for Lustre
- Amazon FSx for Windows File Server

EC2 Instance Store

Some Amazon Elastic Compute Cloud (Amazon EC2) instance types come with a form of directly attached, **block-device storage** known as the instance store. The instance store is ideal for temporary ‘ephemeral’ storage, because the data stored in instance store volumes is not persistent through instance stops, terminations, or hardware failures. Data will persist through reboots.

For data you want to retain longer, or if you want to encrypt the data, use Amazon Elastic Block Store (Amazon EBS) volumes instead.

No additional storage cost, comes with EC2 instance, tho not all EC2 instances come with instance stores. Very high I/O speed. Use same security mechanisms as EC2 itself.

Often used with load balancing group, where data is replicated and pooled between the fleet.

Amazon Elastic Block Store (EBS)

Amazon Elastic Block Store (EBS) is an easy to use, high-performance, block-storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS. An EBS volume can only be attached to a single EC2 instance at any time (except if you use EBS multi-attach), however multiple EBS volumes can be attached to a single EC2. EC2 and EBS must be in the same AZ.

You can choose from different volume types to balance optimal price and performance. You can achieve single-digit-millisecond latency for high-performance database workloads or gigabyte per second throughput for large, sequential workloads. You can change volume types, tune performance, or increase volume size without disrupting your critical applications, so you have cost-effective storage when you need it.

Designed for mission-critical systems, EBS volumes are replicated within an Availability Zone (AZ) and can easily scale to petabytes of data. They are only available in a single AZ, so if AZ fails EBS will be lost. Also, you can use EBS Snapshots with automated lifecycle policies to back up your volumes in Amazon S3, while ensuring geographic protection of your data and business continuity.

EBS volumes preserve their data even when their EC2 instance stops and terminates. They can be removed from one instance and reattached to another, and support full-volume encryption. Best practice for performance on DBs is EBS, as instance store is ephemeral.

EBS allows backups called snapshots, can be done manually or use CloudWatch events to do it automatically on a schedule. Snapshots stored on S3. Snapshots are incremental, only copying the data that has changed since previous snapshot. Can create new EBS volume from a snapshot. Can copy snapshots between regions.

Encryption by AES-256 algorithm available on selected instance types, via KMS. Volumes made of encrypted snapshots will also be encrypted.

Can resize EBS volumes when required.

Different types of volumes: Throughput optimised, IOPS optimised, cold HDD etc. Each has pros & cons

Two storage types:

- SSD: For smaller blocks, eg. DBs, boot volumes for EC2s
- HDD: For larger blocks & higher throughput rate, eg. Processing big data and logging info

EBS Multi-Attach

Let's EBS volume be accessed by many EC2 instances. Dependent on EBS and EC2 instance types: EBS Provisioned IOPS SSD io1 & io2. IOPS of the EBS volume shared between connecting EC2 instances.

Need to use EC2 nitro-based (the underlying virtualisation platform) instances. For linux up to 16 Nitro instances can be attached. Windows doesn't recognise it's a shared volume so can lead to data inconsistencies, so Linux is advised. File systems to be used are clustered file system like GFS2.

'Delete on termination' property dependent on which EBS volume is last to be terminated.

Data Lifecycle Manager

You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs. When you automate snapshot and AMI management, it helps you to:

- Protect valuable data by enforcing a regular backup schedule.
- Create standardized AMIs that can be refreshed at regular intervals.
- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.
- Create disaster recovery backup policies that back up data to isolated accounts.

Amazon Elastic File System (EFS)

- EFS is regional, any application deployment spanning across multiple AZs can all access the same file system. provides a simple, scalable, **fully managed** elastic NFS file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.
- Amazon EFS is designed to provide the throughput, IOPS, and low latency needed for **Linux workloads**. Throughput and IOPS scale as a file system grows and can burst to higher throughput levels for short periods of time to support the unpredictable performance needs of file workloads. For the most demanding workloads, Amazon EFS can support performance over 10 GB/sec and up to 500,000 IOPS. Can scale to petabytes.
- With Elastic File System (EFS), you can share data between multiple EC2 instances and your data is replicated between multiple
- EFS file system can be used by multiple EC2 instances from different data centers in parallel. Additionally, the data of the EFS file system is replicated among multiple data centers/availability zones (AZs)
- Can be used with EC2 instances, accessed via '**Mount Points**'. Can work with 10s, 100s or 1000s of EC2 instances concurrently. Use Linux NFS or EFS Mount Helper. Must be in a VPC with security group to allow access to the EC2 instance.
- Uses APIs, so any app that uses APIs can work with it.
- Uses NFS 4.1 and 4.0
- Standard and Infrequent Access. IA is cheaper but has higher first byte latency. The costs between the two are managed differently, as IA has read/write costs.
- EFS Lifecycle management exists to automatically move between the two classes, generally files not read or written to for a set number of days e.g. 14, 30, 60, 90 day are move to IA, but when they are read they are moved back to standard and the timer is reset. The only exception is files less than 128kb in size and any metadata of files which always remain in standard storage.
- EFS has 2 performance modes:
 - General purpose: Max 7k IOPS, low latency, standard throughput
 - Max I/O: >7k IOPS, virtually unlimited throughput, higher latency
- EFS has 2 throughput modes:
 - Bursting throughput: when you want throughput that scales with the amount of storage in your file system, depending on size of file system you will be allocated an expected allowance of time above your baseline performance, credited to your account as burst credits. You accumulate burst credits during normal operations times when you operate at or below baseline throughput. If, after using Bursting Throughput mode, you find that your application is throughput-constrained (for example, it uses more than 80% of the permitted throughput or you have used all of your burst credits), you may want to switch to another class.
 - Provisioned throughput: allows you to burst above the allocated burst allowance, however with charge

EFS is not supported on Windows instance. Instead use Amazon FSx for Windows File Server provides fully managed Microsoft Windows file servers, backed by a fully native Windows file system.

EFS Security

- Must have 'Allow' security for the following actions:
 - elasticfilesystem>CreateFileSystem
 - elasticfilesystem>CreateMountTarget
 - ec2:DescribeSubnet
 - ec2>CreateNetworkInterface
 - ec2:DescribeNetworkInterfaces
- Encryption at rest (using KMS to manage keys) and encryption in transit supported (using TLS protocol)

Amazon S3 Basics (Simple Storage Service)

- S3 is a regional service, you must specify the region. Available on every region however it is not truly global because while you can replicate your buckets/objects across regions for reliability & disaster recovery purposes, by default S3 objects sit only in one region though they are stored on multiple devices across multiple Availability Zones.
- S3 provides developers and IT teams with secure, durable, highly-scalable binary object storage.
- It has a simple, easy to use, web services interface to store and retrieve any amount of data from anywhere on the web.
- S3 is a safe Object-based storage for e.g. picture, text files, videos NOT databases, application or OS.
- Size of files can be from 0 – 5TB.
- Unlimited storage paid by the GB.
- Stored in buckets (folders in the cloud).
- When you upload a file to s3 you'll get a HTTP 200 code to show successful upload
- Is A Simple Key-value object store
 - Key – name of object
 - Value – data of file (sequence of bytes)
 - Version ID (important for versioning)
 - Metadata – data about what you're storing
- It's essentially a type of NoSQL database. Each bucket is a new "database", with keys being your "folder path" and values being the binary objects (files). It's presented like a file system and people tend to use it like one. Underneath, however, its not a file system at all and lacks many of the common traits of a file system.
- Security
 - Access Control Lists (ACLs) – (file level)
 - are used to define which AWS accounts or groups are granted access and the type of access. When a request is received against an S3 resource, the corresponding resource ACL is checked to verify that the requester has the necessary access permissions.
 - Bucket Policies - (bucket level)
 - is a resource-based AWS Identity andAccess Management (IAM) policy. You add a bucket policy to a bucket to grant other AWS accounts or IAM users access permissions for the bucket and the objects in it.
- have universal namespace e.g. url that can be accessed. Buckets need to be unique globally. For example, bucket url: s3-[region].amazonaws.com/[bucketName]
- Built for 99.5-99.99% availability depending on storage class. Based on uptime data is available.
- Amazon guarantee 11 x 9s durability for S3 information. Based on data loss through corruption or other data destruction methods
- Tiered Storage Available
- Encryption (encrypt your files at rest)
- S3 charged:
 - All according to Storage Tier Pricing
 - For Storage per GB
 - Per # of Requests
 - You pay for requests made against your S3 buckets and objects. You pay for all bandwidth into and out of Amazon S3, except for the following:
 - Data transferred in from the internet.
 - Data transferred out to an Amazon Elastic Compute Cloud (Amazon EC2) instance, when the instance is in the same AWS Region as the S3 bucket (including to a different account in the same AWS region).
 - Data transferred out to Amazon CloudFront (CloudFront).

- For Data Transfer (transferring from one region to another)
- For Transfer Acceleration - Enables fast, easy and secure transfers of files over long distances between your end users and an S3 bucket.
- For taking advantage of Cloudfront's globally distributed edge locations.
- For Cross Region Replication (CRR)
- Bucket names share a common name space. Their names must be unique.
- You view the buckets globally but you can have buckets in individual regions
- Contents uploaded to buckets are private by default
- Up to 100 buckets per account, tho this is soft limit and AWS can lift it
- Folders can be created in buckets to help organise, but S3 is not a file system
- S3 Bucket Life cycle rules can be used to move data from one class of S3 storage to another, or remove it from S3 entirely, e.g. Move to cheaper class after timeframe

(<http://kayleigholiver.com/aws-cloud-practitioner-s3/>)

[What is S3 Replication?](#)

Replication enables automatic, asynchronous copying of objects across Amazon S3 buckets. Buckets that are configured for object replication can be owned by the same AWS account or by different accounts. There are two kinds of S3 replication:

- Cross Region Replication (CRR). When an item has been uploaded to a primary bucket is replicated to a secondary bucket in a different AWS Region.
- Same-Region replication (SRR) is used to copy objects across Amazon S3 buckets in the same AWS Region.

Requirements:

- Both source and destination buckets must have versioning enabled.
- The source bucket owner must have the source and destination AWS Regions enabled for their account. The destination bucket owner must have the destination Region-enabled for their account. For more information about enabling or disabling an AWS Region, see AWS Service Endpoints in the AWS General Reference.
- If the source bucket has S3 Object Lock enabled, the destination bucket must also have S3 Object Lock enabled
- Amazon S3 must have permissions to replicate objects from the source bucket to the destination bucket on your behalf.
- If the owner of the source bucket doesn't own the object in the bucket, the object owner must grant the bucket owner READ and READ_ACP permissions with the object access control list (ACL)

[S3 Tiers](#)

S3 Standard

- Built for 99.99% availability
- Amazon guarantee 11 x 9s durability for S3 information
- Stored redundantly across multiple devices in multiple facilities
- Designed to sustain the loss of 2 facilities concurrently

S3 – Intelligent Tiering

- Amazon guarantee 11 x 9s durability for S3 information
- Uses ML looking at your usage patterns
- Has three tiers: frequent access, infrequent access and archival. Moves data to the most cost-effective access tiers without performance impact or operational overhead.
- Monthly object monitoring and automation charge
- ~~Available from only one AZ~~ Redundantly store objects on multiple devices across a minimum of three Availability Zones in an AWS Region

S3 – IA (Infrequently Accessed)

- Amazon guarantee 11 x 9s durability for S3 information
- For data accessed less frequently, but requires rapid access.
- Lower fee than S3, but you are charged a retrieval fee

S3 One Zone – IA

- Amazon guarantee 11 x 9s durability for S3 information
- Lower cost option for infrequently accessed data
- Only available in one availability zone

S3 Glacier

- Uses vaults and archives rather than buckets
- No GUI, move data into vaults using APIs and SDKs. Or S3 life cycle rules.
- Retrieving data will require API, SDK or CLI. 3 retrieval options:
 - Expedited: Urgent requirement, request must be <250mb. Takes 1-5mins. Most expensive.
 - Standard: Any size. 3-5 hours. 2nd most expensive.
 - Bulk: Retrieve petabytes. 5-12 hours to complete. Cheapest of options.
- Amazon guarantee 11 x 9s durability for S3 information
- Low cost storage
- Used for archival only
- Comes in the models: Expedited, Standard or Bulk.
- Standard retrieval configurable from minutes to hours

S3 Glacier Deep Archive

- Just like S3 glacier
- Lowest cost of storage class
- Retrieval time of 12 hours or less, is only retrieval option

S3 Glacier Instant Retrieval

- is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds.
- With S3 Glacier Instant Retrieval, you can save up to 68% on storage costs compared to using the S3 Standard-Infrequent Access (S3 Standard-IA) storage class, when your data is accessed once per quarter.
- S3 Glacier Instant Retrieval delivers the fastest access to archive storage, with the same throughput and milliseconds access as the S3 Standard and S3 Standard-IA storage classes. S3 Glacier Instant Retrieval is designed for 99.99999999% (11 nines) of data durability and 99.9% availability by redundantly storing data across multiple physically separated AWS Availability Zones.
- It is designed for rarely accessed data that still needs immediate access in performance-sensitive use cases like image hosting, online file-sharing applications, medical imaging and health records, news media assets, and genomics.

(<http://kayleigholiver.com/aws-cloud-practitioner-s3/>)

[S3 Versioning](#)

- Allows multiple versions of same object to exist, allows recovery of earlier version, should files be accidentally or on-purposely deleted. Not enabled by default, once enabled it can't be disabled only suspended, keeps existing versioned files but won't create any new versioned files.
- Storage costs can balloon when versioning exists
- Deleting versioned objects, only puts delete marker on the latest version. All versions are still accessible tho just harder to access via for example API calls. Must use delete version commands in SDK to permanently delete objects and all versions.

[Server Access Logging](#)

- S3 will capture requests made to bucket and its objects. Important for security, analysis and can be required for regulatory purpose. Not guaranteed every access attempt is logged, works on a best-effort basis by S3. Logs collated and sent every few hours by S3.
- Access logging must be enabled manually, by default it is disabled.
- Need to specify a target bucket to store logs, preferred source and target buckets are not the same bucket. Must be in same region. Will need to enable permissions 'log delivery group'. Need to enable 'S3 access log group' via ACL to read logs.
- If encryption enabled on target bucket, KMS encryption is not supported so you must use SSE-S3
- prefixYYYY-MM-DD-HH-MM-SS-uniqueString
- Log will contain – Bucket owner canonical User ID, bucket name, timestamp, remote IP address of requestor, requestor (IAM or unauthenticated users), request ID random string, operation of request, key, request URI, HTTP Status, error code, bytes sent, object size, total time (in ms), turnaround time, referrer from HTTP header, user agent from HTTP header, version ID requested, host ID, signature version, cypher suite, authentication type, host header, TLS version
- Hyphen indicates empty data

Object Level Logging

CloudTrail logging that performs logging against certain S3 data events, such as API events GetObject, DeleteObject and PutObject. Stores these events in a log file stored in S3. Stores identity of API caller, timestamp and source IP address.

Can be configured for all S3 buckets within the AWS CloudTrail console.

Alternatively, it can be configured at bucket level using ‘Object level Logging’ property.

Object Lock

Used to meet WORM (write once read many) compliance required by bodies such as FINRA, gives objects in buckets protection from being deleted, either for a set period of time, or simply forever.

Setting object lock for a bucket can only be done at bucket creation.

Versioning needs to be enabled also.

2 retention modes:

- Governance mode: Prevents deletions and overwrites of any versions in the bucket throughout the duration. Can be bypassed by having very specific set of permissions.
- Compliance mode: Like governance mode, except no bypass allowed, even by AWS root account.

Can be done on a per-object basis also.

Legal hold will only apply to versions and not at bucket level. Prevents objects from being deleted. Do not have an expiration date.

Can have object locking and legal hold simultaneously.

Tags

See cost allocation tags

Can also be used to store non-cost-based information, such as project, environment, etc...

Transfer Acceleration

Utilises Amazon CloudFront, to deliver data to S3 and retrieve it later. Request will go via a CloudFront edge location, it will then be routed via a highspeed optimised network path back to S3. This will incur additional costs. Bucket name must be DNS compliant, containing no periods. Making sure to use the Transfer Acceleration specific endpoint. There are a small number of calls not possible with transfer acceleration.

Requestor Pays

Requestor will take on cost of requests and data transfer, rather than the bucket owner. The bucket owner will still pay for storage costs of the bucket. All access to bucket must be authenticated as anonymous requests will not be able to comply with requestor pays attribute.

Cross Origin Resource Sharing (CORS)

Allows specific resources on a webpage to be requested from a different domain than its own. Allows building of client-side web applications and then letting them access S3 resources. Determined by the CORS configuration of each bucket. The first policy which matches all criteria is used to process the requests. Policies can contain multiple rules to relate to multiple origin website for example.

Creating a Website on S3

- Select static website hosting property. Must select index document, error document. Redirection rules are optional. Can redirect your bucket endpoint to any specified URL.
- You can use bucket policies to make entire S3 buckets public (instead of individually updating the permissions on each object within the bucket) by enabling “**Edit public access settings**” to make everything in a bucket public by default. Must add a bucket policy to allow public to read contents of bucket.
- You can use S3 to host only a **static** website e.g. .html. websites that require a database connection e.g. a WordPress site cannot be hosted on S3.
- S3 scales automatically to meet your demand. Useful when there will be a large number of requests e.g. movie previews.
- Recognise the url that a statically hosted website will use i.e. <http://sitename-website2019.s3-website-us-east-1.amazonaws.com>
- Does not support HTTPS. Does not support requestor pays

(<http://kayleigholiver.com/aws-cloud-practitioner-s3/>)

How can you create an Amazon S3 bucket that cannot have any public objects due to compliance requirements?

Enable the ‘Block public access’ option. This can be done in S3 Console, the CLI, the S3 APIs or from within CloudFormation templates. Additionally you have the option to enable/disable the following options:

- Block public access to buckets and objects granted through new access control lists (ACLs)
- Block public access to buckets and objects granted through any access control lists (ACLs)
- Block public access to buckets and objects granted through new public bucket policies
- Block public and cross-account access to buckets and objects through any public bucket policies

Field-level encryption allows you to enable your users to securely upload sensitive information to your web servers. The sensitive information provided by your users is encrypted at the edge, close to the user, and remains encrypted throughout your entire application stack. This encryption ensures that only applications that need the data—and have the credentials to decrypt it—are able to do so.

By deploying an S3 VPC gateway endpoint, the application can access the S3 buckets over a private network connection within the VPC, eliminating the need for data transfer over the internet. This can help reduce data transfer fees as well as improve the performance of the application. The endpoint policy can be used to specify which S3 buckets the application has access to.

S3 MFA Delete requires additional authentication to permanently delete an object version. This prevents accidental deletion.

You can use presigned URLs to grant time-limited access to objects in Amazon S3 without updating your bucket policy. A presigned URL can be entered in a browser or used by a program to download

an object. The credentials used by the presigned URL are those of the AWS user who generated the URL.

You can also use presigned URLs to allow someone to upload a specific object to your Amazon S3 bucket. This allows an upload without requiring another party to have AWS security credentials or permissions. If an object with the same key already exists in the bucket as specified in the presigned URL, Amazon S3 replaces the existing object with the uploaded object.

You can use the presigned URL multiple times, up to the expiration date and time.

S3 event notification can only send notifications to SQS, SNS and Lambda, BUT not Sagemaker

CloudFront's Origin Access Identity (OAI) is a special CloudFront user that you can associate with your distribution. By applying an OAI to your S3 bucket, you're able to securely lock down all direct access to your S3 files and require all requests to come through CloudFront. Amazon Web Application Firewall (WAF) is a security feature that helps protect your resources against common exploits. You can configure AWS WAF directly on your CloudFront distribution to inspect incoming requests to your web application.

If your origin is an Amazon S3 bucket configured as a website endpoint, you must set it up with CloudFront as a custom origin. That means you can't use OAC (or OAI). However, you can restrict access to a custom origin by setting up custom headers and configuring the origin to require them. For more information, see Restricting access to files on custom origins.

S3 Policies to Control Access

[Identity-based policies](#)

Attached to the IAM identity requesting access, using IAM permissions policies. Can be associated with user, role or group. Resource is defined in the policy eg bucket name. Can use conditions to refine access further.

[Resource-based policies](#)

Associated with the resource. Come in form of Access Control Lists (ACLs) and bucket policies. Need to define who will be allowed or denied access.

S3 Bucket Policies

(Written in JSON) contain five key elements. Effect, Action, Resource, Condition and Principal:

- Effect – Use Allow or Deny to indicate whether the policy allows or denies access.
- Action – Include a list of actions that the policy allows or denies.
- Resource (Required in only some circumstances) – If you create an IAM permissions policy, you must specify a list of resources to which the actions apply. If you create a resource-based policy, this element is optional. If you do not include this element, then the resource to which the action applies is the resource to which the policy is attached.
- Condition (Optional) – Specify the circumstances under which the policy grants permission.
- Principal is used by Resource Policies (SNS, S3 Buckets, SQS, etc) to define who the policy applies to. In most cases the Principal is the root user of a specific AWS account. That AWS account can then delegate permission (via IAM) to users or roles. That means when you trust the root of another AWS Account, you're trusting all the IAM or federated users in that account.

[Access Control Lists](#)

Allows control access of a bucket and specific objects within buckets by AWS accounts.

Much less granular than S3 bucket policies.

Cannot deny access via ACLs or implement conditional elements.

Signed (authenticated) and unsigned (unauthenticated) requests are possible for public access ACLs.

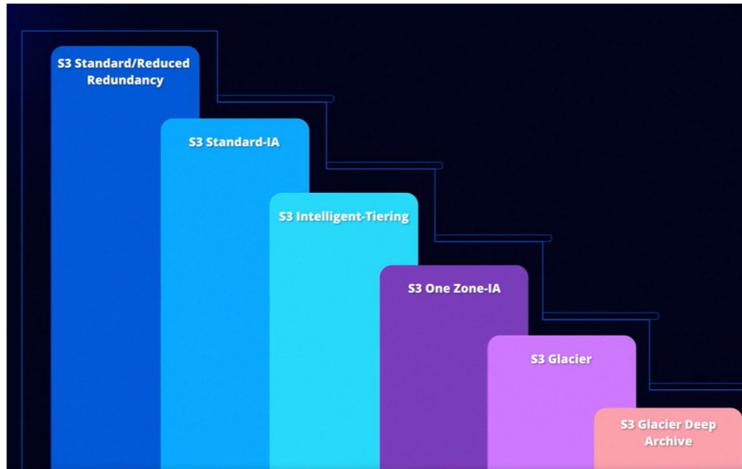
[Identity-based or Resource-based Policies, which to use?](#)

IAM policies are easier to manage if large numbers required i.e. 1 or 2 policies can cover many buckets easily, rather than a policy per each bucket. Bucket policies may be more suitable if you want to work exclusively within S3 and the scale is smaller.

Both can be used simultaneously.

If multiple policy types are used, or even multiple instances of the same policy type: the principal must have allow access granted with zero denies present. A single deny will take precedent and prevent access.

S3 Lifecycle Configuration



Can only transition objects 'down the staircase'.

You are charged for transitioning objects to different storage classes, the fee increases as you move 'down the staircase', it is charged on a per object basis. Therefore it is best to amass smaller objects into larger objects before transitioning.

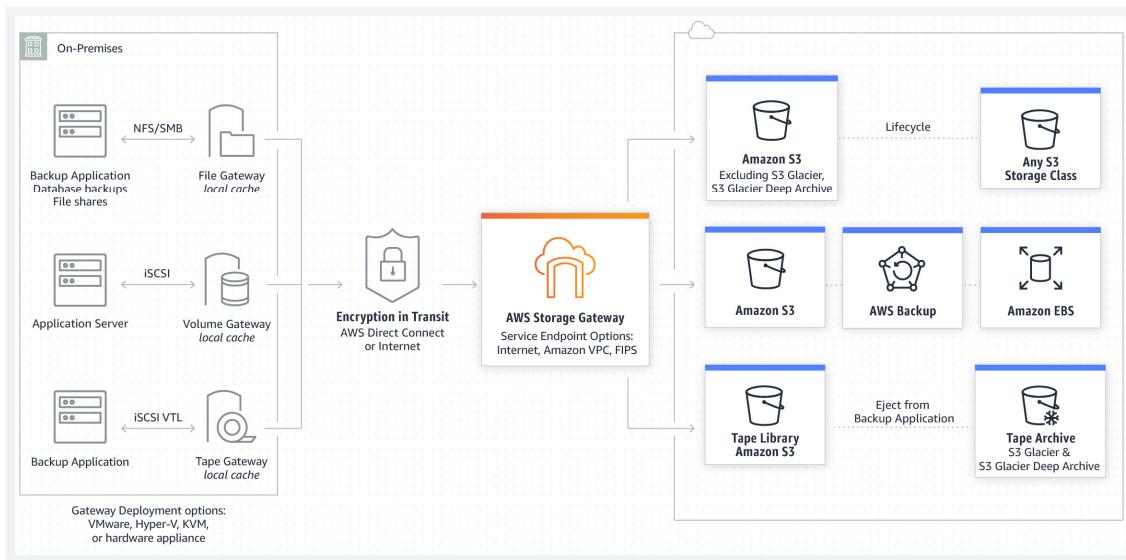
Most storage classes have a minimum storage duration before you can delete, overwrite or transition those objects. These durations increase as you go 'down the staircase'. If you breach these durations and delete, overwrite or transition early you are charged.

XML file. Comprises of a set of rules, each rule comprises of:

- ID – Uniquely ID's the rule
- Filter – Defines which objects in the bucket to take action on, can choose all objects or a subset. Can filter on prefix, object size or object tag; or a combination of these.
- Status – Can enable or disable each rule individually
- Actions – Where you want the objects to move to or are you deleting the objects.
 - Transition – Move objects between storage classes – can use object age to filter. Only works on the current version of the object.
 - Expiration – Delete objects – can use object age to filter. Only works on the current version of the object.
 - Non-current version transition – Same as transition, but works on non-current versions of the object. Can filter based on the number of days the version has been non-current and the maximum number of versions to retain.
 - Non-current version expiration – Same as expiration, but works on non-current versions of the object. Can filter based on the number of days the version has been non-current and the maximum number of versions to retain.
 - Expired object delete marker – To process objects with 0 versions and a delete marker.
 - Abort Incomplete Multipart Upload – To clean up incomplete multipart uploads. Can define how long these can uploads can remain in progress before they are processed.

AWS Storage Gateway

- is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. It provides a link between your on premises storage e.g SAN, NAS and DAS, with Amazon S3 and Amazon Glacier. Can be installed as hardware or software appliance within existing datacentre.
- Customers use Storage Gateway to simplify storage management and reduce costs for key hybrid cloud storage use cases. These include moving backups to the cloud, using on-premises file shares backed by cloud storage, and providing low latency access to data in AWS for on-premises applications, as well as various migration, backup, archiving, processing, moving data to S3 for in-cloud workloads and tiered storage; and disaster recovery use cases.
- It seamlessly integrates on-premises enterprise applications and workflows with Amazon's block and object cloud storage services through industry standard file-storage protocols.
- It provides low-latency performance by caching frequently accessed data on premises, while storing data securely and durably in Amazon cloud storage services. It provides an optimized data transfer mechanism and bandwidth management, which tolerates unreliable networks and minimizes the amount of data being transferred.
- It brings the security, manageability, durability, and scalability of AWS to existing enterprise environments through native integration with AWS encryption, identity management, monitoring, and storage services.
- First 100GB free. Max \$125 per gateway/month
- 3 options:
 - File Gateway: Securely store files as objects within S3, mount/map drives to an S3 bucket as if it was shared locally. On premises cache for most often accessed files to reduce latency and reduce egress traffic costs.
 - Volume Gateway:
 - Stored volume gateways. Backup local storage to S3 as EBS snapshots. Primary data library is on premises for low latency.
 - Cached volume gateways. Primary data storage is on S3. Local volumes act as buffer and cache for low latency.
 - Tape Gateway (Virtual Tape Library) – Backup data to S3 and leverage S3 Glacier & S3 Glacier Deep Archive storage classes at far lower cost than S3.



AWS Snow Family

Used to transfer massive data quantities both into AWS and out of AWS. Device sent to customer with computer and storage capabilities, data is transferred onto it, then returned to AWS for upload onto AWS infrastructure. Anti-tamper casing. Encryption using KMS. Ruggedised enclosure, operates in extreme conditions. Tracked with e-ink digital labels, checkable via management console, SNS, etc. Data is deleted once transfer completed.

AWS Snowcone

- smallest member of the AWS Snow Family of edge computing, edge storage, and data transfer devices, weighing in at 4.5 pounds (2.1 kg) with 8 terabytes of usable storage. Up to 10Gbits transfer speed.
- Snowcone is ruggedized, secure, and purpose-built for use outside of a traditional data center. Its small form factor makes it a perfect fit for tight spaces or where portability is a necessity. You can use Snowcone in backpacks on first responders, or for IoT, vehicular, and even drone use cases.
- You can execute compute applications at the edge, and you can ship the device with data to AWS for offline data transfer, or you can transfer data online with AWS DataSync from edge locations.
- Snowcone has multiple layers of security and encryption. You can use either of these services to run edge computing workloads that use AWS IoT Greengrass or Amazon EC2 instances, or to collect, process, and transfer data to AWS. Snowcone is designed for data migration needs up to dozens of terabytes (with up to 8 terabytes per device) and from space-constrained environments where AWS Snowball devices will not fit.
- Can come with battery packs to allow for increased versatility in remote locations
- Can use AWS DataSync to allow fast transfer back to AWS

AWS Snowball

- Up to 80TB data transport with on-board storage and compute capabilities. Can be clustered in groups of 5-10. Rack mountable. No battery expansions. Not as portable as Snowcone. Up to 100Gbits transfer speed. HIPAA compliant.
- Targeted at data migrations. Compatible with S3 and EBS volumes. Can use S3 APIs. Can use SSD storage.
- part of the AWS Snow Family, is an edge computing, data migration, and edge storage device that comes in two options.
 - Snowball Edge Storage Optimized devices provide both block storage and Amazon S3-compatible object storage, and 40 vCPUs. They are well suited for local storage and large scale-data transfer.
 - Snowball Edge Compute Optimized devices provide 52 vCPUs, block and object storage. Useful for computer workloads in edge locations.
 - Snowball Edge Compute Optimized with GPU for use cases like AI, advanced machine learning and full motion video analysis in disconnected environments
- You can use these devices for data collection, machine learning and processing, and storage in environments with intermittent connectivity (like manufacturing, industrial, and transportation) or in extremely remote locations (like military or maritime operations) before shipping them back to AWS. These devices may also be rack mounted and clustered together to build larger temporary installations.
- Snowball supports specific Amazon EC2 instance types and AWS Lambda functions, so you can develop and test in the AWS Cloud, then deploy applications on devices in remote

locations to collect, pre-process, and ship the data to AWS. Common use cases include data migration, data transport, image collation, IoT sensor stream capture, and machine learning.

AWS Snowmobile

- Part of AWS Snow Family. An **Exabyte-scale data transfer service used to move extremely large amounts of data to AWS**. You can transfer up to 100PB per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. Snowmobile makes it easy to move massive volumes of data to the cloud, including video libraries, image repositories, or even a complete data center migration. Transferring data with Snowmobile is more secure, fast and cost effective.
- After an initial assessment, a Snowmobile will be transported to your data center and AWS personnel will configure it for you so it can be accessed as a network storage target. When your Snowmobile is on site, AWS personnel will work with your team to connect a removable, high-speed network switch from Snowmobile to your local network and you can begin your high-speed data transfer from any number of sources within your data center to the Snowmobile. After your data is loaded, Snowmobile is driven back to AWS where **your data is imported into Amazon S3**.
- Snowmobile uses multiple layers of security to help protect your data including dedicated security personnel, GPS tracking, alarm monitoring, 24/7 video surveillance, and an optional escort security vehicle while in transit. All data is encrypted with 256-bit encryption keys you manage through the AWS Key Management Service (KMS) and designed for security and full chain-of-custody of your data. GPS tracking available. Video surveillance and alarms.
- Connects to backbone of your network, has over 2km of network cabling. Has chilling unit if too hot and power generator option.

AWS Backup

AWS Backup is a cost-effective, **fully managed**, policy-based service that simplifies data protection at scale. Can be used to centralise backups across multiple regions. You need to create backup policies or backup plans. These determine backup requirements and contain info like:

- Backup schedule
- Backup window
- Lifecycle rules, such as transfer to cold storage timeframes
- Backup vault, including KMS encryption
- Regional copies
- Tags

Once backup plans have been created you can assign resources to them. You can set different backup plans to different resources to meet your backup needs.

Through use of tags you can associate multiple resources at once using tag-based backup policies, this ensures you capture all the required resources at once within your plan.

Backup storage costs can be optimised by using lifecycle rules to transferring from warm to cold storage when possible.

Costs for restore in some cases, in other cases it is free.

Database Services

List of Database Services

- Amazon RDS
- Amazon Aurora
- Amazon DynamoDB
- Amazon Redshift
- Amazon ElastiCache for Memcached
- Amazon ElastiCache for Redis
- Amazon DocumentDB (with MongoDB compatibility)
- Amazon Keyspaces (for Apache Cassandra)
- Amazon Neptune
- Amazon Timestream
- Amazon QLDB (Quantum Ledger Database)

Amazon RDS (Relational Database)

RDS makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server. You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to Amazon RDS.

Easy storage scaling - As your storage requirements grow, you can also provision additional storage. The Amazon Aurora engine will automatically grow the size of your database volume as your database storage needs grow, up to a maximum of 64 TB or a maximum you define. The MySQL, MariaDB, Oracle, and PostgreSQL engines allow you to scale up to 64 TB of storage and SQL Server supports up to 16 TB. These database instances use EBS volumes. Storage classes are General Purpose SSD storage, Provisioned IOPS SSD storage and Magnetic Storage. Storage charged in the units GB months. Storage scaling is on-the-fly with zero downtime. Aurora uses a shared cluster storage, no size options for storage as system will autoscale. You will also be charged for the number of I/Os processed which are billed per million requests.

Database Snapshots - are user-initiated backups of your instance stored in RDS that are kept until you explicitly delete them. You can export them to Amazon S3, filtering by table, schema etc..., perhaps for analysis with Amazon Athena. There is a cost for export to S3. You can create a new instance from a database snapshots whenever you desire. Although database snapshots serve operationally as full backups, you are billed only for incremental storage use. They are kept until the user deletes them.

Automated backups - Amazon RDS creates and saves automated backups of your DB instance during the backup window (0-35 days) of your DB instance to Amazon S3. You can specify encryption used. RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. RDS saves the automated backups of your DB instance according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period. Backup and snapshot storage incurs a charge.

Online transaction processing (OLTP) captures, stores, and processes data from transactions in real time. Online analytical processing (OLAP) uses complex queries to analyze aggregated historical data from OLTP systems

	On-demand Instances	On-demand Instances (BYOL)	Reserved Instances	Reserved Instances (BYOL)	Serverless
 MySQL	●		●		
 PostgreSQL	●		●		
 MariaDB	●		●		
 Aurora	●		●		●
 ORACLE	●	●	●	●	
 Microsoft SQL Server	●		●		

On demand is charged per hour used, partial hours used are charged per seconds used. If any changes made to on-demand instances, there is a minimum 10 minute charge even if DB is terminated or altered again before that time has elapsed.

Bring Your Own Licenses (BYOL)

Reserved instances let you to purchase at a discount an instance type with set criteria for a specified time. Can be as much as 75% cheaper. Must be reserved in 1 or 3 year timeframes. Payment method can further reduce cost, 3 methods: All upfront, partial upfront, or no upfront.

Data transfer costs apply for RDS both into and out of DBs, between AZs, between regions etc...

Multi-AZ RDS Deployment

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) in the same region. Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable. In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of Amazon Aurora), so that you can resume database operations as soon as the failover is complete (about 60-120 seconds, depending on database size and activity). Since the endpoint for your DB Instance remains the same after a

failover, your application can resume database operation without the need for manual administrative intervention. Failover will be performed if:

- Patching is being performed on primary instance
- If primary instance's host fails
- If primary instance's AZ fails
- If the primary instances was rebooted with failover
- If the database instance class on the primary instance is modified

RDS Vertical vs Horizontal Scaling

- **Push-button vertical scaling** - You can scale vertically to address the growing demands of an application that uses a roughly equal number of reads and writes. You can scale the compute and memory resources powering your deployment up or down, up to a maximum of 32 vCPUs and 244 GiB of RAM. Compute scaling operations typically complete in a few minutes. To handle a higher load in your database, you can vertically scale up your master database with a simple push of a button. There are currently over 18 instance sizes that you can choose from when resizing your RDS MySQL, PostgreSQL, MariaDB, Oracle, or Microsoft SQL Server instance. In addition to scaling your master database vertically, you can also improve the performance of a read-heavy database by using read replicas to horizontally scale your database.
- **Read Replicas (horizontal scaling)** make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can create one or more replicas of a given source DB instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput. The read replicas will maintain an asynchronous link between itself and primary DB. RDS MySQL, PostgreSQL, and MariaDB can have up to 5 read replicas. Amazon Aurora can have up to 15 read replicas. Read replicas allow you to create read-only copies that are synchronized with your master database. You can also place your read replica in a different AWS Region closer to your users for better performance. Also, you can use read replicas to increase the availability of your database by promoting a read replica to a master for faster recovery in the event of a disaster. However, read replicas are not a replacement for the high availability and automatic failover capabilities that Multi-AZ provides. Currently, RDS read replicas support transparent load balancing of queries or connections.
 - Each replica has a unique Domain Name Service (DNS) endpoint so that an application can implement load balancing by connecting to the replica endpoint.

Sharding with RDS

- also known as horizontal partitioning, is a popular scale-out approach for relational databases to achieve high scalability, high availability, and fault tolerance for data storage.
- Sharding is a technique that splits data into smaller subsets and distributes them across a number of physically separated database servers. Each server is referred to as a database shard. All database shards usually have the same type of hardware, database engine, and data structure to generate a similar level of performance. However, they have no knowledge of each other, which is the key characteristic that differentiates sharding from other scale-out approaches such as database clustering or replication.
- The share-nothing model offers the sharded database architecture unique strengths in scalability and fault tolerance. There is no need to manage communications and contentions among database members. The complexities and overhead involved in doing so don't exist.

- If one database shard has a hardware issue or goes through failover, no other shards are impacted because a single point of failure or slowdown is physically isolated.
- More information on implementing sharding can be found at this link (<https://aws.amazon.com/blogs/database/sharding-with-amazon-relational-database-service/>)

RDS vs EC2

RDS Pros

With RDS, there is no OS need managed by user for RDS. RDS is fully-managed so AWS takes care of security/encryption (in transit & at rest), backups, high availability and patching/minor version upgrades. Storage optionally managed also. Pre-configured templates available for DBs. Multi-AZ deployment very easy with RDS. Good scalability via read replicas.

EC2 Pros

If you need to run a DB not currently supported by RDS. If you need OS specific settings or external software you need EC2. Advanced configuration possible with EC2, with RDS that layer is abstracted away. Also some advanced features in DBs aren't accessible in RDS. If you need to use certain ports/protocols or if there are specific compliance requirements EC2 will be needed. RDS slightly more expensive.

Amazon RDS Proxy

Amazon RDS Proxy is a fully managed, highly available database proxy for Amazon Relational Database Service (RDS) that makes applications more scalable, more resilient to database failures, and more secure.

Many applications, including those built on modern serverless architectures, can have a large number of open connections to the database server and may open and close database connections at a high rate, exhausting database memory and compute resources. Amazon RDS Proxy allows applications to pool and share connections established with the database, improving database efficiency and application scalability. With RDS Proxy, failover times for Aurora and RDS databases are reduced by up to 66% and database credentials, authentication, and access can be managed through integration with AWS Secrets Manager and AWS Identity and Access Management (IAM).

Amazon RDS Proxy can be enabled for most applications with no code changes. You don't need to provision or manage any additional infrastructure to start using RDS Proxy. Pricing is simple and based on the capacity of underlying database instances. You pay per Aurora Capacity Unit (ACU) for Amazon Aurora Serverless v2 instances or per vCPU for provisioned instances. Amazon RDS Proxy is available for Amazon Aurora with MySQL compatibility, Amazon Aurora with PostgreSQL compatibility, Amazon RDS for MariaDB, Amazon RDS for MySQL, Amazon RDS for PostgreSQL, and Amazon RDS for SQL Server.

You can enable encryption for an Amazon RDS DB instance when you create it, but not after it's created. However, you can add encryption to an unencrypted DB instance by creating a snapshot of your DB instance, and then creating an encrypted copy of that snapshot. You can then restore a DB instance from the encrypted snapshot to get an encrypted copy of your original DB instance.

Amazon Aurora

- is a MySQL and PostgreSQL-compatible relational database (you must choose which compatibility you want at launch) built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases.
- Amazon Aurora is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases. It provides the security, availability, and reliability of commercial databases at 1/10th the cost. Amazon Aurora is fully managed by Amazon Relational Database Service (RDS), which automates time-consuming administration tasks like hardware provisioning, database setup, patching, and backups.
- Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 128TB per database instance. It delivers high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across three Availability Zones (AZs).
- Amazon Aurora serverless can be used, it has no instances to be managed, pricing is measured in Aurora Capacity Units (ACUs).
- Storage autoscaling means that you will be only charged for the exact storage used, so no unused capacity with Aurora.
- Using Aurora you can use ‘Backtrack’ to go back in time on the database to recover from an error or incident, without having to perform a restore or create another DB cluster. More changes means higher cost for storage however.

With Aurora, you can create a clone of the production database quickly and efficiently, without the need for time-consuming backup and restore processes. The development team can spin up the staging database on-demand, eliminating delays and allowing them to continue using the staging environment without interruption.

Amazon DynamoDB

Designed to be a fast and flexible NoSQL database service for any scale. Database that delivers single-digit millisecond performance at any scale. It's a **fully managed** serverless, multiregion, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second. Most common NoSQL models are key-value, document, graph and wide-column DBs. It is highly available (replicated across 3 AZs by default) and infinitely scalable (even with multi-terabyte tables). Best used with Online Transaction Processing (OLTP) workloads. Not recommend for Online analytical processing (OLAP) workloads or ad-hoc query access.

Terminology:

- Item – a row or record – eg. row on a car
- Attributes – columns or fields eg. Car color. Each item can have individualised varying set of attributes, a key benefit over SQL DBs, NOSQL is considered schema-less
- Partition keys – to uniquely identify any item. Can be combined with sort keys to create a primary key even if the partition keys aren't unique on their own
- Sort key – sort data within the partition key. Can be used with partition key to create a composite primary key

Eventual consistency and strong consistency modes describe multi-AZ synchronisation methods.

Supports transactions also, and has ACID (atomicity consistency isolation durability) compliance to ensure full set of operations will happen or none of them will happen.

Global tables – Allows replicating of data across regions, these replicas are active-active. So you can read from any table and write to whatever table is closest to you.

Throughput modes:

- Provisioned throughput – specify read capacity units and write capacity units. DynamoDB Autoscaling can be used to set min and max and DynamoDB will adjust accordingly. Can reserve capacity for discounts.
- On demand capacity – Capacity provided when your DB needs it, DB throughput autoscales, tho it is more expensive per request. Tho zero unused capacity

Backups:

- On-demand backups – able to create a full backup of your data
- Point-in-time recovery – Allow you to go back in time to a database state at any time in last 35 days

Can interact with by AWS console, CLI, SDK and NoSQL Workbench for DynamoDB, uses operations that have a name, parameters and an outputs. Control plane operations to manage DynamoDB tables, data plane operations to manage (CRUD) data in tables. Transactions operations for ACID compliance.

DocumentDB has a 400KB max to upload files

The best solution to meet the RPO and RTO requirements would be to use DynamoDB point-in-time recovery (PITR). This feature allows you to restore your DynamoDB table to any point in time within the last 35 days, with a granularity of seconds. To recover data within a 15-minute RPO, you would simply restore the table to the desired point in time within the last 35 days.

DynamoDB global secondary index is a type of index containing a partition key and a sort key different from the base table's primary key. It is known as the "global" secondary index since the queries on the index can access data from multiple partitions of the base table.

A single DynamoDB table can have multiple GSIs. Applications can largely benefit from this since having multiple secondary keys improve access to data with attributes other than the primary key. In addition, GSIs support non-unique attributes, increasing query flexibility by allowing to run queries against non-key attributes.

The throughput of DynamoDB GSIs is independent of the base table. You need to define the provisioned throughput for the table and each associated GSI when you create them. However, throttling on a GSI can affect the base table when the GSI has an insufficient write capacity.

DynamoDB vs Relational DBs

[Pros of DynamoDB](#)

DynamoDB scales horizontally (add more servers) vs vertically for RDBs (get bigger better server), so DynamoDB is infinitely scalable.

RDBs have fixed schemas, DynamoDB is schemaless so you can adjust columns and datatypes on the fly.

[Pros of Relational DBs](#)

RDB SQL queries are very flexible, DynamoDB cannot offer this level of functionality. It does support PartiQL which is a SQL like language for querying and modifying data.

DynamoDB doesn't support all the data-types that RDBs do, eg. Dates will need to be made into strings for processing

Maximum items size in DynamoDB is 400kB. Can use S3 as a storage option with reference in DB as a workaround.

Soft limits can be adjusted by AWS eg max number of tables per account.

DynamoDB Accelerator (DAX)

You may have requirements where you need microsecond response times in read heavy workloads, this is where DAX comes into play. DAX is a fully managed, highly available caching service built for Amazon DynamoDB. DAX delivers up to a 10 times performance improvement—from milliseconds to microseconds—even at millions of requests per second. All without requiring developers to manage cache invalidation, data population, or cluster management. No need to modify any existing DynamoDB logic as it will work with it also.

DAX deployment can start with a multi-node cluster, containing a minimum of 3 nodes which can then expand to a maximum of 10 nodes (1 primary and 9 read replicas).

As DAX caches data it allows for reduction in provisioned read capacity for DynamoDB, reducing overall cost. If results are not in the cache, it is termed a cache miss. DAX will take read data and then add it to the cache.

DAX supports encryption at rest using AES 256 bit with KMS.

DynamoDB sits outside your VPC and is called by an endpoint. DAX sits within your VPC. You'll need to specify a subnet group (subnets across AZs), with each subnet getting a node.

To let EC2 instances to interact with DAX you'll need a DAX client on the instances, the client will intercept and directs all API calls from the instance to the DAX endpoint. Uses TCP port 811 for communication.

Amazon ElastiCache

- **Fully-managed** in-memory data store, compatible with Redis or Memcached. Power real-time applications with sub-millisecond latency. Amazon ElastiCache allows you to seamlessly set up, run, and scale popular open-source compatible in-memory data stores in the cloud. Can reduce need for scaling up on persistent data store as cache does some of the work for it.
- ElastiCache does not run at edge locations, instead it simply improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory data stores, instead of relying entirely on slower disk-based databases.
- Database query results caching, persistent session caching, Gaming, Geospatial Services, Real-Time Analytics, Social Networking sites, Queuing and full-page caching are all popular examples of caching. Build data-intensive apps or boost the performance of your existing databases by retrieving data from high throughput and low latency in-memory data stores. Cannot replace persistent data stores, but it has many use cases.
- A node is a fixed size chunk of secure network attached RAM. A shard is a group of ElastiCache nodes. Failed nodes automatically replaced.
- Memcached
 - is an easy-to-use, high-performance, in-memory data store. It offers a mature, scalable, open-source solution for delivering sub-millisecond response times making it useful as a cache or session store. Memcached is a popular choice for powering real-time applications in Web, Mobile Apps, Gaming, Ad-Tech, and E-Commerce. Unlike databases that store data on disk or SSDs, Memcached keeps its data in memory. By eliminating the need to access disks, in-memory key-value stores such as Memcached avoid seek time delays and can access data in microseconds. Memcached is also distributed, meaning that it is easy to scale out by adding new nodes. And since Memcached is multithreaded, you can easily scale up compute capacity. As a result of its speed and scalability as well as its simple design, efficient memory management, and API support for most popular languages Memcached is a popular choice for high-performance, large-scale caching use cases.
- Redis
 - Redis, which stands for Remote Dictionary Server, is a fast, open-source, in-memory key-value data store for use as a database, cache, message broker, and queue. Redis now delivers sub-millisecond response times enabling millions of requests per second for real-time applications. All Redis data resides in-memory, in contrast to databases that store data on disk or SSDs. By eliminating the need to access disks, in-memory data stores such as Redis avoid seek time delays and can access data in microseconds. Redis features versatile data structures, high availability, geospatial, Lua scripting, transactions, on-disk persistence, and cluster support making it simpler to build real-time internet scale apps. Cluster mode allows each cluster to have up to 90 shards
- Redis vs Memcached
 - Both are in-memory, open-source data stores. Memcached, a high-performance distributed memory cache service, is designed for simplicity while Redis offers a richer set of features that make it more effective for a wide range of use cases. They work with relational or key-value databases to improve performance, such as MySQL, Postgres, Aurora, Oracle, SQL Server, DynamoDB, and more...

USE CASES



MEMCACHED

Caching

Session Store



REDIS

Caching

Media Streaming

Session Store

Queues

Chat and Messaging

Real-Time Analytics

Gaming Leaderboards

Geospatial

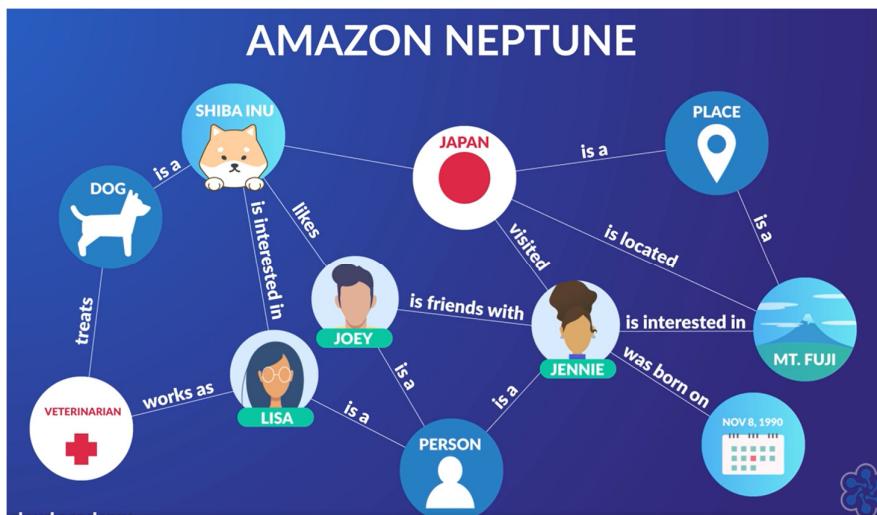
Machine Learning

Use Amazon ElastiCache to manage and store session data.

In order to support distributed session data management in this scenario, it is necessary to use a distributed data store such as Amazon ElastiCache. This will allow the session data to be stored and accessed by multiple EC2 instances across multiple Availability Zones, which is necessary for a scalable and highly available architecture.

Amazon Neptune

- Amazon Neptune is a fast, reliable, fully managed graph database service (stores data, identifies relationships and helps navigate relationships through highly connected data – use cases included social networks identifying trends and links; detecting fraud in finance; eCommerce suggesting products) that makes it easy to build and run applications that work with highly connected datasets. The core of Amazon Neptune is a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency. Amazon Neptune supports popular graph models Property Graph and W3C's RDF, and their respective query languages Apache TinkerPop Gremlin and SPARQL, allowing you to easily build queries that efficiently navigate highly connected datasets. Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security.
- Amazon Neptune is highly available, with read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across Availability Zones. Neptune is secure with support for HTTPS encrypted client connections and encryption at rest. Neptune is fully managed, so you no longer need to worry about database management tasks such as hardware provisioning, software patching, setup, configuration, or backups.
- A Neptune DB cluster contains single or multiple DB instances across different AZs (with the shared volume replicated across at least 3 AZs), in addition to a virtual DB cluster volume which contains data from all instances in the cluster. Uses SSDs up to 64TB to store data.
- Neptune Storage Auto-Repair – Automatically repairs SSDs with faults using other replicated volumes
- Can run replicas, with primary instance handling read and write, while replicas handle read only. Up to x15 replicas per cluster. If the primary instance fails a replica can be promoted to primary.
- Connect to a Neptune instance via endpoint.
 - Cluster endpoint - will point to the designated primary instance in a cluster. For applications that will need both read and write access
 - Reader endpoint – Connect to read replicas, only a single reader endpoint will exist, connections will be performed on round-robin basis, endpoint does not load balance. For applications that will need both read access
 - Instance endpoints – Every instance in the cluster will have an instance endpoint, allows directing certain traffic to certain instances eg for load balancing reasons



Amazon Redshift

- Amazon Redshift is the most widely used cloud data warehouse. Petabyte scale. It makes it very fast, simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. Operates as a relational DB management system. It allows you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data, using sophisticated query optimization, columnar storage on high-performance storage, and massively parallel query execution. Most results come back in seconds.
- A data warehouse allows consolidating data from multiple sources to run business intelligence operations on the data to identify actionable business information. Can perform data operations such as data cleansing to remove incomplete or unwanted records from a record set. ETL (extract transform and load) – extraction describes obtaining data from various sources, data is loaded into a staging area, data is processed, mapped transformed etc... to make data more easily consumed, loading involves transferring data into the data warehouse.
- Very fast and effective DB. Massively Parallel Processing (MPP) across node slices improves performance. Columnar data storage reduces times DB has to perform disk IO. Result caching reduces time to carry out queries by caching results in leader node for later use. Query and load performance data is generated to allow you to track overall performance.
- Can select up to 10 IAM roles to associate with a cluster, allowing RedShift access to other services on your behalf, e.g. S3.
- Amazon Redshift manages the work needed to set up, operate, and scale a data warehouse. For example, provisioning the infrastructure capacity, automating ongoing administrative tasks such as backups, and patching, and monitoring nodes and drives to recover from failures. Redshift also has automatic tuning capabilities, and surfaces recommendations for managing your warehouse in Redshift Advisor. For Redshift Spectrum, Amazon Redshift manages all the computing infrastructure, load balancing, planning, scheduling and execution of your queries on data stored in Amazon S3.
- Cluster is the main component of RedShift, each cluster will run a RedShift engine containing at least 1 DB. Each cluster contains compute nodes, each cluster will have at least 1, if more are provisioned a leader node will be designated which will handle communication between nodes and external apps using the data warehouse. Each node will have its own CPU, memory, etc... which you can select based on your needs. Each node is split into 'node slices'. One table can be distributed across several node slices. Users can configure distribution styles to give control over this.
- Can connect your own BI apps to Redshift, typically using ODBC and JDBC connections.
- The name means to shift away from Oracle, red being an allusion to Oracle

Amazon DocumentDB (with MongoDB compatibility)

- A non-relational fully-managed service which is highly scalable, very fast and has high availability. Runs in a VPC. Highly scalable, very fast and high availability.
 - Allows storage of any JSON-like documents.
 - Can be indexed, which improves speed of retrieving data.
 - Can scale compute and storage separately from each other, allows for flexible scaling pattern:
 - Storage automatically increased by 10GB when needed, up to 64TB
 - Full compatibility with MongoDB. Can easily migrate from MongoDB using AWS DB migration service with minimal changes required.
 - Composed of a cluster, which is comprised of a single or multiple DB instances (up to 16 total) across multiple AZs within the same region. There is a shared cluster storage volume shared between all instances. Single primary instance to do read & write, then optionally read replicas for only read operations. Data synchronisation is maintained synchronously between primary DB and all RRs in the region.
 - Amazon DocumentDB uses endpoints to connect to different components of the DB. This is a URL address with an identified port that points to infrastructure. 3 endpoint types:
 - Cluster – points to the primary DB instance of the cluster. Used if Read and Write access both required. If primary instance fails another instance will be promoted and the endpoint will point to it automatically.
 - Reader – Point to read replicas. Allows read requests to be performed. Only one Reader endpoint will exist even if multiple RRs exist.
 - Instance – For each instance in the cluster a unique endpoint will be assigned. Allows directing traffic to specific instances. Might want to do this for load balancing reasons.
 - Automatic backups can be performed based on schedule specified during DB creation. Can return to any time during the retention period called ‘Point in Time’ recovery, keeps transaction logs to allow this to be possible. Backup window is time in which backups are made, typically during low utilisation time of day.
-

Amazon Keyspaces (for Apache Cassandra)

Apache Cassandra is a free and open-source, distributed, wide-column store, NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. – Wikipedia

Apache Cassandra comprises of a cluster of nodes that need to be provisioned, managed, patched and backed up by you. As database size grows, so much the cluster size also grow, leading to greater administrative load on you. Uses CQL (Cassandra Query Language).

Amazon Keyspaces is a serverless, fully-managed service designed to be highly scalable, highly available and compatible with Apache Cassandra databases. No need to provision, patch and manage instances yourself. Frees you up to focus on operation of the database itself.

Unlimited throughput for massive-scale solutions. Offers extreme performance, scalability and elasticity. Growing at rate of demand for your applications, ensuring you only pay for what you use.

Good for apps where low latency is essential, e.g. route optimisation or trade monitoring. A good solution for managing existing Casandra databases in the cloud.

A ‘Keyspace’ is grouping of tables that are related and is used by your apps to read and write data. Keyspaces help to define how tables are replicated across multiple nodes in the cluster.

A ‘table’ is where database writes are stored. In each table will be a primary key comprising of a partition key and one or more columns.

Encryption at rest automatically enabled, and any clients connecting to tables will require a TLS connection for encrypted in transit connectivity.

Two Throughput Options:

- On-demand – Default mode – Pay for what you use. Will scale to meet demand up to the previous highest demand automatically. If additional throughput needed Keyspaces works quickly to respond. Good for unknown or unpredictable workloads
- Provisioned – Better for predictable workloads. Specify your predicted read and writes per second. These set throughput can be met quickly. Can use automatic scaling if you experience fluctuation or when database naturally grows.

CQL (Cassandra Query Language) similar to SQL, can be used with the CQL editor Amazon Keyspaces dashboard (gets up to 1000 records per query). Also a CQLSH client or an Apache Cassandra client driver can be used to run queries.

Amazon Quantum Ledger Database (QLDB)

A fully managed and serverless database, which has been designed as a ledger database (an example use case would be to recording financial data over a period of time, allowing maintaining of complete history of accounting and transactional data between multiple parties in an immutable, transparent and cryptographic way (via SHA256)).

A database Journal is set to append-only, it is the immutable transaction log that records all entries in a sequenced manner over time. It is similar to blockchain technology, except centralised with a trusted authority, removing consensus requirement across wide network being removed. Integrity assurance is given by the ledger's past not being alterable.

Infrastructure admin not needed as fully managed, all scaling handled by AWS.

Ledger stores data in tables of 'Amazon Ion Documents' (with revisions of those tables over time), an open-source self describing data serialisation format, a superset of JSON, so any JSON document is also a valid Amazon Ion Document. The Amazon Ion Documents allow storage of structured and unstructured data. Database transactions used to make changes to the journal. Each time a change is committed to the journal a sequence number is generated to identify its place in the change history, in addition SHA256 is used for verification purposes to create a cryptographic digest of the journal, helps to ensure data has not been altered.

2 methods of storage, automatically taken care of:

- Journal storage – Holds history of changes made within the ledger database
- Indexed storage – Used to provision the tables and indexes within your ledger. Optimised for querying.

Integrates with Amazon Kinesis via QLDB streams. Kinesis can ingest data, process and analyse it and respond in real time, for example using data from application logs or IoT telemetry data. In this way you can feed a QLDB stream into Kinesis (and other services) to provide benefits, e.g. event driven architectures such as a Lambda event that triggers an SNS notification.

AWS Database Migration Service

- AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases.
- AWS Database Migration Service supports homogeneous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle or Microsoft SQL Server to Amazon Aurora. With AWS Database Migration Service, you can continuously replicate your data with high availability and consolidate databases into a petabyte-scale data warehouse by streaming data to Amazon Redshift and Amazon S3.
- When migrating databases to Amazon Aurora, Amazon Redshift, Amazon DynamoDB or Amazon DocumentDB (with MongoDB compatibility) you can use DMS free for six months.
- The only requirement to use AWS DMS is that one of your endpoints must be on an AWS service. You can't use AWS DMS to migrate from an on-premises database to another on-premises database.

AWS Lake Formation

Can aim for existing data sources within S3, eg log files, or databases (such as relational databases, noSQL databases) etc... It will crawl and catalog the data, in preparation for the analytics to happen. All data can be grabbed all at once or taken incrementally. Blueprints used to identify source data, where to store the data and the frequency with which to load that data. Blueprint will allow discovering source data schema, automatically converting to new formats, partitioning the data based on partitioning schema and keeping track of what was already processed. Blueprints allow for high degree of customisation. Lake Formation will handle security by creating self-service access to the data analytic services, it does this by setting up user access within Lake Formation, by tying data access with Access Control Policies within the Data Catalog; instead of with each individual data analytic service.

Lake Formation is free but the services it uses will be charged for.

Data Lake v Data Warehouse

A data lake is a place to store data, it can be structured or unstructured, meaning it may have a defined schema or not. From here you can perform Data analytics, machine learning and other processing. Generally cheaper to store mass data here than data warehouse, using lower priority and lower cost storage tiering. Data lake must handle:

- Data storage - S3 is optimal for storage of all kinds of data so fits this purpose, especially if tiering and lifecycles are used
- Data movement – Automated is best - active streaming with Kinesis, DirectConnect from on-premises, Database Migration Service or Snowball device.
- Data cataloging and discovery – for purpose of keeping everything organised and structured
 - using metadata eg data format, tags – can be an automated process eg AWS Glue's data catalog - without managing data it will become a data swamp
- Generic analytics
 - Access real time data about your data with Kinesis Data Analytics
 - Use Amazon Athena to analyse data using SQL queries
 - Amazon Quicksights – to create dashboards and graphs
 - Redshift also has analytical functionality
- Predictive analytics
 - Sagemaker to create, train and run analytical models
 - AWS has deep learning AMIs also

A data warehouse allows performance of meaningful analysis on a portion of the overall stored data, to allow decisions to be made. The data within is optimised, by normalisation, transformation and cleaning up the data.

Network Services

Amazon VPC

With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. Up to 5 VPCs per regions per AWS account. Needs a name and IP address range within it can operate, in form of CIDR block

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, in a virtual network that you define. You have complete control over your virtual networking environment, including **selection of your own IP address range, creation of subnets, associate security groups, modifying access control lists and configuration of route tables and network gateways.**

A **subnet** is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.

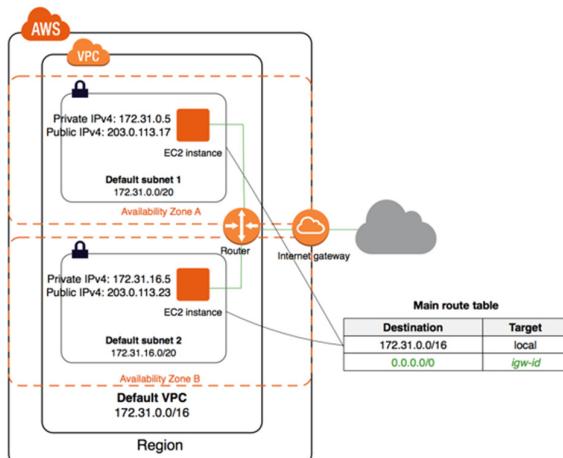
You can easily customize the network configuration for your Amazon VPC. For example, you can create a public-facing subnet for your web servers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access.

Additionally, with AWS, you can choose how network routing is delivered between Amazon VPC and your networks, leveraging either AWS or user-managed network equipment and routes.

To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications. To assign IPv6 address you must associate an IPv6 CIDR block with your VPC.

(<https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html>)



Subnets

Subnets reside in a VPC, allow you to segment VPC into different networks. A subnet is a range of IP addresses in your VPC. You can create AWS resources, such as EC2 instances, in specific subnets.

Each subnet must reside entirely within one Availability Zone and cannot span zones, best practice to spread out subnets across AZs. Can be public or private, public subnets accessible from outside of the VPC ie. available from the internet; this means instances within public subnets will have an IP address.

Each default subnet is a public subnet. Each instance that you launch into a default subnet has a private IPv4 address and a public IPv4 address. These instances can communicate with the internet through the internet gateway.

By default, each instance that you launch into a non-default subnet has a private IPv4 address, but no public IPv4 address, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute. These instances can communicate with each other, but can't access the internet.

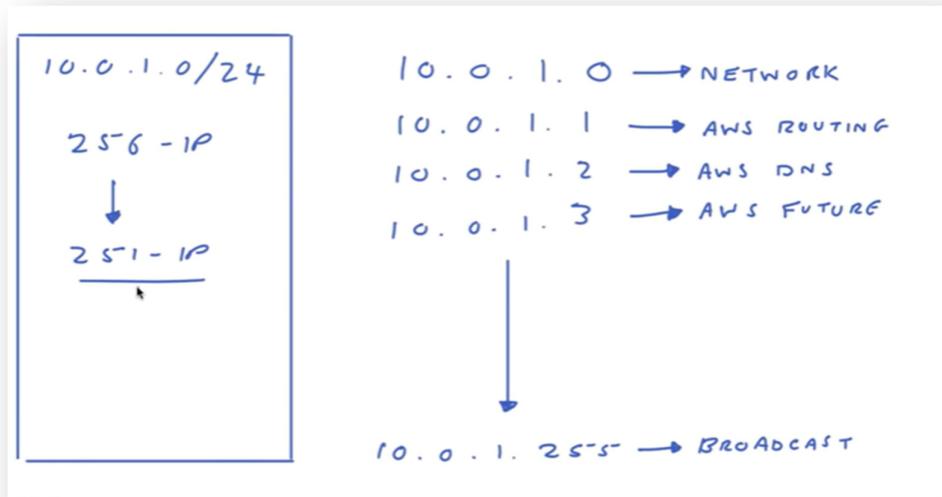
You can enable internet access for an instance launched into a non-default subnet by attaching an internet gateway to its VPC (if its VPC is not a default VPC) and associating an Elastic IP address with the instance.

Route table

Will contain entries which contain 'Destination' (destination trying to reach) and 'Target' (route to the destination). The local destination allows subnets within the VPC to talk to each other. Non-known IP targets can be sent to Internet Gateway via a 0.0.0.0/0 destination via the IGW instance.

Each subnet will also have a **route table**. Same route table can be associated with multiple subnets. Each subnet will have only 1 route table maximum.

1st four and final address are reserved for AWS use:



Security Groups & NACLs

Security Group

- A security group acts as a virtual firewall for your instance 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.
- Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups.
- If you launch an instance and don't specify a security group, the instance is automatically assigned to the default security group for the VPC
- For each security group, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic
- All rules will be assessed in a security group, there is no rule # or order. By default if the conditions match it is assessed as 'Allow', otherwise 'Deny'.
- Security groups can be applied to multiple instances within a VPC, and work across subnets within that VPC.
- Security groups are stateful, so you don't have to configure specific rules to return traffic from requests.

Create security group rules using the security group ID as the source or destination.

This way, the security team can ensure that the least privileged access is given to the application tiers by allowing only the necessary communication between the security groups. For example, the web tier security group should only allow incoming traffic from the load balancer security group and outgoing traffic to the application tier security group. This approach provides a more granular and secure way to control traffic between the different tiers of the application and also allows for easy modification of access if needed.

It's also worth noting that it's good practice to minimize the number of open ports and protocols, and use security groups as a first line of defense, in addition to network access control lists (ACLs) to control traffic between subnets.

Network Access Control List (Network ACL)

- an optional layer of security for your subnet that acts as a firewall for controlling traffic in and out of the VPC and between subnets. Created by default when a subnet is created. When created lets all traffic in-bound and out-bound.
- You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC. Separate rule lists for inbound and outbound traffic & rule # determines order of operation, when rule conditions match rule is applied. Can make an all traffic deny rule at end, so if no other rule is met, then all other traffic is denied.
- Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
- Same NACL can apply to number of subnets, however each subnet can only have one NACL maximum.
- NACLS are stateless, so you have to configure specific rules to return traffic from requests.

[What is the difference between security groups and Network Access Control Lists \(NACLs\)?](#)

Security group	Network ACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in order, starting with the lowest numbered rule, when deciding whether to allow traffic
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	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)

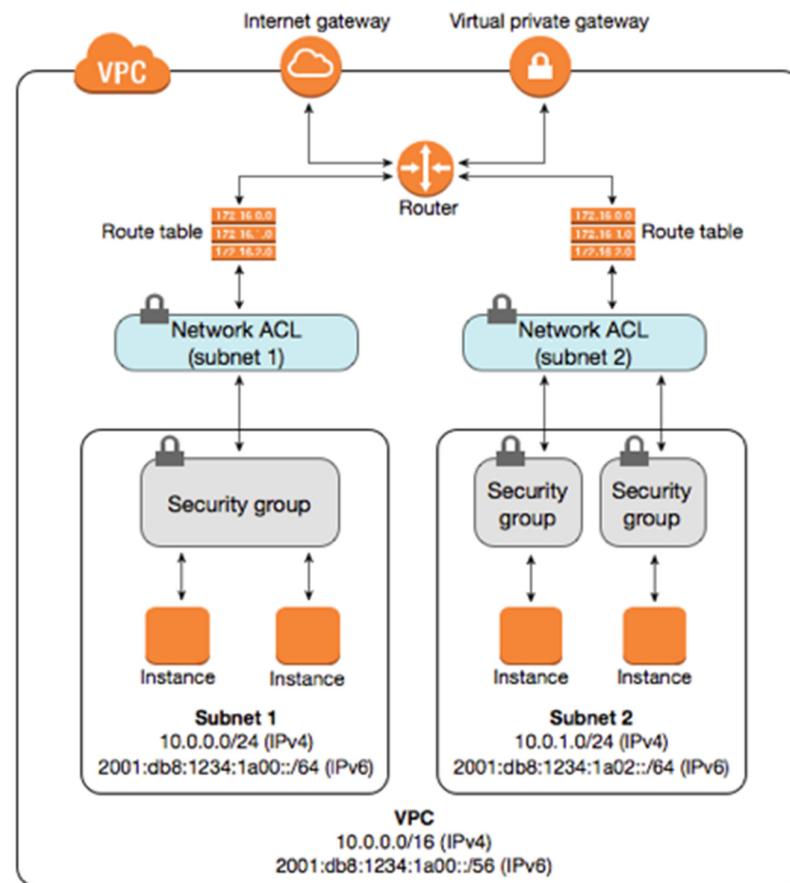
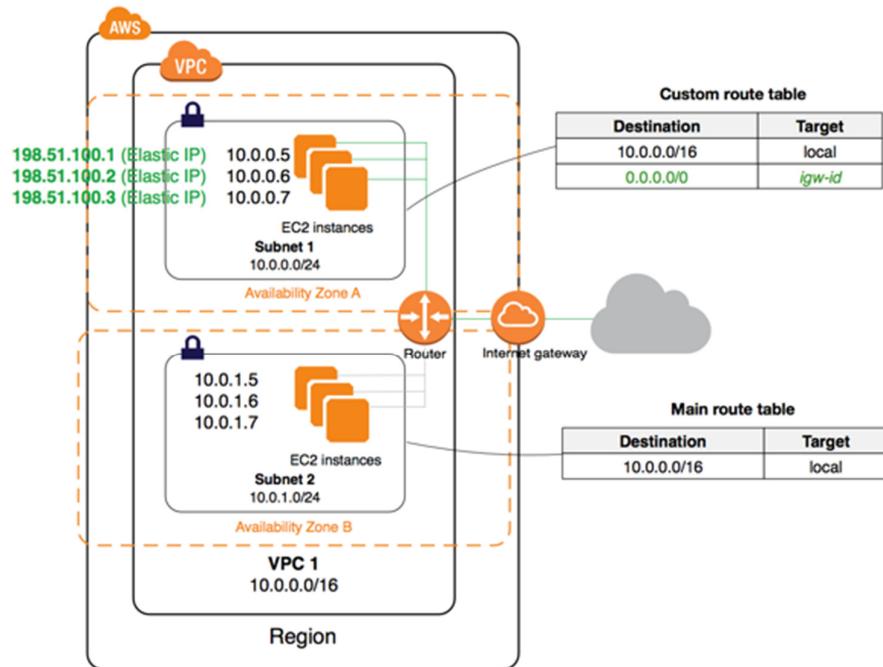


Figure 2: The following diagram illustrates the layers of security provided by security groups and network ACLs. For example, traffic from an internet gateway is routed to the appropriate subnet using the routes in the routing table. The rules of the network ACL that is associated with the subnet control which traffic is allowed to the subnet. The rules of the security group that is associated with an instance control which traffic is allowed to the instance.

Internet Gateway

You control how the instances that you launch into a VPC access resources outside the VPC. Your default VPC includes an **internet gateway**. An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet.



Alternatively, to allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a network address translation (NAT) device for IPv4 traffic. NAT maps multiple private IPv4 addresses to a single public IPv4 address. A NAT device has an Elastic IP address and is connected to the internet through an internet gateway. You can connect an instance in a private subnet to the internet through the NAT device, which routes traffic from the instance to the internet gateway, and routes any responses to the instance.

(<https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html>)

What is VPC Peering?

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection). It is a 1:1 connection, so stringing VPCs together will not work. No IP address overlaps allowed.

Can be used across regions ie. inter-region VPC connections are possible.

pcx- peering connection pre-fix for routing.

VPC Flow Logs

- VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs or Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.
- Flow logs can help you with a number of tasks, such as:
 - Diagnosing overly restrictive security group rule
 - Monitoring the traffic that is reaching your instance
 - Determining the direction of the traffic to and from the network interfaces
- Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance.

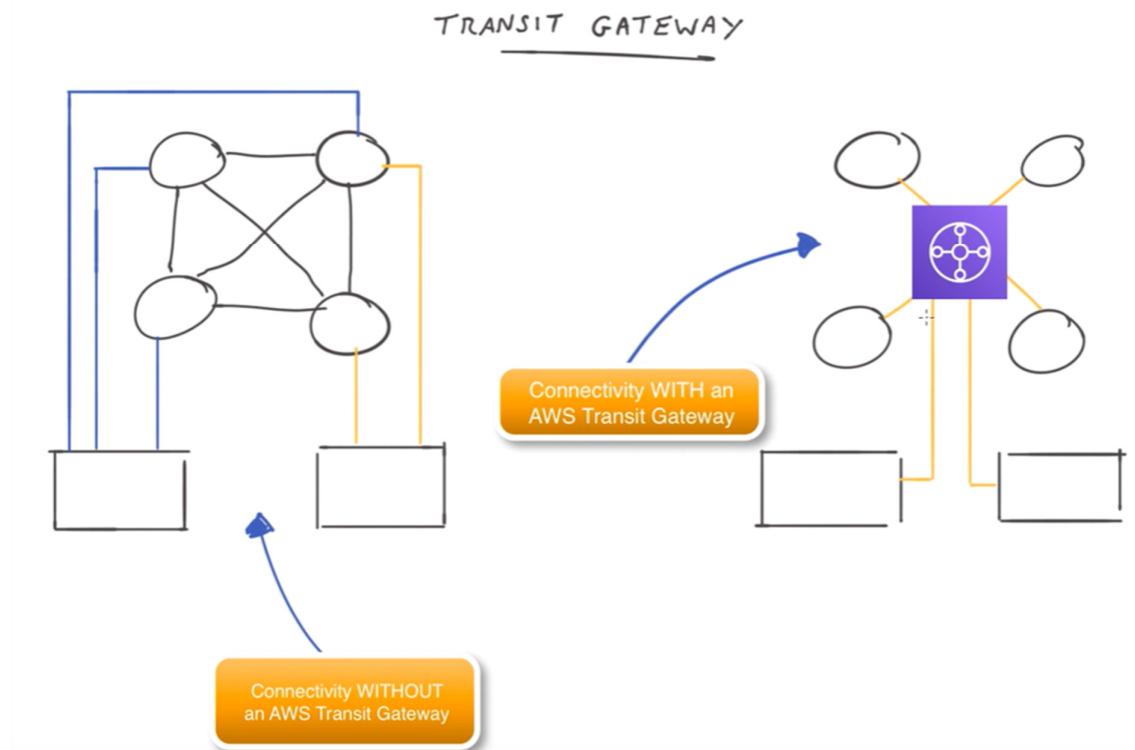
What is a Transit Gateway?

AWS Transit Gateway connects VPCs and on-premises networks through a central hub using a hub-and-spoke (star) connection model. This simplifies your network and puts an end to complex peering relationships, as might occur in a network containing many VPC peering, VPN and Direct Connect connections. It acts as a cloud router – each new connection is only made once. Transit Gateway abstracts away the complexity of maintaining VPN connections with hundreds of VPCs.

Allows for centralisation of monitoring via a single dashboard.

As you expand globally, inter-Region peering connects AWS Transit Gateways together using the AWS global network. Your data is automatically encrypted, and never travels over the public internet. And, because of its central position, AWS Transit Gateway Network Manager has a unique view over your entire network, even connecting to Software-Defined Wide Area Network (SD-WAN) devices.

Max limit is 125 peering connections per VPC currently.



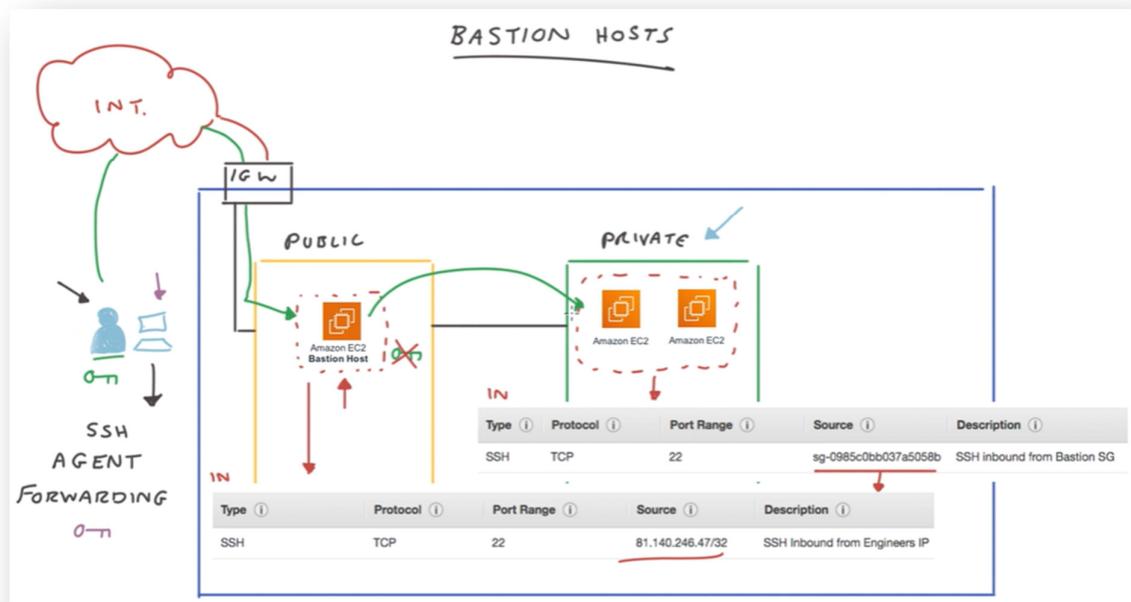
NAT Gateway

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances. Need to assign a route in private subnet route table to the NAT gateway. NAT gateway instance will have an elastic IP. Sits within public subnet. AWS may set up multiple NATs for resiliency, tho user will only see one NAT ID. If you use multi VPCs across multiple AZs, you will need multiple NAT gateways.

NAT gateway will only respond to requests initiated within the VPC.

Bastion Host

The bastion hosts provide secure access to Linux instances located in the private and public subnets of your virtual private cloud (VPC). A bastion host is a very secure and hardened EC2 instance located in the public subnet. Will have its own security group allowing inbound connectivity for SSH (port 22) requests from the IP address of the users you want to access the bastion host. Essentially using the bastion host as a 'jump server' (an intermediary device responsible for funneling traffic through firewalls using a supervised secure channel). The target instances in the private subnet should have their own security group only allowing SSH requests from the bastion host's security group. Keys pairs will be needed, though do not store these on the public subnet, instead use SSH agent forwarding to store SSH keys on outside users' local machines.



QC. Replace the current security group of the bastion host with one that only allows inbound access from the external IP range for the company. Most Voted.

QD. Replace the current security group of the application instances with one that allows inbound SSH access from only the private IP address of the bastion host.

AC. This will restrict access to the bastion host from the specific IP range of the on-premises network, ensuring secure connectivity. This step ensures that only authorized users from the on-premises network can access the bastion host.

AD. This step enables SSH connectivity from the bastion host to the application instances in the private subnet. By allowing inbound SSH access only from the private IP address of the bastion host, you ensure that SSH access is restricted to the bastion host only.

A. Web Server Rules: Inbound traffic from 443 (HTTPS) Source 0.0.0.0/0 - Allows inbound HTTPS access from any IPv4 address

C. Database Rules : 1433 (MS SQL)The default port to access a Microsoft SQL Server database, for example, on an Amazon RDS instance

The security group for the web tier should allow inbound traffic on port 443 from 0.0.0.0/0. This will allow clients to connect to the web tier using HTTPS. The security group for the web tier should also allow outbound traffic on port 443 to 0.0.0.0/0. This will allow the web tier to connect to the internet to download updates and other resources.

The security group for the database tier should allow inbound traffic on port 1433 from the security group for the web tier. This will allow the web tier to connect to the database tier to access data. The security group for the database tier should not allow outbound traffic on ports 443 and 1433 to the security group for the web tier. This will prevent the database tier from being exposed to the public internet.

Elastic IP address

An *Elastic IP address* is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is allocated to your AWS account, and is yours until you release it. By using an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Alternatively, you can specify the Elastic IP address in a DNS record for your domain, so that your domain points to your instance.

Elastic Network Interfaces (ENIs)

An elastic network interface is a logical networking component in a VPC that represents a virtual network card. You can create and configure network interfaces and attach them to instances in the same Availability Zone. You can attach and detach ENI and it will retain its configuration e.g. an Elastic IP address. EC2 has eth0 (primary network interface) as a default network interface, using ENIs you can create more, allowing EC2 instances to be more flexible with how they handle network traffic. Having more than one of them connected to your instance allows it to communicate on two different subnets.

Your account might also have requester-managed network interfaces, which are created and managed by AWS services to enable you to use other resources and services.

Elastic Network Adaptor (ENA)

A custom interface used to optimise network performance, can reach speeds of up to 100Gbps for Linux compute instances, also offers higher bandwidth with increased packets per second (PPS) performance.

It is offered at no extra cost, it is enabled by default, though only supported on some instance types.

VPC Endpoint

A VPC endpoint enables customers to privately connect to supported AWS services and VPC endpoint services powered by AWS PrivateLink. Amazon VPC instances do not require public IP addresses to communicate with resources of the service. Traffic between an Amazon VPC and a service does not leave the Amazon network. Removes need for internet gateways, NAT gateways, VPNs and Direct Connect connections.

VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available Amazon VPC components that allow communication between instances in an Amazon VPC and services without imposing availability risks or bandwidth constraints on network traffic. There are two types of VPC endpoints:

Interface endpoints

Interface endpoints enable connectivity to services over AWS PrivateLink. These services include some AWS managed services, services hosted by other AWS customers and partners in their own Amazon VPCs (referred to as endpoint services), and supported AWS Marketplace partner services. The owner of a service is a service provider. The principal creating the interface endpoint and using that service is a service consumer.

An interface endpoint is a collection of one or more elastic network interfaces with a private IP address that serves as an entry point for traffic destined to a supported service.

Interface endpoints currently support many AWS managed services. Check the documentation for VPC endpoints for a list of AWS services that are available over AWS PrivateLink.

The service that is configured with the interface endpoint can only send responses once a request has been made by your VPC.

Gateway endpoints

A gateway endpoint is a target within your route tables that allows reaching supported services (S3 and DynamoDB) efficiently. Gateway endpoints targets specific IP routes in an Amazon VPC route table, in the form of a prefix-list, used for traffic destined to Amazon DynamoDB or Amazon Simple Storage Service (Amazon S3). Gateway endpoints do not enable AWS PrivateLink. Only works with IPv4.

Deploying a gateway VPC endpoint for Amazon S3 is the most cost-effective way for the company to avoid Regional data transfer charges. A gateway VPC endpoint is a network gateway that allows communication between instances in a VPC and a service, such as Amazon S3, without requiring an Internet gateway or a NAT device. Data transfer between the VPC and the service through a gateway VPC endpoint is free of charge, while data transfer between the VPC and the Internet through an Internet gateway or NAT device is subject to data transfer charges. By using a gateway VPC endpoint, the company can reduce its data transfer costs by eliminating the need to transfer data through the NAT gateway to access Amazon S3. This option would provide the required connectivity to Amazon S3 and minimize data transfer charges.

Load Balancers

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, Lambda functions, and virtual appliances. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers four types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant. Elastic Load Balancing scales with web traffic. ELB may seem like single point of failure, however it is actually multiple instances **managed by AWS**.

Application Load Balancers

Best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing and visibility features targeted at the delivery of modern application architectures, including microservices and containers. Operates at the request level. Advanced routing, TLS termination, visibility features targeted at application architectures. Application Load Balancer routes traffic to targets within Amazon VPC based on the content of the request. You can load balance HTTP/HTTPS applications for layer 7 specific features. Layer 7 uses services such as HTTP, FPT, SMTP and NFS. Analyses HTTP header to direct traffic. Cross-zone load balancing always enabled.

To meet the requirement of forwarding all requests to the website so that the requests will use HTTPS, a solutions architect can create a listener rule on the ALB that redirects HTTP traffic to HTTPS. This can be done by creating a rule with a condition that matches all HTTP traffic and a rule action that redirects the traffic to the HTTPS listener. The HTTPS listener should already be configured to accept HTTPS traffic and forward it to the target group.

Network Load Balancers

Ultra high performance, low latencies. Operates at connection level, Routes traffic to targets within the VPC. Handles millions of requests per second. best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Transport Layer Security (TLS) traffic where extreme performance is required. Network Load Balancer routes traffic to targets within Amazon VPC and is capable of handling millions of requests per second while maintaining ultra-low latencies. You can use strict layer 4 load balancing for applications that rely on the TCP and UDP protocols. If your application requires a static IP, NLB will need to be your choice of ELB. Uses an algorithm based on source port, source IP, TCP sequence, protocol, etc... to determine the target. Preserves source IP addresses while routing. Supports Elastic IP addresses.

I would choose A, as NLB supports HTTP and HTTPS Health Checks, BUT you can't put any URL (as proposed), only the node IP addresses.

Since the NLB does not detect HTTP errors, relying solely on the UnhealthyHostCount metric may not accurately capture the health of the application instances.

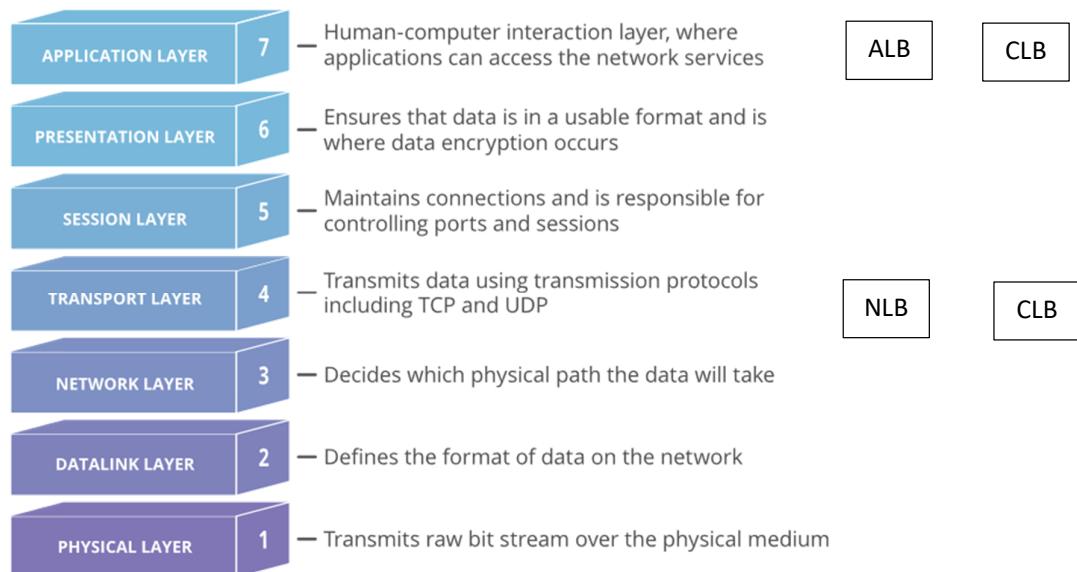
Classic Load Balancers

Provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and the connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network. Supports TCP, SSL/TLS, HTTP and HTTPS protocols. Has advantages over ALB, such as supporting EC2 classic, support for TCP and SSL listeners and support for sticky sessions using application-generated cookies.

Gateway Load Balancers

Makes it easy to deploy, scale, and run third-party virtual networking appliances. Providing load balancing and auto scaling for fleets of third-party appliances, Gateway Load Balancer is transparent to the source and destination of traffic. This capability makes it well suited for working with third-party appliances for security, network analytics, and other use cases. Does health checks to ensure to not send traffic to unresponsive virtual network appliances. Can configure your own rules for routing traffic. Has a VPC Gateway Load balancer endpoint (GWLBe) located in its own VPC, and gateway load balancer located in the virtual appliance VPC. Uses a tunnelling protocol called Geneve, uses port 6081. Uses HTTP port 80 for health checks. Will need to update route tables to allow traffic to move correctly.

Open Systems Interconnection (OSI) Model



The 7-layer OSI (Open Systems Interconnection) Model

(<https://www.cloudflare.com/en-gb/learning/ddos/glossary/open-systems-interconnection-model-osi/>) – Good explanation of OSI model
(<https://www.youtube.com/watch?v=dV8mjZd1OtU>) – Good video explanation of OSI model

Load Balancer Terminology

- **ELB Listener**
 - a process that checks for connection requests using the protocol and port number that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. **Rules** will be used to determine which target group an incoming request will be routed to. These rules will contain conditions and actions that can be setup to determine how the listener will operate.
- **ELB Target**
 - A destination for traffic based on established listener rules
- **ELB Target Group**
 - is a group of targets, for example, a group of EC2 instances
 - routes requests to one or more registered targets using the protocol and port number specified. Eg. HTTP port 80 to target group A, HTTPS 443 to target group B
 - A target can be registered with multiple target groups.
 - Health checks can be configured on a per target group basis. The load balancer continually monitors the health of all targets registered with the target group that are in an Availability Zone enabled for the load balancer. The load balancer routes requests to the registered targets that are healthy
- **Health checks**
 - Performed against the resources defined within the target group. Allow the ELB to contact each target using a specific protocol and receive a response. The load balancer routes requests only to the healthy instances. When the load balancer determines that an instance is unhealthy, it stops routing requests to that instance.
- **Internet-Facing ELB/Internal ELB**
 - Internet-Facing ELB -Nodes of ELB are accessible via the internet and so have a public DNS name that can be resolved to its public IP address, in addition to an internal IP address. Allows ELB to serve incoming request from the internet, before distributing them to the target groups.
 - Internal ELB – Only has an internal IP address. Can only serve requests that originate from within the VPC itself.
- **ELB Nodes**
 - For each AZ selected, an ELB node will be placed within that AZ. You need to ensure you have an ELB node associated to any AZ for which you want to route traffic to. This is because the nodes are used by the ELB to distribute traffic to the target groups.
- **Cross-Zone load balancing**
 - Distributes load evenly even when the targets are located across multiple-AZs

[HTTPS & SSL](#)

HTTPS - This is an encrypted communications channel. To allow an ALB to receive encrypted traffic over HTTPS it will need a server certificate and an associated security policy. The certificate will be an X.509 certificate provisioned by a Certificate Authority such as AWS Certificate Manager (ACM). The certificate is used to terminate the request and in the process the request is decrypted and forwarded to the resources in the ELB target group. To use a third party certificate IAM must be used, this may be needed in regions where ACM is not available.

SSL - Secure Sockets Layer – a cryptographic protocol, much like TLS (Transport Layer Security).

[Load Balancers & Encryption](#)

Elastic Load Balancing simplifies the process of building secure web applications by terminating HTTPS and TLS traffic from clients at the load balancer. The load balancer performs the work of encrypting and decrypting the traffic, instead of requiring each EC2 instance to handle the work for TLS termination. When you configure a secure listener, you specify the cipher suites and protocol versions that are supported by your application, and a server certificate to install on your load balancer. You can use AWS Certificate Manager (ACM) or AWS Identity and Access Management (IAM) to manage your server certificates. Application Load Balancers support HTTPS listeners. Network Load Balancers support TLS listeners. Classic Load Balancers support both HTTPS and TLS listeners.

[ELB and EC2 Auto Scaling](#)

By associating ELB with your EC2 autoscaling group you can dynamically manage load across your resources based on target groups and rules, and scale the groups based on demand. For ALB and NLB you associate the autoscaling group with the ELB target group. When attaching a classic load balancer the EC2 fleet is registered directly with the load balancer. When ELB is attached it will automatically detect the instances in the group and start to distribute traffic.

Amazon Route 53

- Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.
- When using Route 53 a public hosted zone is created which defines how traffic is routed on the public internet. A private hosted zone defines how traffic routed inside VPC, these need to have DNS hostname and DNS support enabled to be used with Route 53. Route 53 creates 4x NS and 1x SOA per zone created. Records have Time To Live (TTL). These zones are comprised of records, there are many record types, most important are:
 - NS - Name Server to identify DNS server for a zone
 - SOA - Start of Authority to define authoritative DNS servers
 - A - Used to map hostname to IP address – for IPv4
 - AAAA - Used to map hostname to IP address – for IPv6
 - MX - To identify email servers for a domain, can have multiple and use #s to specify
 - TXT - To provide info in text format for systems outside domain eg verification and identification. General use type record
 - CNAME – To map a hostname to another hostname – to allow multiple hostnames to be used for a single domain
 - Alias – Maps a custom hostname to an AWS resource
 - Apex – Top node of a DNS namespace
- Routing policy defines how to answer a DNS query, can be:
 - Simple – Provides IP address associated with a name, can contain multiple IP addresses which are selected from randomly. Only routing policy without health checks.
 - Weighted – Assign weights to different records, each weight determines the proportion of times the record is used. If chosen record is unhealthy process repeats until healthy record is found
 - Geolocation – Tags records with location that can be default, continent or country. Allows distribution of resources, to allow catering for users in different countries. Can include a default value for IP addresses that do not map to the listed locations.
 - Geoproximity – Requires Route 53 Traffic Flow. Records are tagged with Lat/Long coordinates. Record chosen is based on distance and bias (allowing you to make traffic flow according to the weight of bias you assign)
 - Failover – Routes to primary resource, then if health check fails, direct to a secondary resource
 - Latency – Chooses record with lowest latency to the customer. AWS maintains DB of latency between general area of users and the regions tagged in DNS records. Lowest latency healthy record is used. Not always closest resource, depends on traffic and other factors.
 - Multivalue answer – Returns up to 8 IP addresses corresponding to healthy records
- Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS.
- You can use Amazon Route 53 to configure DNS health checks (every 30 seconds, or every 10 seconds if desired) to route traffic to healthy endpoints or to independently monitor the health of your application and its endpoints. Can verify health of different tiers of an application ie. different endpoints to verify total app health. Can specify health threshold values for Route 53. Health checks can also work with CloudWatch Alarms to monitor resources. If you do not associate a health check with a record, the record is considered

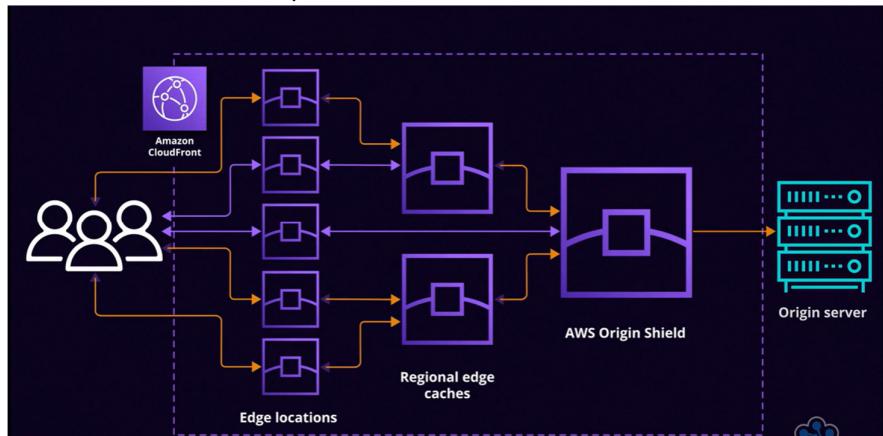
healthy by default. Health checks are specified by domain name or IP address. Can set a string-match condition, so endpoint is considered healthy only if it contains the string within the first 5kb of the response. You can receive SNS notifications of failures.

- Amazon Route 53 Traffic Flow makes it easy for you to manage traffic globally through a variety of routing types, including Latency Based Routing, Geo DNS, Geoproximity, and Weighted Round Robin—all of which can be combined with DNS Failover in order to enable a variety of low-latency, fault-tolerant architectures. Using Amazon Route 53 Traffic Flow's simple visual editor, you can easily manage how your end-users are routed to your application's endpoints - all contained within 'traffic policies', which are automatically versioned so its easy to revert changes —whether in a single AWS region or distributed around the globe. 'Policy records' are created to associate traffic policies with a hosted zone domain or subdomain. Same traffic policy can be used in multiple hosted zones.
- Amazon Route 53 also offers Domain Name Registration – you can purchase and manage domain names such as example.com and Amazon Route 53 will automatically configure DNS settings for your domains.
- Inbound query capability is provided by Route 53 Resolver Endpoints, allowing DNS queries that originate on-premises to resolve AWS hosted domains. Need DirectConnect or VPN connection, need to configure endpoints for DNS queries into and out of VPCs using subnet IPs. Outbound queries are enabled using outbound conditional 'forwarding rules.'
 - Route 53 Resolver DNS Firewall – Managed service for DNS queries that start in VPCs – rule group is defined to determine how it will inspect and filter traffic coming from the VPC, including domain list to inspect in DNS queries and an action to take when a query results in a match. Can allow a query to pass through; pass through with an alert; or block it with a default or custom response.
- Route 53 Application Recovery Controller – Monitors an application's ability to recover from failures and controls the recovery across multiple-AZs, regions and your own datacentre environment. Readiness checks a variety of resources (eg EC2, EBS volumes, ELBs, autoscaling groups, RDS instance and DynamoDB tables) to ensure recovery environment is scaled and configured in case of failure. Routing controls allow shifting traffic during failure scenario. Safety rules prevent failover to unprepared environments or resources. A control panel is a series of routing control for a specific application.

Amazon CloudFront

- When your web traffic is geo-dispersed, it's not always feasible and certainly not cost effective to replicate your entire infrastructure across the globe. A content delivery network (CDN) provides you the ability to utilize its global network of edge locations to deliver a cached copy of web content such as videos, webpages, images and so on to your customers. To reduce response time, the CDN utilizes the nearest edge location to the customer or originating request location in order to reduce the response time. Throughput is dramatically increased given that the web assets are delivered from cache. For dynamic data, many CDNs can be configured to retrieve data from the origin servers.
- What is CloudFront Regional Edge Cache? CloudFront delivers your content through a worldwide network of data centers called edge locations. The regional edge caches are located between your origin web server and the global edge locations that serve content directly to your viewers. This helps improve performance for your viewers while lowering the operational burden and cost of scaling your origin resources. It is essentially a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services.
- CloudFront works seamlessly with services including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code closer to customers' users and to customize the user experience.
- If you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load Balancing, you don't pay for any data transferred between these services and CloudFront. Amazon's CDN offers a simple, pay-as-you-go pricing model with no upfront fees or required long-term contracts, and support for the CDN is included in your existing AWS Support subscription.
- You also have to use S3 in order to make use of CloudFront. CloudFront doesn't work with EBS and EFS.
- Pay-as-you-use service.
- Uses encrypted SSDs to protect data at rest. Can use signed URLs and cookies to restrict access to certain users. AWS WAF can create web ACLs to restrict content access. Geo restrictions possible to prevent certain regions accessing content. Can use IAM to control admin access to CloudFront. Can use CloudWatch, CloudTrail and CloudFront logs to monitor. Origin Access Identity (OAI) and S3 bucket policies can be used for S3 hosted content.
- Works with static (S3 bucket files) and dynamic content (EC2 generated content). Can work with any public endpoint, AWS or external.
- Must create a CloudFront configuration, determining one or more origins that will hold the content to be distributed eg S3 buckets, load balancers; alternate domain name eg cloudacademy.com; protocols to be used eg HTTP or HTTPS; cache time to live (TTL); custom headers; price class to determine whether to use all edge locations or just a subset; AWS WAF ACL associations; alternate domain names; custom SSL certificates; logging; IPv6 support; and more.
- Must configure using Amazon Route 53 also, to ensure traffic travels via CloudFront. Can prevent unauthorised direct access to origin using Amazon Route 53 with custom headers and HTTPS.

- You must ensure traffic travels via CloudFront rather than direct to the origin otherwise CloudFront benefits are not seen.
- Has 3 caching layers:
 - Edge locations
 - Regional edge caches: 13 currently
 - AWS origin shield: A layer between regional edge caches and the origins. Consolidates identical requests also.



We can configure CloudFront to require HTTPS from clients (enhanced security)

AWS Global Accelerator

- Global AWS service not tied to a specific region. Using the public internet, can be negatively impacted by internet congestion and local outages. Gets user traffic to your applications faster and more reliably through the use of AWS's global network infrastructure and specified endpoints, through 80+ global edge locations, then directed to your application origins, improving your internet user performance by up to 60%. When the internet is congested, Global Accelerator's automatic routing optimizations will help keep your packet loss, jitter, and latency consistently low.
- With Global Accelerator, you are provided two global static customer facing IPs to simplify traffic management. On the back end, add or remove your AWS application origins, such as Network Load Balancers, Application Load Balancers, Elastic IPs, and EC2 Instances, without making user facing changes. Automatically avoids unhealthy resources.
- To mitigate endpoint failure Global Accelerator continually monitors the health of your application endpoints and redirects traffic to healthy endpoints, failover between application origins happens automatically and in less than 30 seconds.
- It can be used regardless of how many AWS Regions you are deployed in.
- Also improved security by not using public internet.
- When creating Global Accelerator you must select two IP addresses; then select a listener to receive and process incoming connections based on protocol and ports specified; define endpoint groups associated with different regions; define endpoints (eg load balancers, EC2 instances, EIP addresses, etc) within endpoint groups and percentage of traffic for each endpoint; traffic dials to determine traffic percentage to each endpoint group; health checks for endpoint groups.

AWS Global Accelerator is designed to improve the availability and performance of applications by using static IP addresses (Anycast IPs) and routing traffic over the AWS global network infrastructure.

AWS Global Accelerator directs traffic to the optimal healthy endpoint based on health checks, it can also route traffic to the closest healthy endpoint based on geographic location of the client. By configuring an accelerator and attaching it to a Regional endpoint in each Region, and adding the ALB as the endpoint, the solution will redirect traffic to healthy endpoints, improving the user experience by reducing latency and ensuring that the application is running optimally. This solution will ensure that traffic is directed to the closest healthy endpoint and will help to improve the overall user experience.

Global accelerators can be used for non http cases such as UDP, tcp , gaming , or voip

AWS VPN

AWS Virtual Private Network (VPN) solutions establish secure connections via the public internet between your on-premises networks, remote offices, client devices, and the AWS global network. You can connect your Amazon VPC to remote networks and users using the following VPN connectivity options:

- AWS Site-to-Site VPN: creates encrypted tunnels between your network and your Amazon Virtual Private Clouds. A VPN Connection utilizes IPSec to establish encrypted network connectivity between your intranet and Amazon VPC.
 - On the AWS side of the Site-to-Site VPN connection, a **virtual private gateway** or **transit gateway** provides two VPN endpoints (tunnels) for automatic failover. Any instances inside the VPC will need their route tables updated to point to this virtual private gateway if they are to connect to the external network. Can also enable route propagation on your route tables, so routes representing the VPN connection will be automatically added.
 - You configure your **customer gateway** device on the remote side of the Site-to-Site VPN connection.
 - A VPC tunnel can be initiated between the two gateways, can only be initiated from customer side.
- AWS Client VPN: a managed client-based VPN service that enables you to securely access your AWS resources or your on-premises network. With AWS Client VPN, you configure an endpoint to which your users can connect to establish a secure TLS VPN session. This enables clients to access resources in AWS or a non-premises from any location using an Open VPN-based VPN client.
- AWS VPN CloudHub: If you have more than one remote network (for example, multiple branch offices), you can create multiple AWS Site-to-Site VPN connections via your virtual private gateway to enable communication between these networks
- Third party software VPN appliance: You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a third party software VPN appliance. AWS does not provide or maintain third party software VPN appliances; however, you can choose from a range of products provided by partners and open source communities. You can find third party software VPN appliances on the AWS Marketplace.

Together, they deliver a highly-available, managed, and elastic cloud VPN solution to protect your network traffic.

What is AWS Direct Connect?

AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacentre, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections. Up to 10Gbps.

AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. This allows you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an Amazon Virtual Private Cloud (VPC) using private IP space, while maintaining network separation between the public and private environments. Virtual interfaces can be reconfigured at any time to meet your changing needs.

How does AWS Direct Connect differ from an AWS VPN Connection?

A VPN Connection utilizes IPSec to establish encrypted network connectivity between your intranet and Amazon VPC over the Internet. VPN Connections can be configured in minutes and are a good solution if you have an immediate need, have low to modest bandwidth requirements, and can tolerate the inherent variability in Internet-based connectivity.

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

AWS Private Link

AWS PrivateLink provides private connectivity between VPCs and services hosted on AWS or on-premises, securely on the Amazon network. By providing a private endpoint to access your services, AWS PrivateLink ensures your traffic is not exposed to the public internet. AWS PrivateLink makes it easy to connect services across different accounts and VPCs to significantly simplify your network architecture.

AWS PrivateLink provides private connectivity between virtual private clouds (VPCs), supported AWS services, and your on-premises networks without exposing your traffic to the public internet.

Interface VPC endpoints, powered by PrivateLink, connect you to services hosted by AWS Partners and supported solutions available in AWS Marketplace. There are benefits for both service providers and service consumers, such as:

- More secure as no public internet connection. Can help with regulatory compliance
- Only service consumer can initiate request, preventing unwanted communication from service provider
- Allows on premises resources ability to connect to AWS service endpoint via AWS Direct Connect or VPN connection.

[How does AWS Direct Connect differ from an AWS Private Link?](#)

AWS Direct Connect is the method used to connect private networks, either from customer premises or data center locations, to AWS, notably to the connecting customer's VPC environment built within the cloud. PrivateLink is similar, but slightly different. Unlike Direct Connect, PrivateLink is used as a networking construct inside AWS to privately expose a service/application residing in one VPC (that of a service provider) to other consumer VPCs within an AWS Region.

AWS DataSync

AWS DataSync is a **fully managed** secure, online service that automates and accelerates moving data between on premises and AWS Storage services with end-to-end security, including data encryption (via TLS and encryption at rest) and data integrity validation. Can also be used to transfer data between 2 different AWS storage services.

DataSync supports AWS VPC endpoints, so is able to utilise high bandwidth, low latency AWS network.

DataSync has its own built-in data transfer network protocol and parallel and multi-threaded architecture.

DataSync can copy data between Network File System (NFS) shares, Server Message Block (SMB) shares, Hadoop Distributed File Systems (HDFS), self-managed object storage, AWS Snowcone, Amazon Simple Storage Service (Amazon S3) buckets, Amazon Elastic File System (Amazon EFS) file systems, Amazon FSx for Windows File Server file systems, Amazon FSx for Lustre file systems, Amazon FSz for OpenZFS file systems, Amazon FSx for NetApp ONTAP file systems and AWS Snowcone.

Can be useful even to move files within AWS, eg between EFS instances

3 components:

- Agent – Used on customer side, outside of AWS, typically a VM. Reads and writes data to your storage solution. Not required if transfer is AWS service to AWS service transfer.
- Location – Endpoint of DataSync task, both where you want to move data from and where to
- Task – Details of the operation you are looking to carry out. Location, configuration e.g. data validation techniques, whether you all data transferred or only that which has changed, whether you want to overwrite or delete files, pattern filters to restrict file types transferred and logging details.

• *DataSync uses agents at the source and destination to automatically copy files and file metadata over the network. This optimizes the data transfer and minimizes the impact on your network bandwidth.*

• *DataSync allows you to schedule data transfers and configure transfer rates to suit your needs. You can transfer 30 TB within 5 days while controlling bandwidth usage.*

• *DataSync can resume interrupted transfers and validate data to ensure integrity. It provides detailed monitoring and reporting on the progress and performance of data transfers.*

AWS DataSync is a fully managed data transfer service that simplifies moving large amounts of data between on-premises storage systems and AWS services. It can also transfer data between different AWS services, including different AWS Regions. DataSync provides a simple, scalable, and automated solution to transfer data, and it minimizes the operational overhead because it is fully managed by AWS.

Amazon FSx

Two services:

- Amazon FSx for Windows File Server file systems – Fully-managed native Windows file systems on AWS. Operates as shared file storage. Build on Windows Server. Uses SSD for enhanced throughput and performance. Full support for SMB protocol, Windows NTFS, Active Directory (AD) integration and Distributed File System (DFS). 3 price points: capacity, throughput and backups. Single or multi-AZ possible, but this makes prices increase fast. This option generally more flexible FSx version.
 - Data deduplication – Can save on storage costs by avoiding storing duplicate files.
- Amazon FSx for Lustre file systems – Fully managed file system designed for compute intensive workloads for example machine learning. Able to process massive data sets. Millions of IOPS, throughput of 100s of GB/s and sub-ms latencies. Integration with S3. Supports cloud-bursting workloads from on-premises over DirectConnect and VPN connections.

Joining the FSx for Windows File Server file system to the on-premises Active Directory will allow the company to use the existing Active Directory groups to restrict access to the file shares, folders, and files after the move to AWS. This option allows the company to continue using their existing access controls and management structure, making the transition to AWS more seamless.

Finance Services

AWS Billing and Cost Management

This is the service that you use to pay your AWS bill, monitor your usage, and analyse and control your costs.

AWS automatically charges the credit card that you provided when you signed up for a new account with AWS. Charges appear on your monthly credit card bill. You can view or update your credit card information, including designating a different credit card for AWS to charge, on the Payment Methods page in the Billing and Cost Management console. You can set a specific payment currency here also. AWS Billing and Cost Management provides useful tools to help you gather information related to your cost and usage, analyse your cost drivers and usage trends, and take action to budget your spending.

Analyzing Costs with Cost Explorer

- The AWS Billing and Cost Management console includes the no-cost Cost Explorer tool for viewing your AWS cost data as a graph. With Cost Explorer, you can filter graphs by values such as API operation, Availability Zone, AWS service, custom cost allocation tag, Amazon EC2 instance type, purchase option, AWS Region, usage type, usage type group, and more. If you use consolidated billing, you can also filter by member account. In addition, you can see a forecast of future costs based on your historical cost data.
- Cost allocation tags – are key-value pairs that allow you to organize your AWS resources into groups. For each resource, each tag key must be unique, and each tag key can have only one value. AWS provides two types of cost allocation tags, an AWS generated tags and user-defined tags. You can use tags to:
 - organize your resources, and cost allocation tags to track your AWS costs on a detailed level
 - Visualize information about tagged resources in one place, in conjunction with Resource Groups.
 - View billing information using Cost Explorer and the AWS Cost and Usage report.
 - Send notifications about spending limits using AWS Budgets.
 - Use logical groupings of your resources that make sense for your infrastructure or business. For example, you could organize your resources by:
 - Project
 - Cost center
 - Development environment
 - Application
 - Department

AWS Credits

Reward system to help reduce bills, get credits multiple ways such as developing and publishing skills for Alexa, attending webinars, attending events, AWS Credit program for non-profits, doing AWS certifications.

AWS Cost Anomaly Detection

AWS Cost Anomaly Detection leverages advanced Machine Learning technologies to identify anomalous spend and root causes, so you can quickly take action.

AWS Budgets

- You can use AWS Budgets to track your AWS usage and costs. Budgets use the cost visualization provided by Cost Explorer to show you the status of your budgets. This provides forecasts of your estimated costs and tracks your AWS usage, including your free tier usage. You can also use budgets to create Amazon Simple Notification Service (Amazon SNS) notifications that tell you when you go over your budgeted amounts, or when your estimated costs exceed your budgets.

Total Cost of Ownership (TCO) Calculator

The TCO tool makes a comparison between On Premise IT infrastructure expense the equivalent expense that would exist in the AWS cloud. It then lets the customer know what their cost savings would be if they decided to move their existing IT infrastructure to the AWS cloud.

AWS Pricing Calculator

Configure a cost estimate that fits your unique business or personal needs with AWS products and services. Previously known as Simply Monthly Calculator. Transparent pricing lets you see the math behind the price for your service configurations. View prices per service or per group of services to analyse your architecture costs.

Configure services, or groups of services, in multiple AWS Regions. Prices and availability of AWS services vary per Region.

See and analyse service costs grouped by different parts of your architecture.

Export your estimate to a .csv file to quickly share and analyse your proposed architecture spend.

Amazon CloudWatch Billing Monitoring and Alerts

You can monitor your estimated AWS charges by using Amazon CloudWatch. When you enable the monitoring of estimated charges for your AWS account, the estimated charges are calculated and sent several times daily to CloudWatch as metric data.

Billing metric data is stored in the US East (N. Virginia) Region and represents worldwide charges. This data includes the estimated charges for every service in AWS that you use, in addition to the estimated overall total of your AWS charges.

Alerts and alarms can be set up to notify you when you have reached a specific usage cost in your AWS account. It's a notification that you will receive automatically when a certain level of AWS spend has been reached. This can be set up globally in your AWS account in the Billing & Cost Management Dashboard and region specific in the CloudWatch service.

(<http://kayleigholiver.com/aws-cloud-practitioner-aws-cost-management/>)

AWS Cost & Usage Report (CUR)

The AWS Cost and Usage Reports contains the most comprehensive set of cost and usage data available. You can use Cost and Usage Reports to publish your AWS billing reports to an Amazon Simple Storage Service (Amazon S3) bucket that you own.

You can receive reports that break down your costs by the hour or day, by product or product resource, or by tags that you define yourself. AWS updates the report in your bucket (once a day by default, but customisable to up to three times a day) in comma-separated value (CSV) format. Each update is cumulative, so each version of the Cost and Usage Reports includes all of the line items and information from the previous version.

The reports generated throughout the month are estimated, and subject to change during the rest of the month as you continue to use your AWS services. AWS finalizes the report at the end of each month. Finalized reports have the calculations for your blended and unblended costs, and cover all of your usage for the month. AWS might update reports after they have been finalized if AWS applies refunds, credits, or support fees to your usage for the month.

The report is available within 24 hours of the date that you create a report on the Cost & Usage Reports page of the Billing and Cost Management console.

You can view the reports using spreadsheet software such as Microsoft Excel or Apache OpenOffice Calc, or access them from an application using the Amazon S3 API. You can also load your cost and usage information into Amazon Athena, Amazon Redshift, AWS QuickSight, or a tool of your choice.

You can create, retrieve, and delete your reports using the AWS CUR API Reference

AWS Cost and Usage Reports tracks your AWS usage and provides estimated charges associated with your account. Each report contains line items for each unique combination of AWS products, usage type, and operation that you use in your AWS account.

Encryption Services

What is encryption?

Data encryption is the mechanism by which information is altered, rendering plaintext data unreadable, through use of mathematical algorithms and encryption keys. The resulting output is ciphertext, a decryption key is needed to revert data to readable format.

Encryption key is a string of characters used with an algorithm, longer key means more robust encryption.

Algorithms:

- AES – Advanced encryption standard
- DES – Digital encryption standard
- Triple DES
- Blowfish
- RSA
- Diffie-Hellman
- Digital Signature Algorithm

Symmetric cryptography – Single key used to encrypt and decrypt data. Key must be protected, otherwise all encryption is insecure. Faster than asymmetric at encrypting and decrypting.

Asymmetric cryptography – Separate keys used to encrypt and decrypt data. Public key shared with anyone, who will use it to encode messages for you. Private key is kept by you to decrypt the messages. Public key does not need secure transmission, cannot decrypt data, need a private key to decrypt. There is mathematical link between the two keys. Asymmetric cryptography data overcomes the problem of the key being discovered during transmission.

Encryption is a way of scrambling data so that only authorized parties can understand the information. In technical terms, it is the process of converting human-readable plaintext to incomprehensible text, also known as ciphertext. Ideally, only authorized parties can decipher a ciphertext back to plaintext and access the original information.

Encryption Keys

The Public and Private key pair comprise of two uniquely related cryptographic keys (basically long random numbers) known as a **key pair**. Below is an example of a Public Key:

```
3048 0241 00C9 18FA CF8D EB2D EFD5 FD37 89B9 E069 EA97 FC20 5E35 F577 EE31C4FB C6E4 4811  
7D86 BC8F BAFA 362F 922B F01B 2F40 C744 2654 C0DD 2881 D673 CA2B4003 C266 E2CD CB02 0301  
0001
```

The Public Key is what its name suggests - Public. It is made available to everyone via a publicly accessible repository or directory. On the other hand, the Private Key must remain confidential to its respective owner.

Because the key pair is mathematically related, whatever is encrypted with a Public Key may only be decrypted by its corresponding Private Key and vice versa.

For example, if Bob wants to send sensitive data to Alice, and wants to be sure that only Alice may be able to read it, he will encrypt the data with Alice's Public Key. Only Alice has access to her corresponding Private Key and as a result is the only person with the capability of decrypting the encrypted data back into its original form.

As only Alice has access to her Private Key, it is possible that only Alice can decrypt the encrypted data. Even if someone else gains access to the encrypted data, it will remain confidential as they should not have access to Alice's Private Key.

AWS Certificate Manager (ACM)

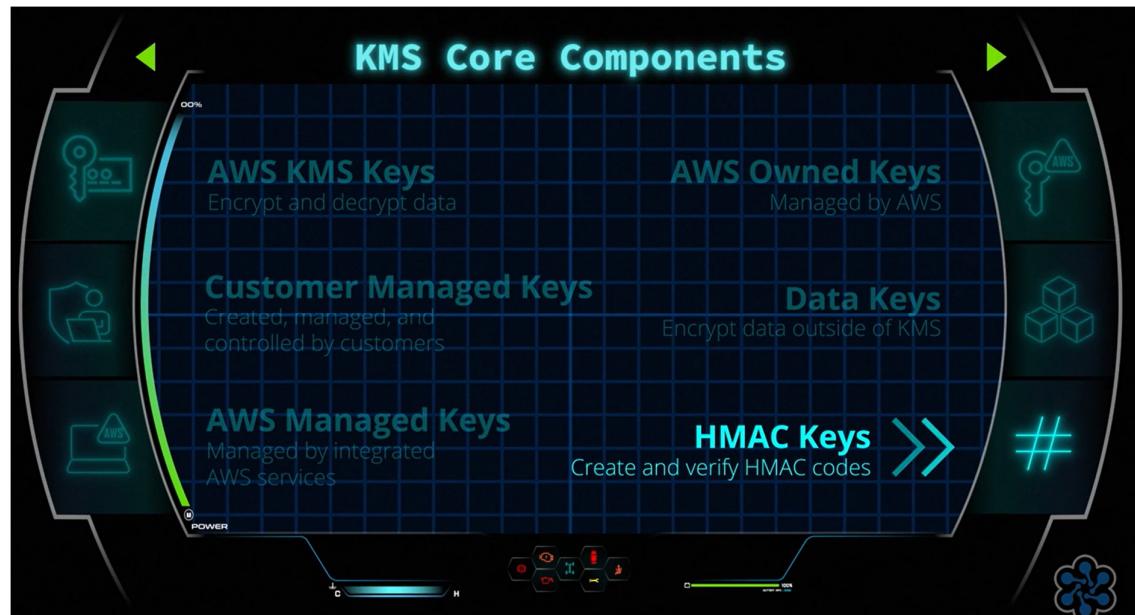
- AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources. SSL/TLS certificates are used to secure network communications and establish the identity of websites over the Internet as well as resources on private networks. AWS Certificate Manager removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates.

To increase the application's performance, the solutions architect should import the SSL certificate into AWS Certificate Manager (ACM) and create an Application Load Balancer with an HTTPS listener that uses the SSL certificate from ACM.

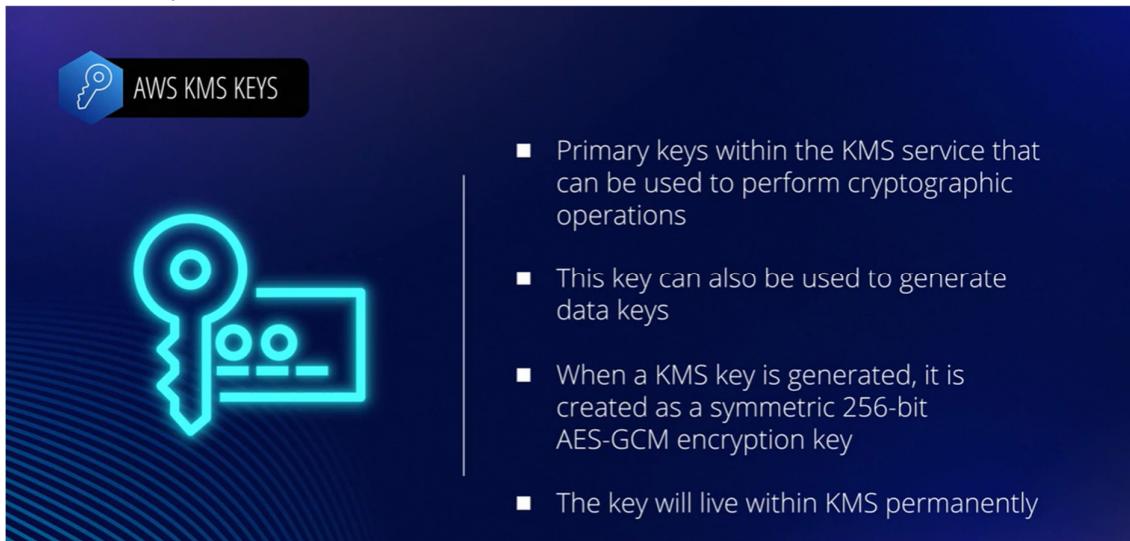
An Application Load Balancer (ALB) can offload the SSL termination process from the EC2 instances, which can help to increase the compute capacity available for the web application. By creating an ALB with an HTTPS listener and using the SSL certificate from ACM, the ALB can handle the SSL termination process, leaving the EC2 instances free to focus on running the web application.

AWS Key Management Service (KMS)

- A managed **regional** service to easily create, store and control the customer master keys (CMKs), the encryption keys used to encrypt or digitally sign your data. Makes it easy for you to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications. AWS KMS is a secure and resilient service that uses hardware security modules that have been validated under FIPS 140-2, or are in the process of being validated, to protect your keys. AWS KMS is integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs. Keys must remain highly secure so admins and employees of AWS will not have access to customer keys and cannot recover them if they are deleted. KMS can only perform encryption at rest using these keys, KMS does not perform encryption for data in transit, for this you'd need to use SSL or another method. Works with CloudTrail to track key usage.



AWS KMS Keys



The slide features a blue header bar with the text "AWS KMS KEYS" and a key icon. Below the header is a large graphic of a glowing blue key icon. To the right of the key, a vertical line separates the icon from a bulleted list of four points.

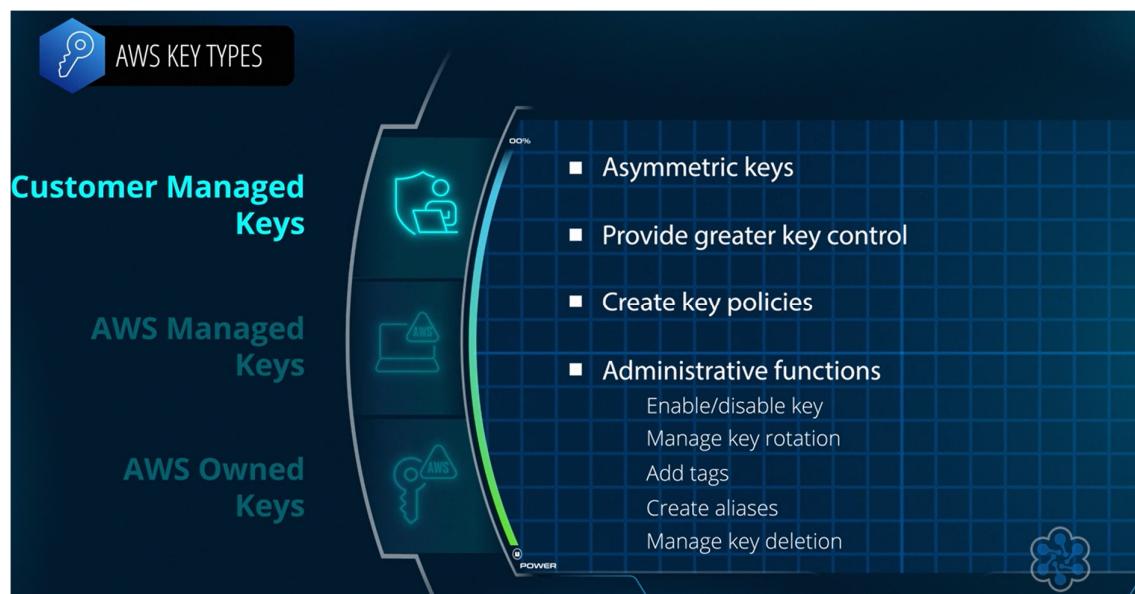
- Primary keys within the KMS service that can be used to perform cryptographic operations
- This key can also be used to generate data keys
- When a KMS key is generated, it is created as a symmetric 256-bit AES-GCM encryption key
- The key will live within KMS permanently

Will reside in KMS permanently as if a symmetrical key is compromised all encryption used for it is insecure then. Almost always simply stored on HSMs.

Asymmetric KMS Keys can be used for encryption and signing but not both. In this case the public key will be usable outside of KMS, accessible via KMS operation calls and downloadable. The private key will never leave KMS without being encrypted. Generally the use of asymmetric keys are required for when encryption is needed outside of AWS or when users can't call on KMS directly

Customer Managed Keys

Keys the customer creates

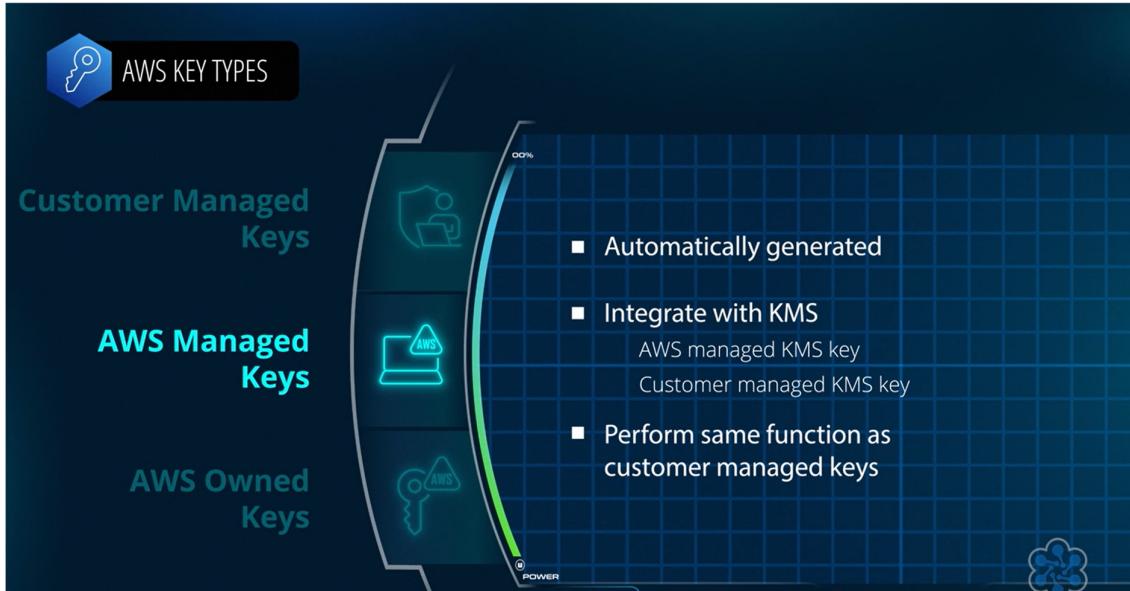


The slide features a blue header bar with the text "AWS KEY TYPES" and a key icon. Below the header are three categories listed vertically: "Customer Managed Keys", "AWS Managed Keys", and "AWS Owned Keys". To the right of these categories is a vertical line with a circular arrow icon, leading to a grid of icons and a list of features.

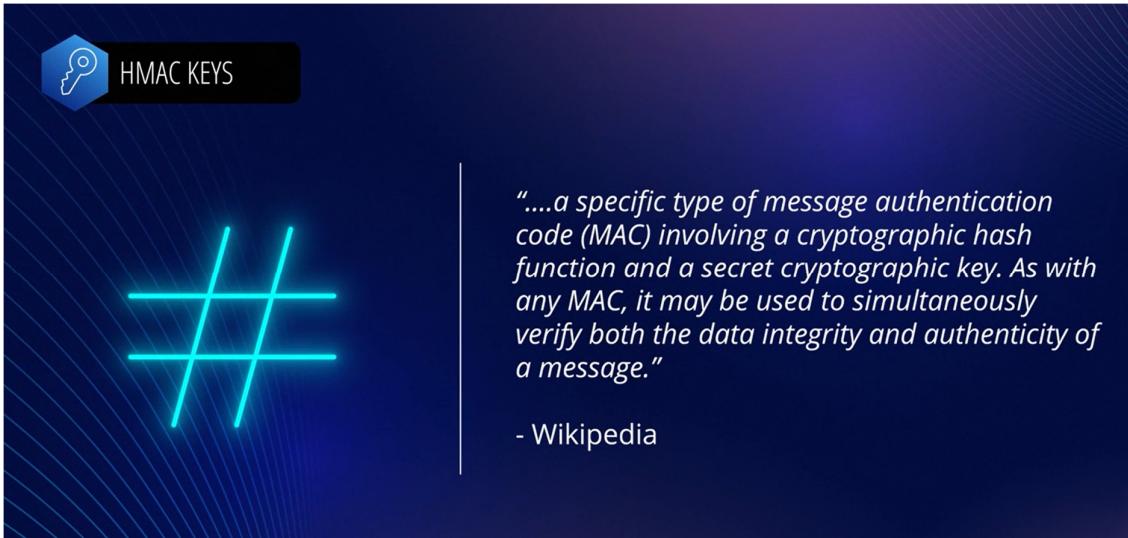
- Asymmetric keys
- Provide greater key control
- Create key policies
- Administrative functions
 - Enable/disable key
 - Manage key rotation
 - Add tags
 - Create aliases
 - Manage key deletion

AWS Managed Keys

Generated by AWS services and will only be used by the service that generates them.



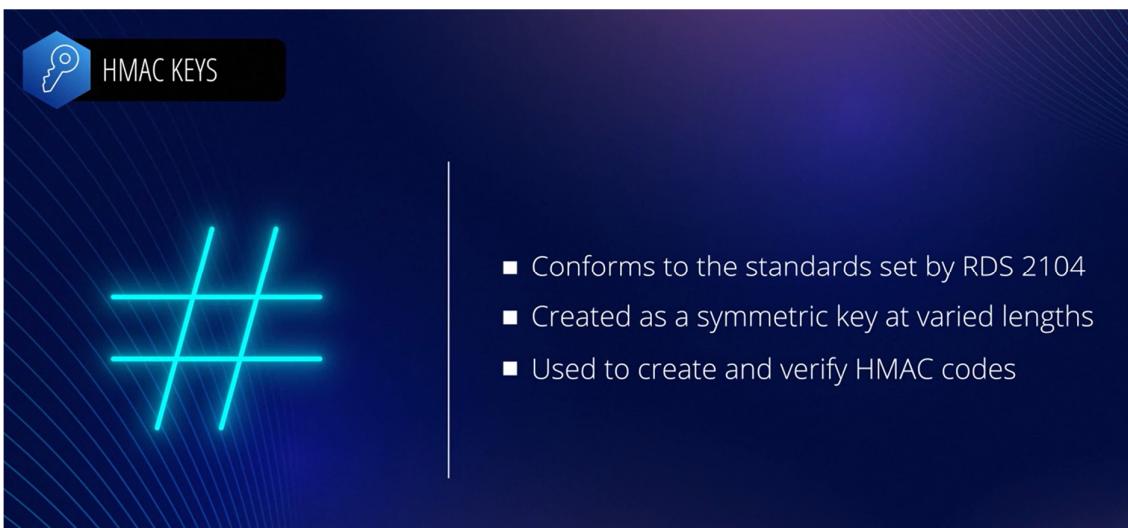
HMAC - Hash-based metric authentication codes



HMAC KEYS

.....*a specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and authenticity of a message.*

- Wikipedia



HMAC KEYS

.....

- Conforms to the standards set by RDS 2104
- Created as a symmetric key at varied lengths
- Used to create and verify HMAC codes

HMAC algorithms take HMAC algo takes message/data and combine it with HMAC key material to create a unique fixed size tag that is associated with the data. When viewing the encrypted data it will be able to detect if any of the data has been changed. Key material embedded within the HMAC key never leaves the KMS service without being encrypted first, as it contains the secret key that associated HMAC algos will use. Operations GenerateMac and VerifyMac will be used.

Data Keys

Generated by KMS, used for encrypting outside of KMS e.g. with AWS Encryption SDK. Use GenerateDataKey operation to generate a key from an existing encryption key. Data key used to encrypt data, will need the original KMS encryption key to help do this. Classed as symmetric encryption as it only uses one key.

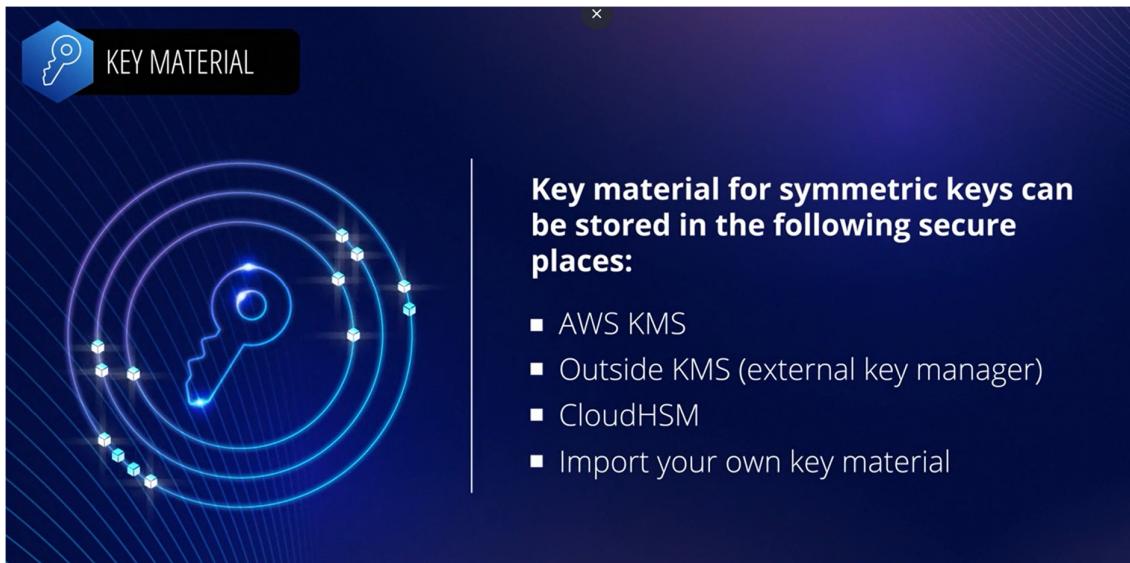
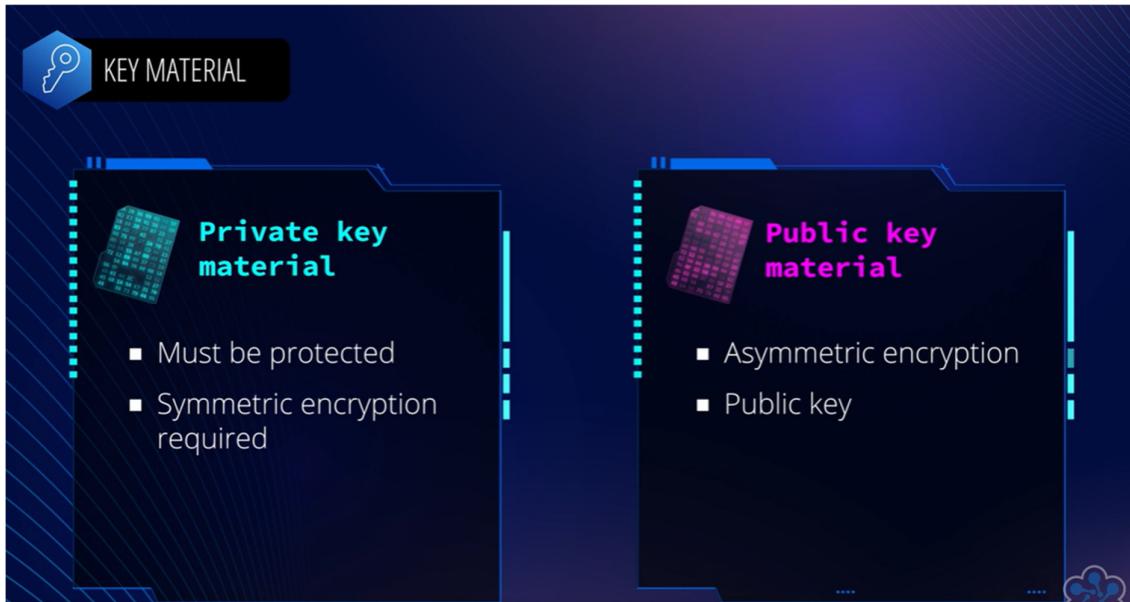
Data Key Pairs

Asymmetric encryption, for use outside of KMS. Private key kept in KMS, plaintext or encrypted with KMS key of your choice. Public key is created also.



Key Material

Part of all keys in AWS, part of key used with cryptographic algorithm, effectively just a string of bits.



If KMS key will be stored across multiple regions, the key material for each will be the same



Key Rotation

Processing of changing the key material with new key material for keys. KMS automatable method is recommended. Key doesn't change just key material. All prior key material is obtained, allowing KMS to decrypt for all prior versions of the key material.

Key Policy

Tied to keys making them resource-based policies.



Grants



GRANTS



- Temporary-based policy
- Resource-based policy
- Allows you to delegate a subset of your access to a KMS Key for another principal, such as another user within your AWS account



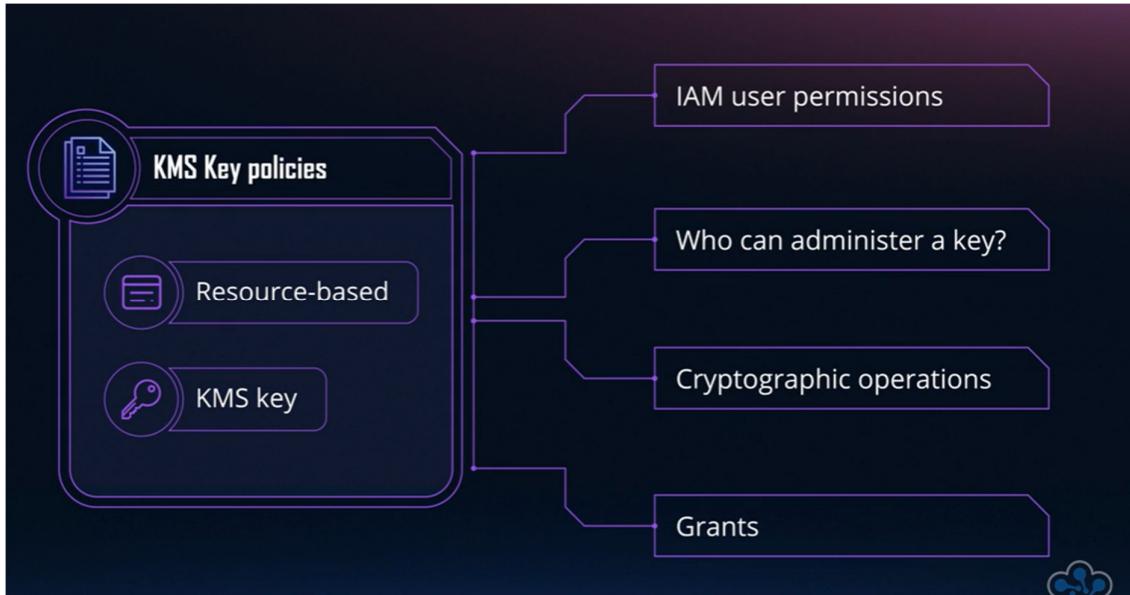
GRANTS



- There is less risk of someone altering the access control permissions for that KMS Key
- A grant is created and applied to the KMS Key for each principle requiring access

[KMS Key Policy](#)

IAM policies can be used also, but IAM must be allowed from within the key policy.



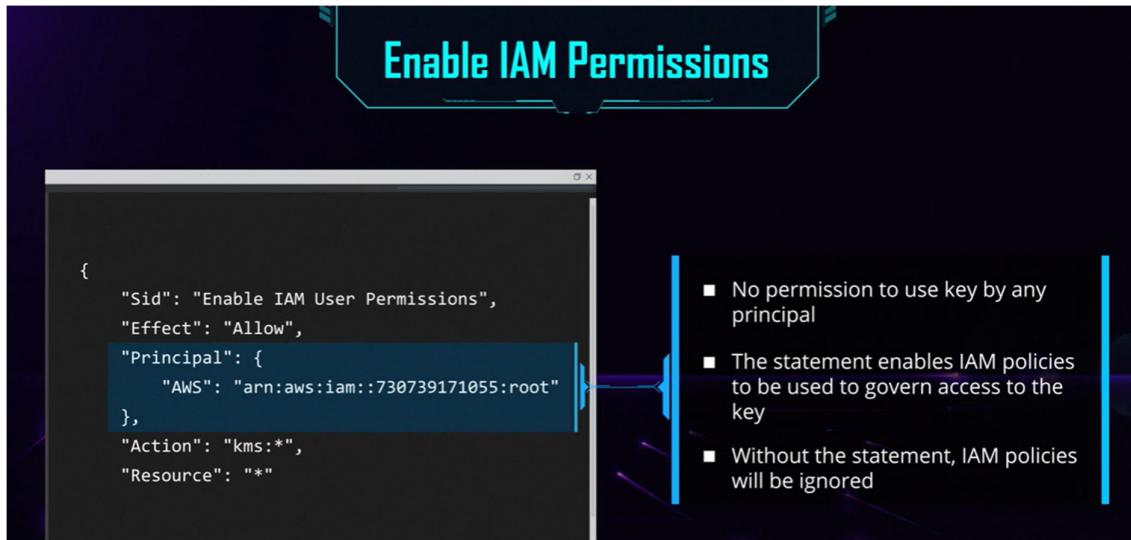


KMS Key policies

JSON



```
        "Resource": "*"
    },
{
    "Sid": "Allow access for Key Administrators",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::730739171055:user/Cloudacademy"
    },
    "Action": [
        "kms>Create*",
        "kms>Describe*",
        "kms>Enable*",
        "kms>List*",
        "kms>Put*",
        "kms>Update*",
        "kms>Revoke*",
        "kms>Disable*",
        "kms>Get*",
        "kms>Delete*",
        "kms>TagResource",
        "kms>UntagResource",
        "kms>ScheduleKeyDeletion",
        "kms>CancelKeyDeletion"
    ],
    "Resource": "*"
}
```



First Sid we look at, states if IAM User permissions are enabled. Without this statement IAM policies will be ignored unless they're being used to deny access.



If you don't want to manage keys with IAM, need to remove Sid entry from key policy.



Good to have root access, as its not possible to delete root account, so keys wont ever become unusable.



This Sid specifies users who can administer KMS keys, but not perform any encryption functions using that key. Can specify if you want to allow key deletion. They can update key policies, so they could update policy to give themselves permissions as a user of the key.

Allow Use of Key: Two Users

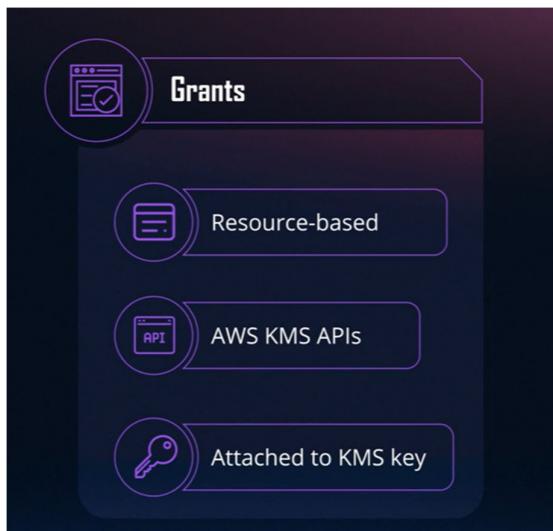
```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::730739171055:user/Stuart",
            "arn:aws:iam::730739171055:user/Cloudacademy"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
```

Allow use of key sid, states who can use the key, e.g. encrypt, decrypt, generate data keys, get info about keys.

Grants

```
{  
    "Sid": "Allow attachment of persistent resources",  
    "Effect": "Allow",  
    "Principal": {  
        "AWS": [  
            "arn:aws:iam::730739171055:user/Stuart",  
            "arn:aws:iam::730739171055:user/Cloudacademy"  
        ]  
    },  
    "Action": [  
        "kms>CreateGrant",  
        "kms>ListGrants",  
        "kms:RevokeGrant"  
    ],  
    "Resource": "*",  
    "Condition":{  
        "Bool":{  
            "kms:GrantIsForAWSResource": "true"  
        }  
    }  
}
```

Next sid is allow attachment of persistent resources, i.e. allow use of grants to delegate access a certain user has to another user on a temporary basis, without having to update key policy. Some AWS services use grants.



Grants must be generated via AWS KMS APIs



Can only be for single KMS key. Grantee principal can be user, role or federated user or role; cant be IAM group, AWS Organization or AWS service role. Key policy statements for grant operations only apply to allow operations. Grants may have slight delays before becoming effective, if you want immediate results you will need to use a grant token. Grant ID is issued.

Access to KMS Keys



Multi-Region Keys

AWS KMS supports multi-Region keys, which are AWS KMS keys in different AWS Regions that can be used interchangeably – as though you had the same key in multiple Regions. Each set of related multi-Region keys has the same key material and key ID, so you can encrypt data in one AWS Region and decrypt it in a different AWS Region without re-encrypting or making a cross-Region call to AWS KMS.

Like all KMS keys, multi-Region keys never leave AWS KMS unencrypted. You can create symmetric or asymmetric multi-Region keys for encryption or signing, create HMAC multi-Region keys for generating and verifying HMAC tags, and create multi-Region keys with imported key material or key material that AWS KMS generates. You must manage each multi-Region key independently, including creating aliases and tags, setting their key policies and grants, and enabling and disabling them selectively. You can use multi-Region keys in all cryptographic operations that you can do with single-Region keys.

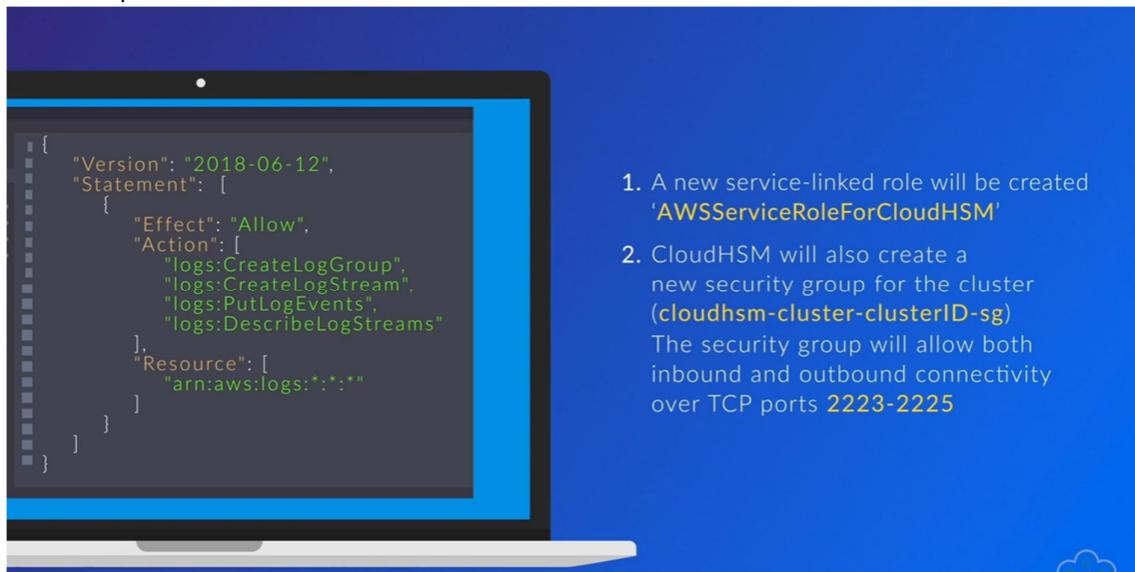
Multi-Region keys are a flexible and powerful solution for many common data security scenarios.

AWS CloudHSM (Hardware Security Module)

The infographic is divided into three main sections:

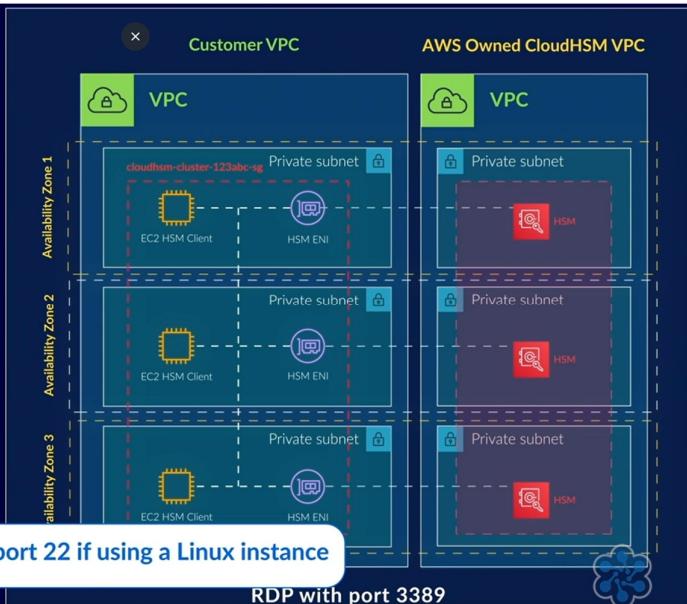
- Top Section:** A title "Hardware Security Module" is followed by two bullet points:
 - A physical tamper-resistant hardware appliance that is used to protect and safeguard cryptographic material and encryption keys
 - Provides Federal Information Processing Standards (FIPS) **140-2 Level 3**A graphic shows a smartphone with a lock icon and a key icon, symbolizing security.
- Middle Section:** A title "CloudHSM is a physical device" is followed by three bullet points:
 - NOT a multi-tenant device
 - CloudHSM used for secure encryption key management and storageA graphic shows a physical padlock with a key, symbolizing secure key management.
- Bottom Section:** A title "Creation, storage and management of cryptographic keys" is followed by five bullet points:
 - Ability to generate cryptographically secure random data
 - The ability to use cryptographic hash functions
 - HMACs
 - Creation, storage and management of cryptographic keysA graphic shows two large yellow keys and a digital representation of a key being hashed into a digital signature.

- service helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated Hardware Security Module (HSM) instances within the AWS cloud. AWS and AWS Marketplace partners offer a variety of solutions for protecting sensitive data within the AWS platform, but for some applications and data subject to contractual or regulatory mandates for managing cryptographic keys, additional protection may be necessary.
- CloudHSM complements existing data protection solutions and allows you to protect your encryption keys within HSMs that are designed and validated to government standards for secure key management. CloudHSM allows you to securely generate, store, and manage cryptographic keys used for data encryption in a way that keys are accessible only by you.
- A Hardware Security Module (HSM) provides secure key storage and cryptographic operations within a tamper-resistant hardware device. HSMs are designed to securely store cryptographic key material and use the key material without exposing it outside the cryptographic boundary of the hardware.
- Will be run as a cluster of HSMs to provide high availability, load balanced across availability zones and between HSMs in the cluster. You'll need a VPC for this cluster based setup. HSM ENI is placed within subnet of VPC.



- Once cluster is defined and created in different availability zones, it will have been provisioned in an uninitialized state. HSMs will need to be initialised.
- EC2 instance can connect to HSM ENI, provisioned in the same VPC. You must configure a security group (cloudhsm-cluster-clusterID-sg) and also install AWS CloudHSM client software on instance.
- IAM role allows HSM to send log data to CloudWatch Logs log groups and log streams.

1. Configure a security group
Add your instance to the cloudhsm-cluster-*clusterID*-sg
2. Install the AWS CloudHSM client software on your instance





Once you have installed the client and tools, you need to modify the client configuration to enable you to connect to your cluster:

Copy your issuing certificate

/opt/cloudhsm/etc/customerCA.crt

Run the following command

```
sudo /opt/cloudhsm/bin/configure -a <IP address>
```

Updating server config in /opt/cloudhsm/etc/cloudhsm_client.cfg

Updating server config in /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg



Windows 10

If you are using a Windows instance, then you will need to download the installation from here:

↓

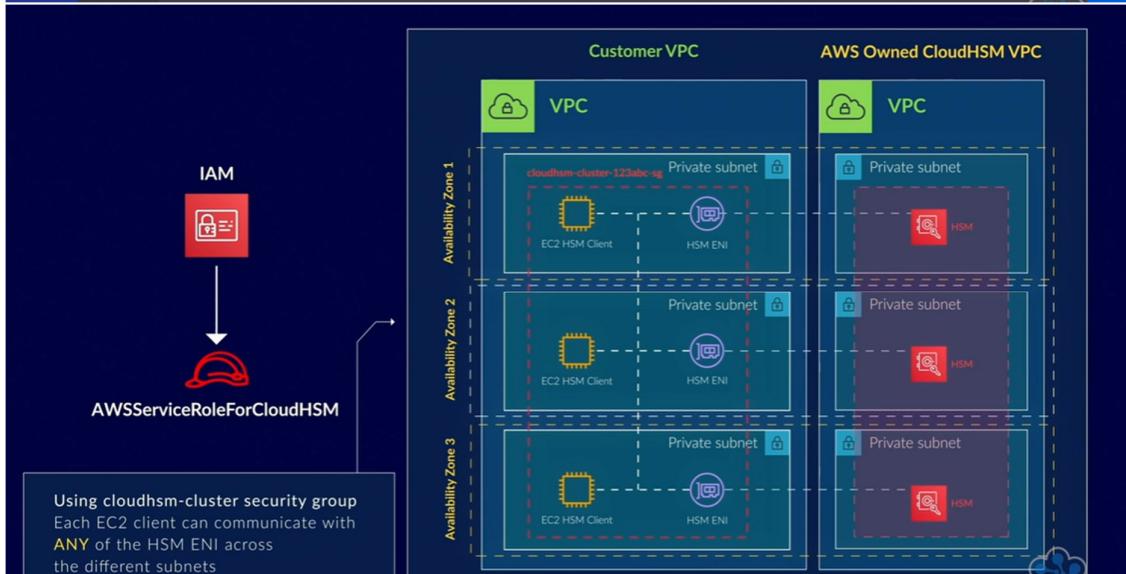
<https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Windows/AWSCloudHSMClient-latest.msi>

Copy your issuing certificate

C:\ProgramData\Amazon\CloudHSM

Run the following command

C:\Program Files\Amazon\CloudHSM\configure.exe -a <HSM IP address>



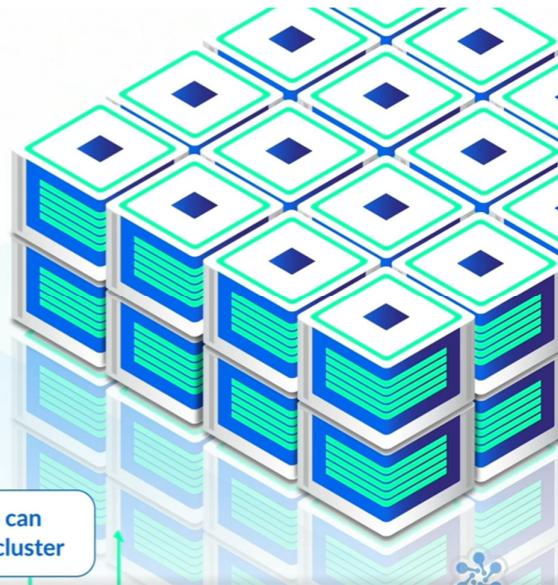
The default key stores are:

- Managed by KMS
- Stored on HSMs managed by AWS

This means that you have no control over these HSMs



 However, by creating a custom key store you can have full management over your CloudHSM cluster



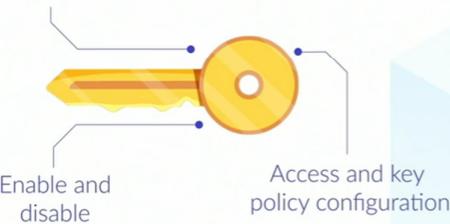
AWS KMS allows you to perform server-side encryption often at the click of a button with minimal configuration required

- Keys created by customers of AWS, using KMS
- Keys managed and created by AWS themselves

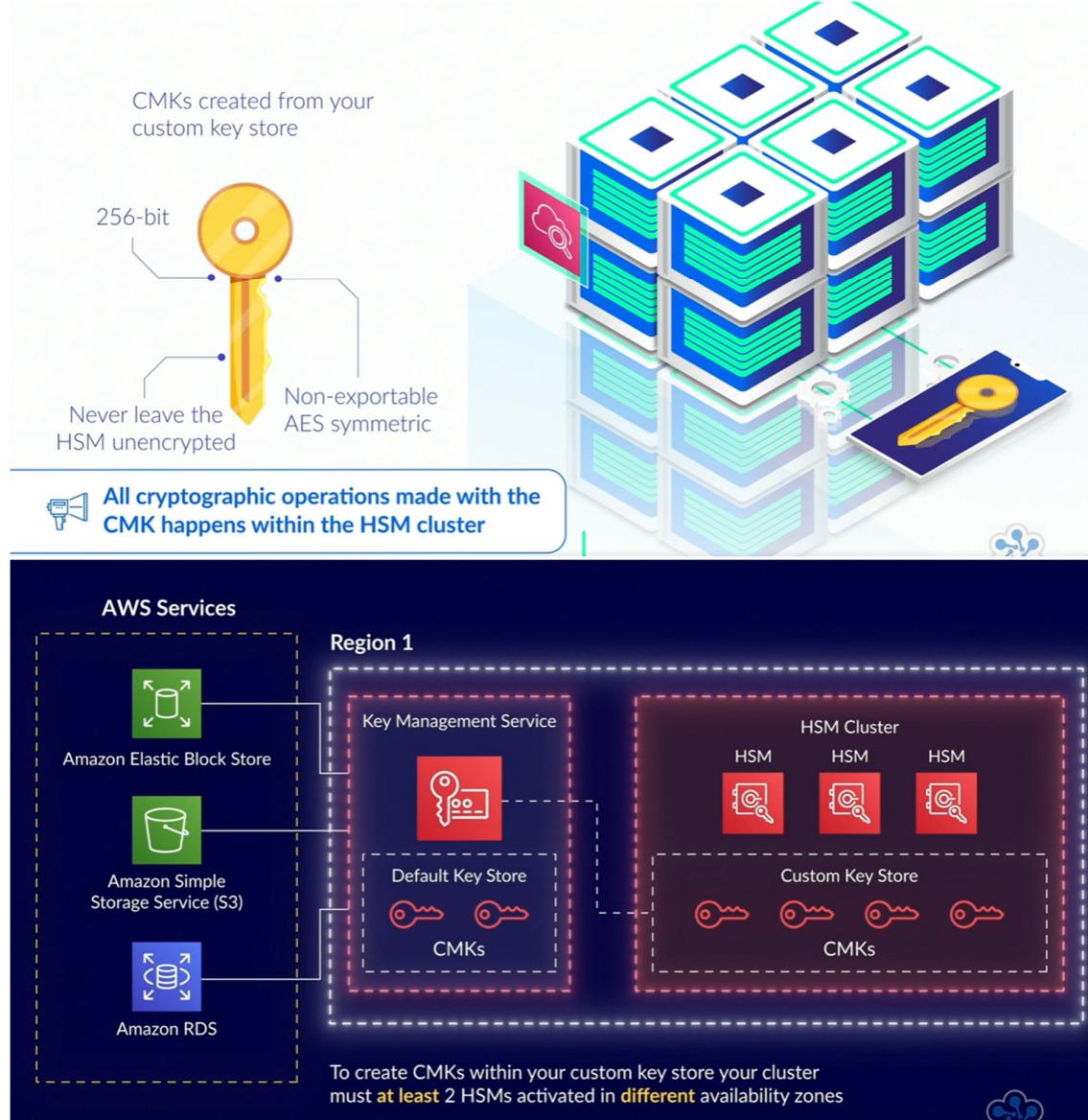


Keys created by customers of AWS, using KMS

Rotation

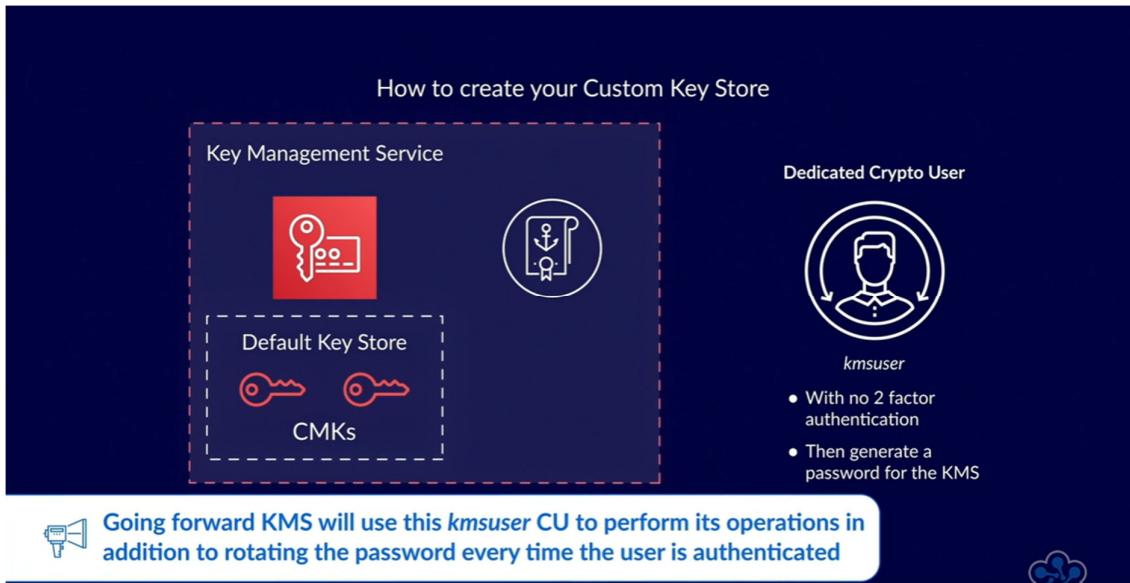


If you want to use seamless integration of KMS with many AWS services, but require security and compliance of storing key material outside of KMS. Then you can create a custom keystore backed by CloudHSM cluster. This is managed from within KMS.



One custom keystore per cluster.

How to create your Custom Key Store



Must upload the Trust Anchor certificate.

AWS Encryption Options

- **Encryption of Data at Rest:**

- You can create an encrypted file system so all your data and metadata is encrypted at rest using an industry-standard AES-256 (Advanced Encryption Standard) encryption algorithm. Encryption and decryption is handled automatically and transparently, so you don't have to modify your applications. If your organization is subject to corporate or regulatory policies that require encryption of data and metadata at rest, we recommend creating an encrypted file system.
- You have the following options for protecting data at rest in Amazon S3:
 - Server-Side Encryption – Request Amazon S3 to encrypt your object before saving it on disks in its data centres and then decrypt it when you download the objects.
 - AWS "Server-side encryption means that if you send unencrypted raw data to AWS, on the AWS infrastructure, the raw data is encrypted and finally stored on disk. When you retrieve data, AWS reads the encrypted data from the disk, decrypts the data, and sends raw data back to you. The encryption /decryption is transparent to the AWS user.
 - Client-Side Encryption – Encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

- **Encryption of Data in Transit:**

- You can mount a file system so all NFS traffic is encrypted in transit using Transport Layer Security 1.2 (TLS, formerly called Secure Sockets Layer [SSL]) with an industry-standard AES-256 cipher. TLS is a set of industry-standard cryptographic protocols used for encrypting information that is exchanged over the wire. AES-256 is a 256-bit encryption cipher used for data transmission in TLS. If your organization is subject to corporate or regulatory policies that require encryption of data and metadata in transit, we recommend setting up encryption in transit on every client accessing the file system.

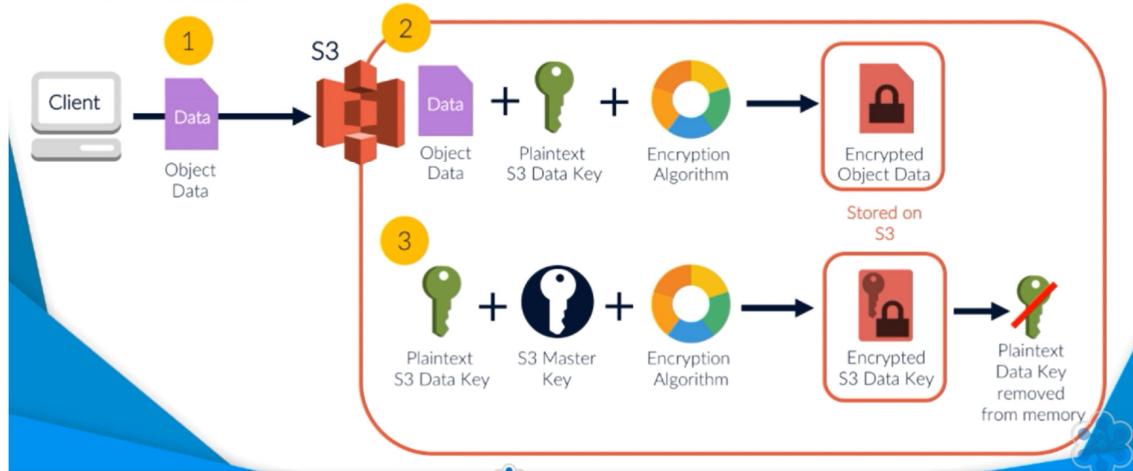
S3 Encryption

Overview of S3 Encryption Mechanisms



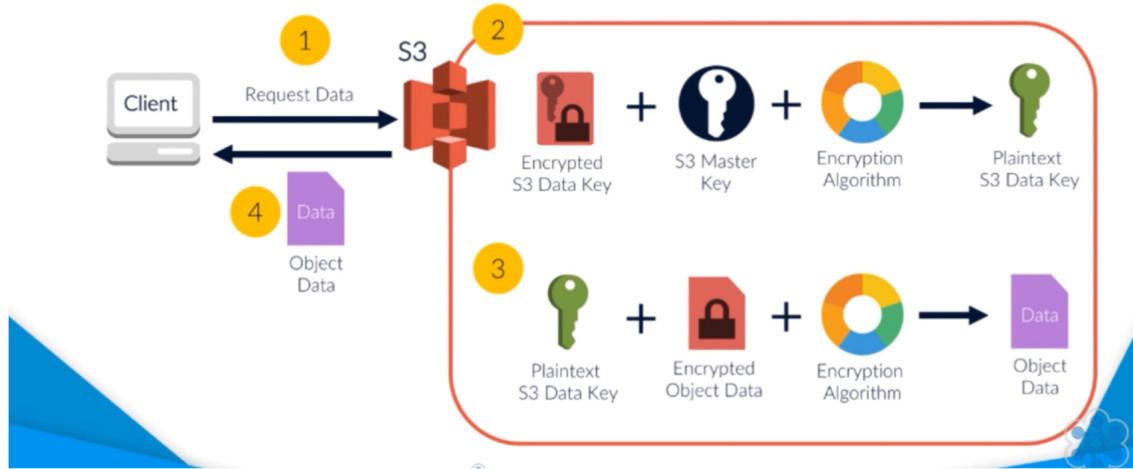
Server-Side Encryption: SSE-S3

Encryption:



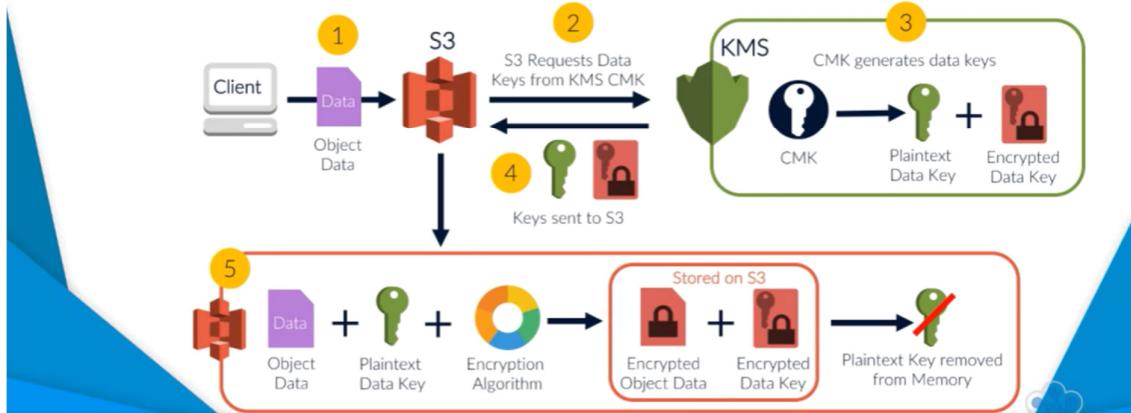
Server-Side Encryption: SSE-S3

Decryption:



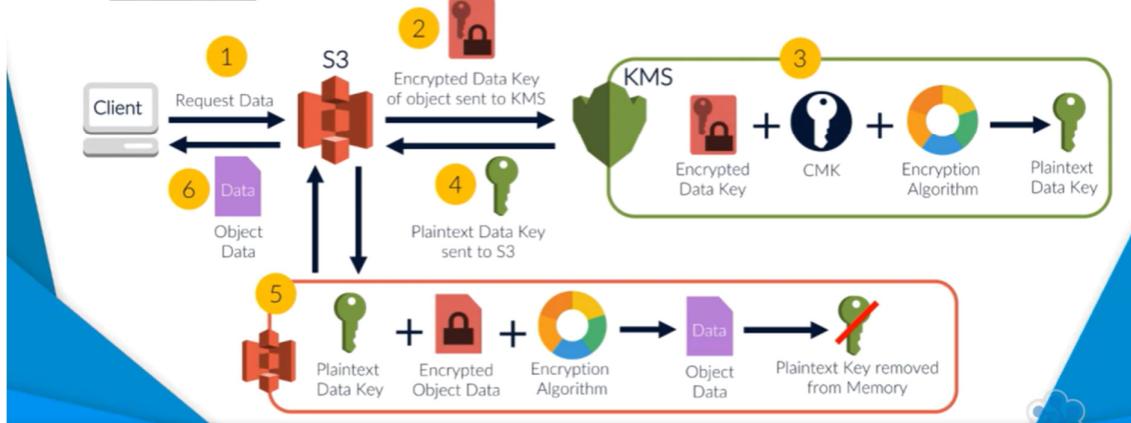
Server-Side Encryption: SSE-KMS

Encryption:



Server-Side Encryption: SSE-KMS

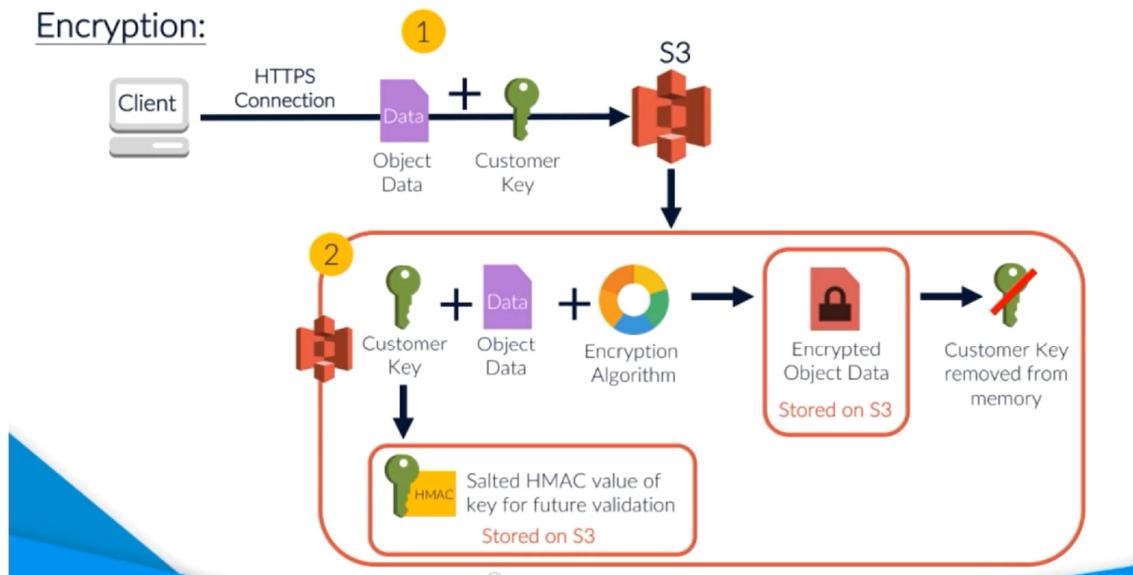
Decryption:



Press Esc to exit full screen

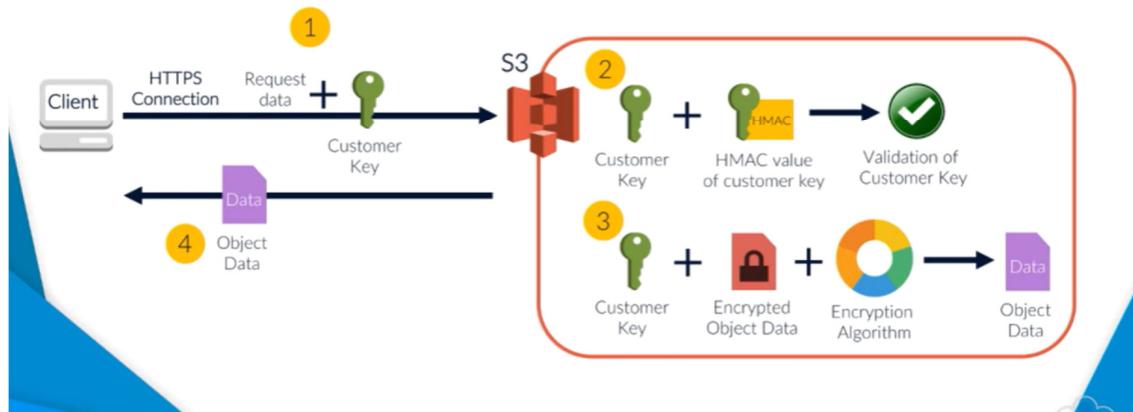
Server-Side Encryption: SSE-C

Encryption:



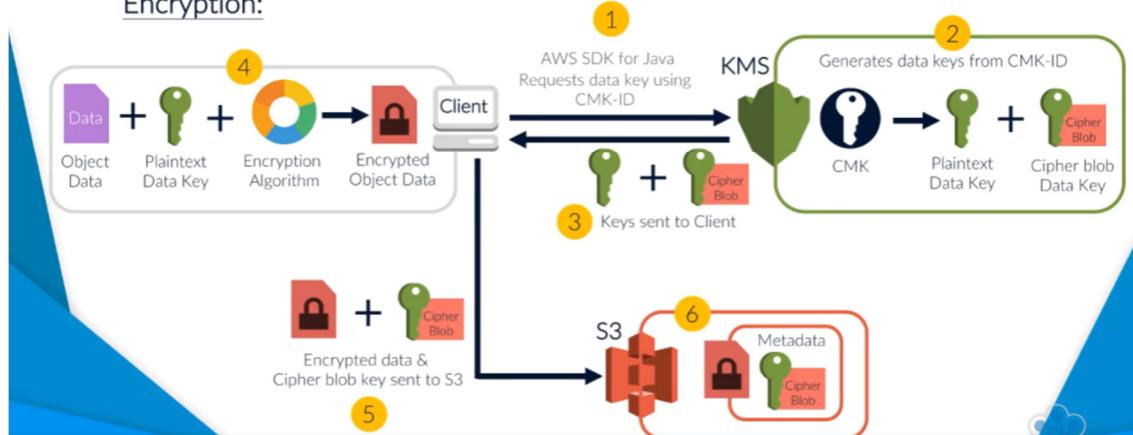
Server-Side Encryption: SSE-C

Decryption:



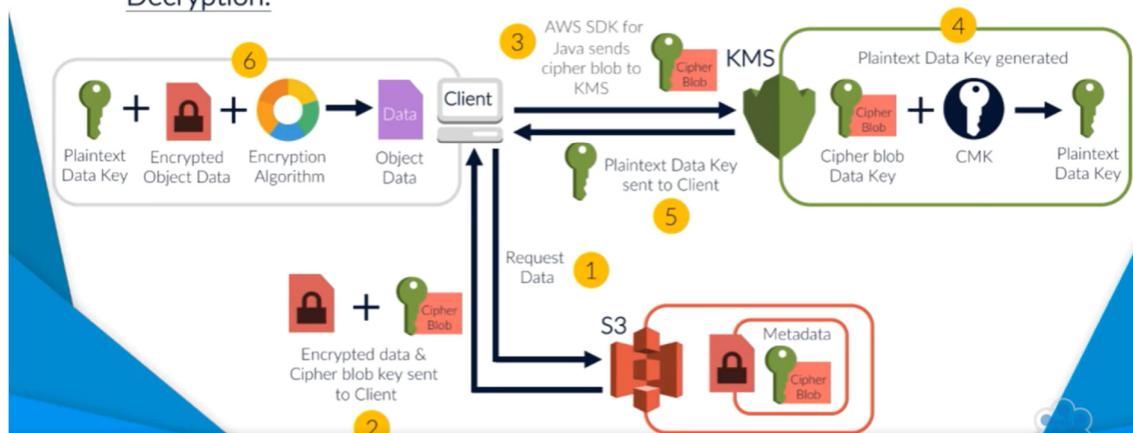
Client-Side Encryption: CSE-KMS

Encryption:

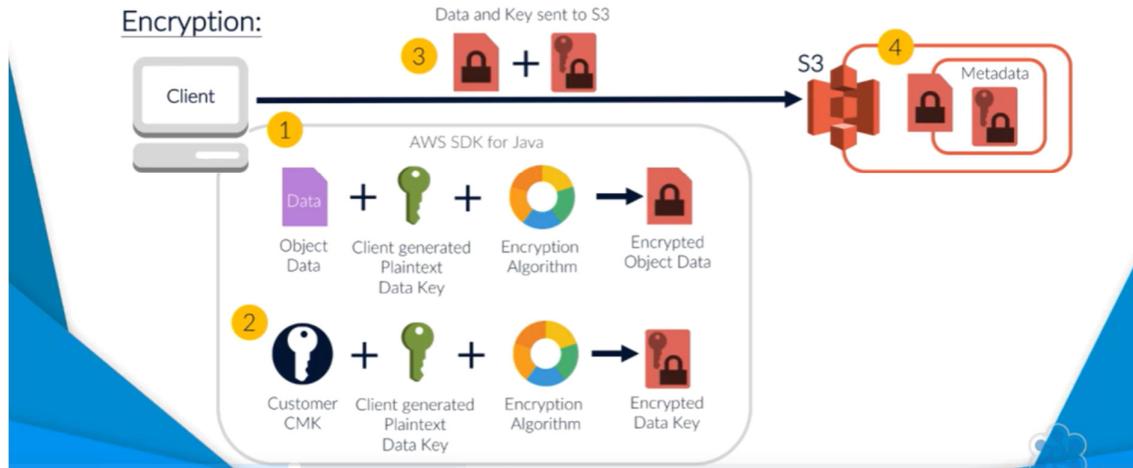


Client-Side Encryption: CSE-KMS

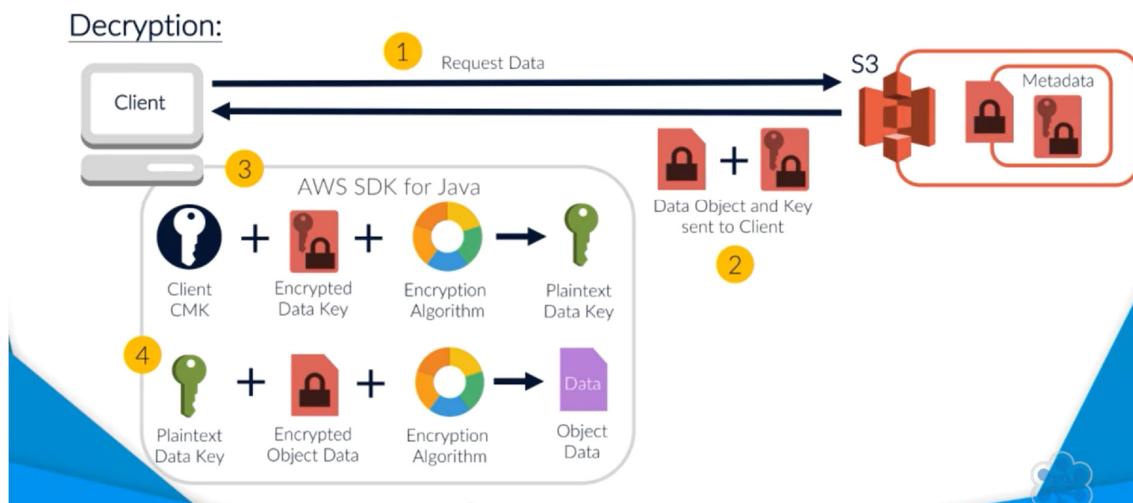
Decryption:



Client-Side Encryption: CSE-C



Client-Side Encryption: CSE-C



Summary



AWS Certificate Manager

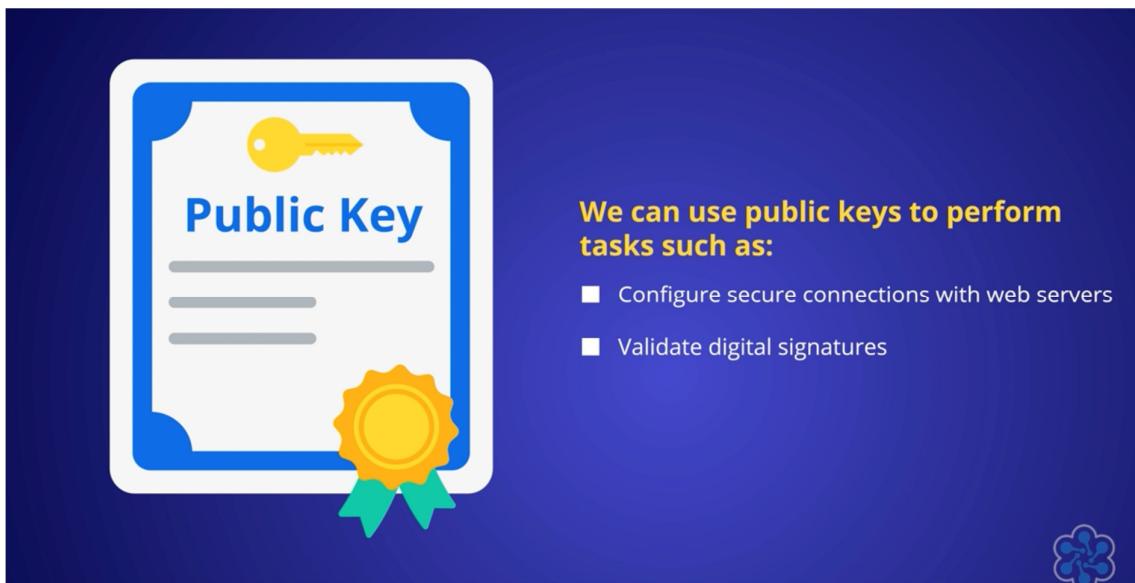
Provision and manage SSL/TLS certificates with AWS services and connected resources. A regional service. Digital certificates allow us to know if the entity we are communicating is valid.



Digital certificates are also used:

- During the authentication of endpoints taking part in site-to-site VPNs
- During the validation of digital signatures
- As part of multi-factor authentication (MFA)
- ...

Digital certificates come embedded with a public key.



We can use public keys to perform tasks such as:

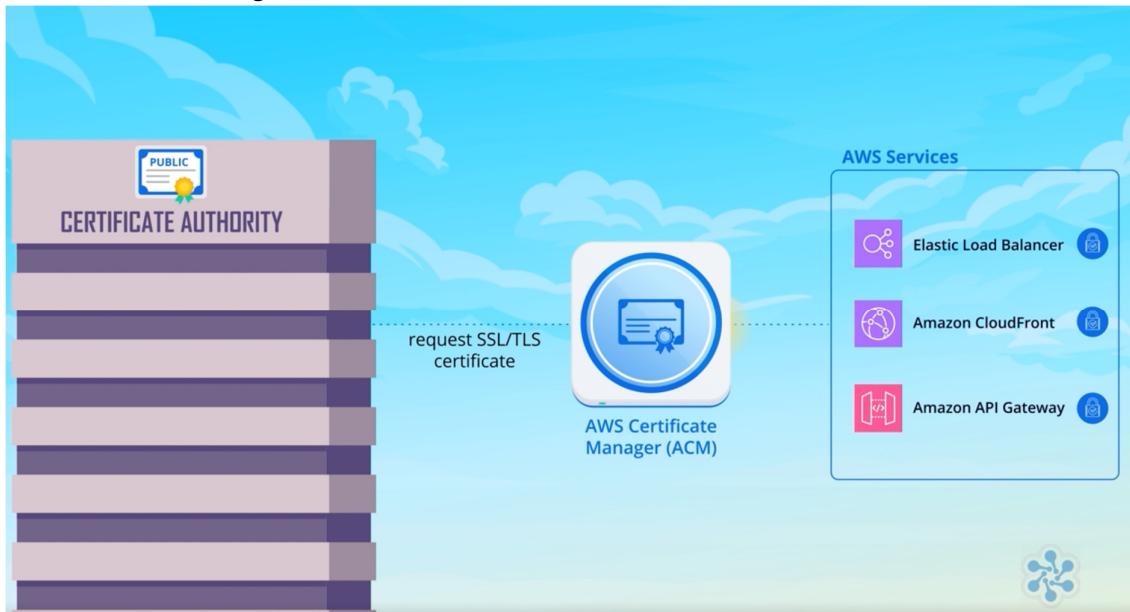
- Configure secure connections with web servers
- Validate digital signatures



We must use Trusted Certificate Authorities (TCAs), can be public or private.

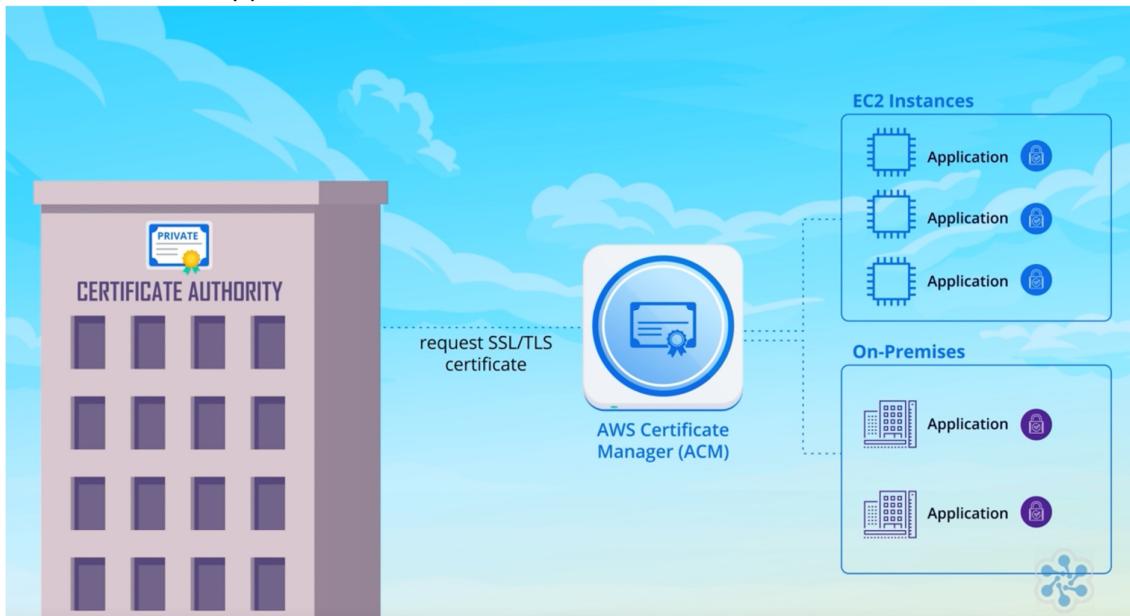


When you request a certificate from a public certificate authority you generate a key-pair at your location. You keep your private key and generate a certificate signing request (CSR), this will contain the public key, the DNS names you wish to secure and your digital signature. Once submitted you must validate you own the domain names specified, then a digital certificate is issued. Certificates issued by public CA's typically are not free. In AWS we can request TLS and SSL certificates for free. AWS Certificate Manager autorenews certificates.



Private Certificate Authorities are not automatically trusted by our browsers and OS's. We must configure trust by importing the root certificate into the certificate store of the OS. Similarly to public certificates a CSR is submitted. For private Certificate Authorities, we deploy the infrastructure including security, backingup, high availability and day-to-day management.

Certificates issued by private CA's are free.





AWS Certificate Manager (ACM)

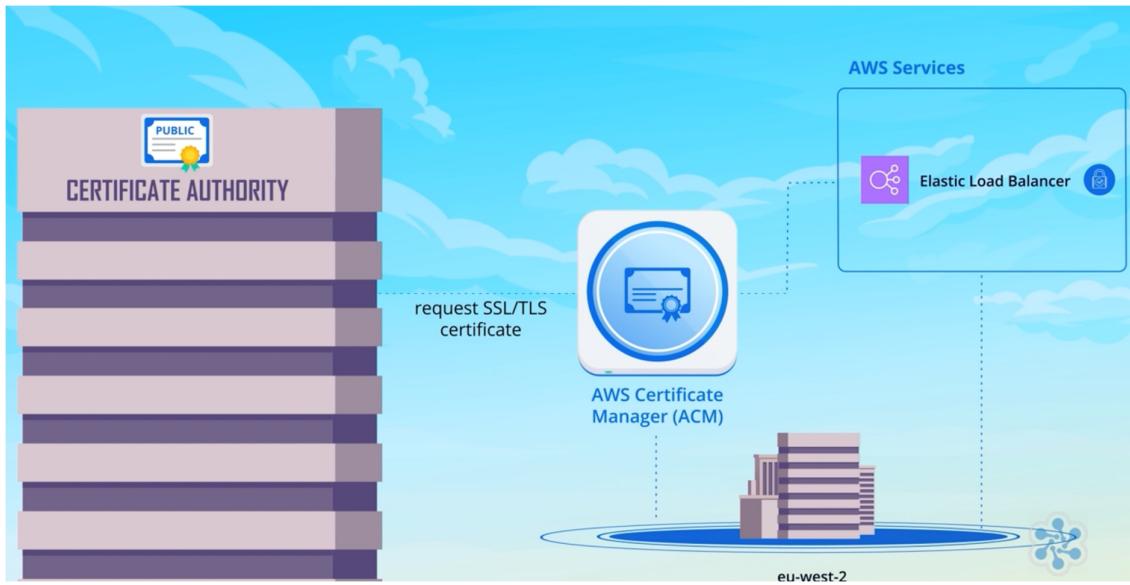
Benefits of AWS Certificate Manager

- Publicly trusted certificates are available for free
- Key pairs and CSRs are created automatically during a certificate request
- AWS is responsible for the HA, backup, and day-to-day management of the servers hosting your CA



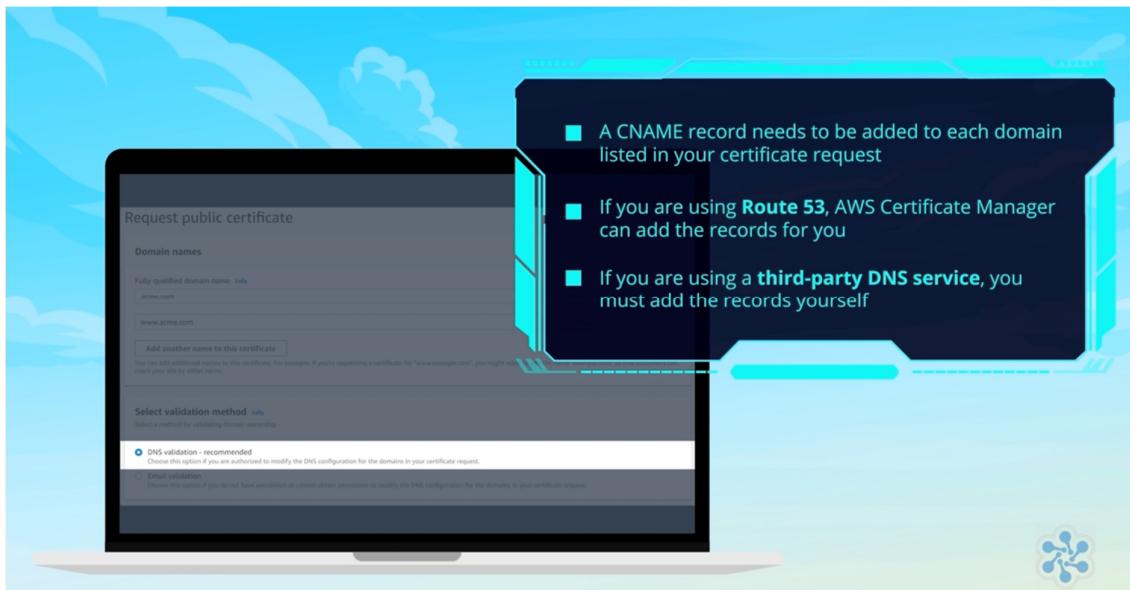
Challenges With Certificate Management

- Managing certificate requests
- Renewing and replacing digital certificates that are due to expire
- The cost of certificates from public CAs
- Securing the certificate authority infrastructure
- Managing certificate revocation lists (CRLs) for your private CAs

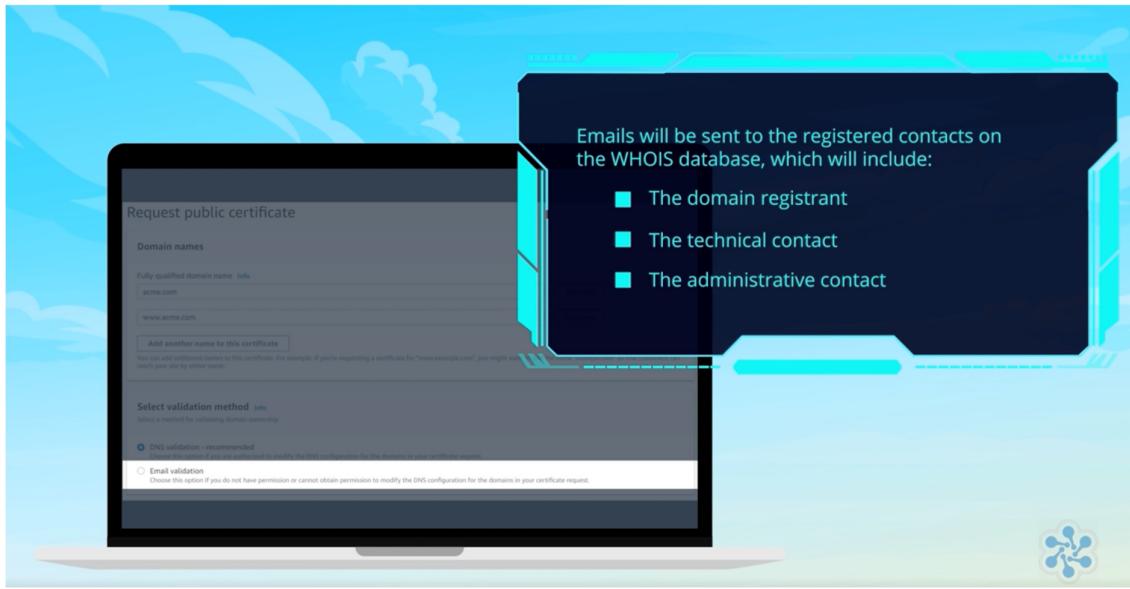


For CloudFront US-East-1 must be used

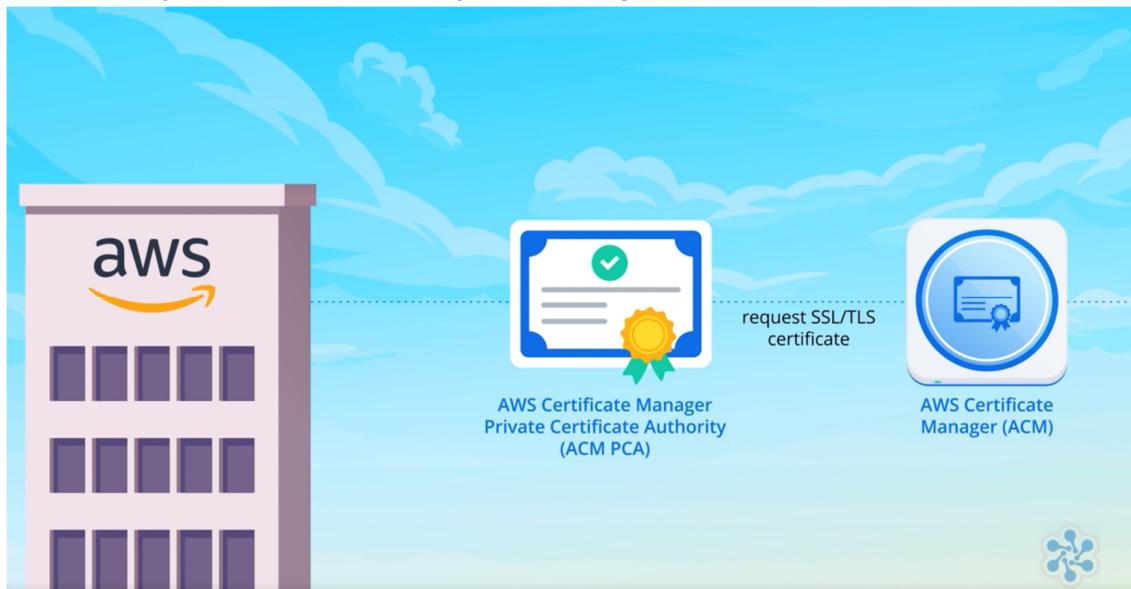
For DNS validation:



For email verification:



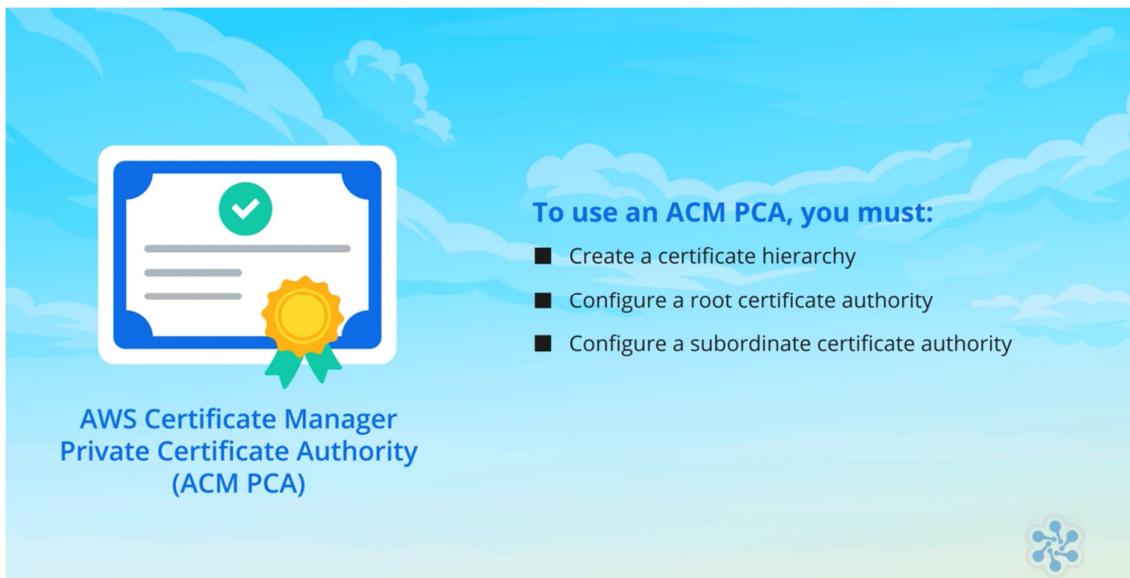
Private Certificates With AWS Certificate Manager



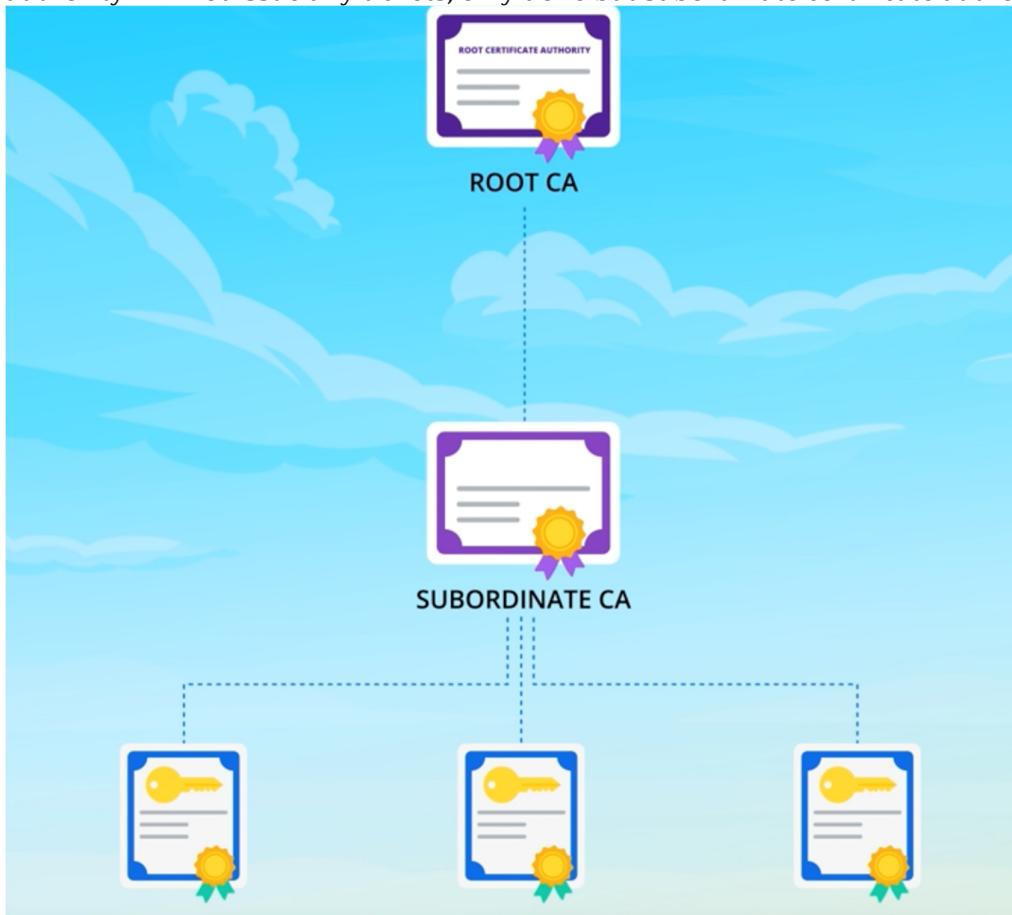
AWS Private CA enables creation of private certificate authority (CA) hierarchies, including root and subordinate CAs, without the investment and maintenance costs of operating an on-premises CA. Your private CAs can issue end-entity X.509 certificates useful in scenarios including:

- Creating encrypted TLS communication channels
- Authenticating users, computers, API endpoints, and IoT devices
- Cryptographically signing code
- Implementing Online Certificate Status Protocol (OCSP) for obtaining certificate revocation status

AWS Private CA operations can be accessed from the AWS Management Console, using the AWS Private CA API, or using the AWS CLI.



Root certificate authority is a self-signed ticket, it is the start of the chain of trust. Any certificates lower in the hierarchy, will be trusted if this is held. The root certificate authority will not issue any tickets, only done by subordinate certificate authorities.



Higher up certificates digitally sign lower certificates to verify them.



**AWS Certificate Manager
Private Certificate Authority
(ACM PCA)**

Use Cases

- Internal applications hosted in AWS or on-premises that require SSL/TLS will need digital certificates issued by a CA
- Certificates intended for internal domains and namespaces that we can't or do not want to validate when requesting a public certificate
- Simplifying certificate authority management by giving day-to-day responsibility to AWS



Monthly fees for each certificate authority you create



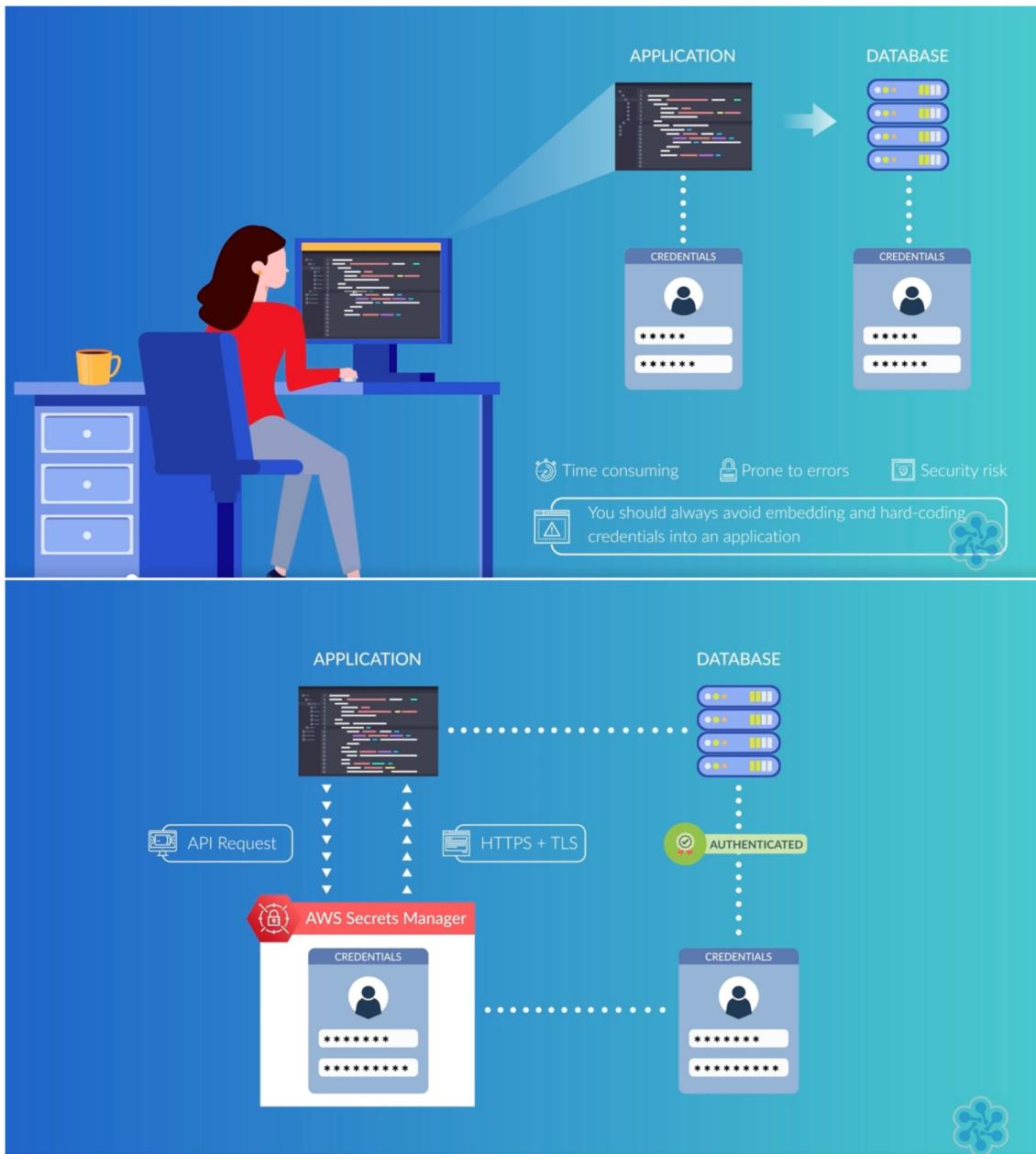
One-off fees for each private certificate that is issued by your PCA



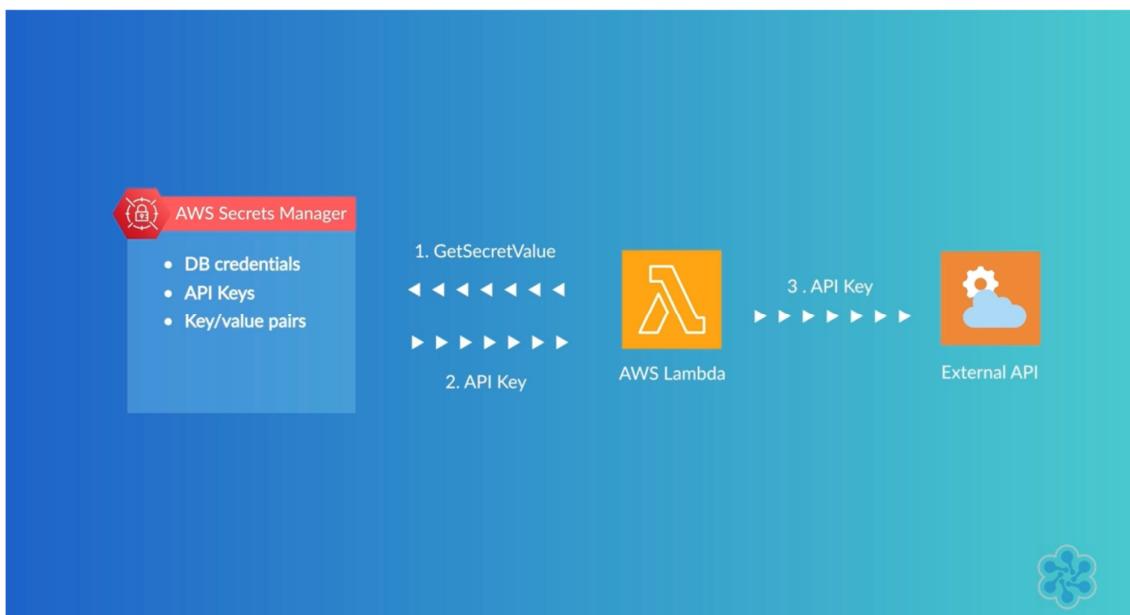

ACM does not manage the renewal process for imported certificates. You are responsible for monitoring the expiration date of your imported certificates and for renewing them before they expire.

AWS Secrets Manager

AWS Secrets Manager helps you manage, retrieve, and rotate database credentials, API keys, and other secrets throughout their lifecycles.

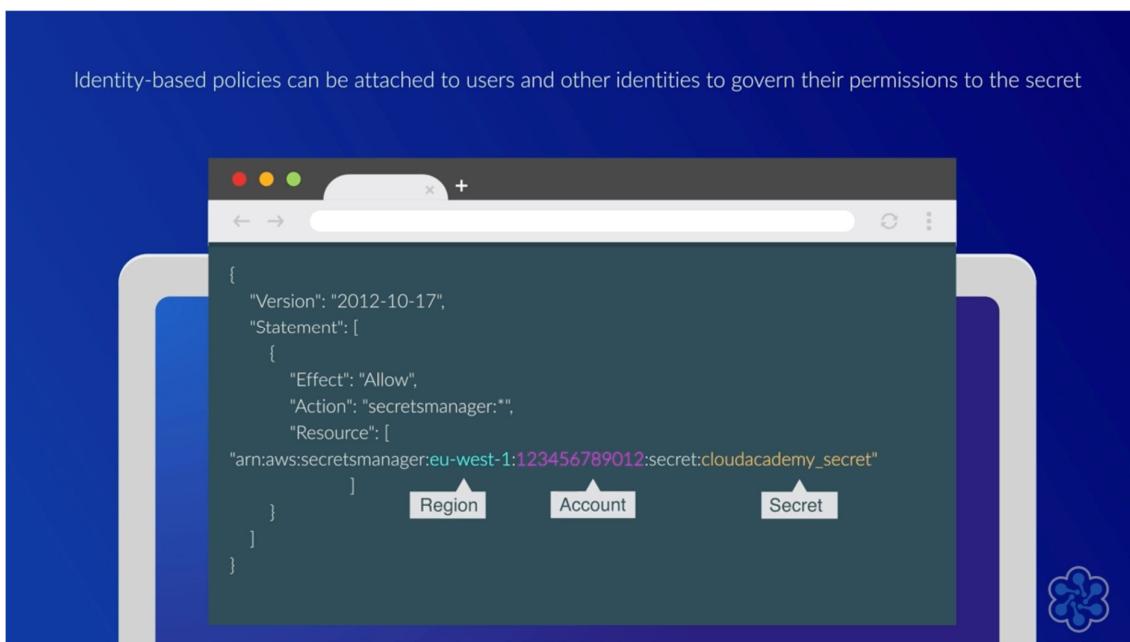


Lambda functions can help rotate passwords.

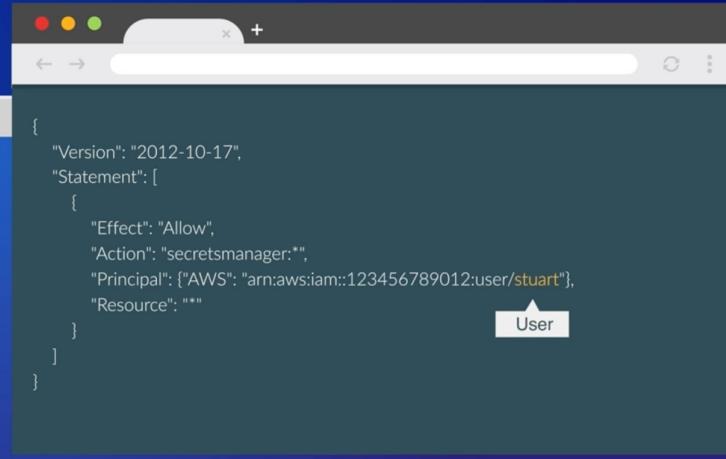


KMS will encrypt your keys.

Access to each secret can be controlled also by IAM policies and resource based policies.

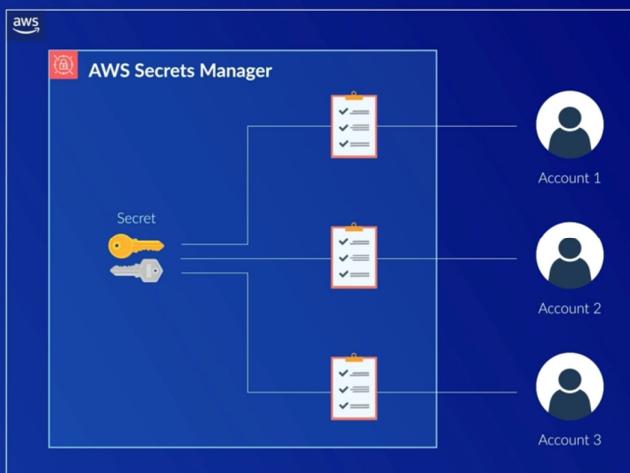


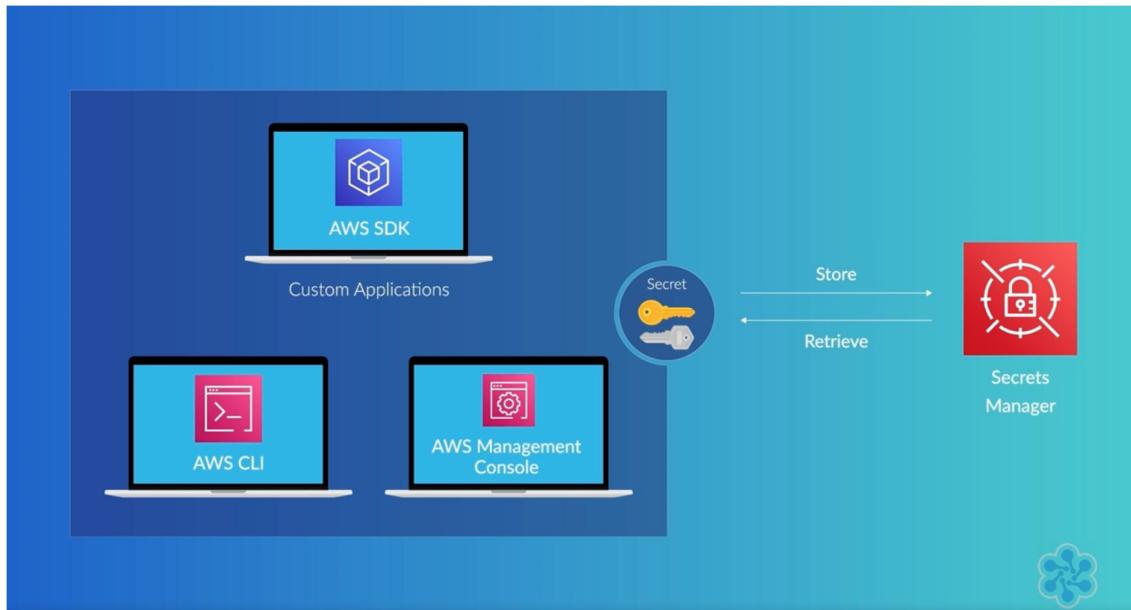
It can be done using the Management Console or the AWS CLI



```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "secretsmanager:",  
            "Principal": ["AWS": "arn:aws:iam::123456789012:user/stuart"],  
            "Resource": "*"  
        }  
    ]  
}
```

Can centralise stores between accounts





Secrets manager supports Autorotation unlike Parameter store.

Administration Services

AWS Organizations

- AWS Organizations helps you centrally manage and govern your environment as you grow and scale your AWS resources. As an administrator of an organization, you can create accounts in your organization and invite existing accounts to join the organization.
- Allows you to:
 - programmatically create new AWS accounts and allocate resources
 - group accounts to organize your workflows
 - apply policies to accounts or groups for governance
 - define central configurations and audit requirements
 - simplify billing by centralising it and using a single payment method for all of your account. These account management and consolidated billing capabilities enable you to better meet the budgetary, security, and compliance needs of your business.
 - control access, manage compliance, coordinate security mechanisms (including restricting the AWS services, resources, and individual API actions accessible by specific users, groups and roles)
 - share resources across your AWS accounts.
 - combine usage from all accounts in the organization to qualify you for volume pricing discounts. If you have multiple standalone accounts, your charges might decrease if you add the accounts to an organization.
- AWS Organizations is tightly integrated with other AWS services
- AWS Organizations is offered at no additional charge. You are charged only for AWS resources that users and roles in your member accounts use.
- Important to not place resources in the master account
- AWS Support plans on the master account of an organization do not automatically apply to member accounts in the organization

Organisation Permissions

Condition keys: AWS provides condition keys that you can query to provide more granular control over certain actions. The following condition keys are especially useful with AWS Organizations:

- aws:PrincipalOrgID – Simplifies specifying the Principal element in a resource-based policy. This global key provides an alternative to listing all the account IDs for all AWS accounts in an organization. Instead of listing all of the accounts that are members of an organization, you can specify the organization ID in the Condition element.
- aws:PrincipalOrgPaths – Use this condition key to match members of a specific organization root, an OU, or its children. The aws:PrincipalOrgPaths condition key returns true when the principal (root user, IAM user, or role) making the request is in the specified organization path. A path is a text representation of the structure of an AWS Organizations entity.

SCPs (Service control policies)

- Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization. SCPs offer central control over the maximum available permissions for all accounts in your organization. SCPs help you to ensure your accounts stay within your organization's access control guidelines. SCPs are available only in an organization that has all features enabled.
- Central security administrators use SCPs with AWS Organizations to establish controls that all IAM principals (users and roles) adhere to. Now, you can use SCPs to set permission guardrails with the fine-grained control supported in the AWS Identity and Access Management (IAM) policy language. This makes it easier for you to fine-tune policies to meet the precise requirements of your organization's governance rules.

Sharing Resources in an Organizations Master Account

- If you would like to share resources with your organization or organizational units, then you must use the AWS RAM console or CLI command to enable sharing with AWS Organizations.
 - The account that originally purchased the Reserved Instance receives the discount first. If the purchasing account doesn't have any instances that match the terms of the Reserved Instance, the discount for the Reserved Instance is assigned to any matching usage on another account in the organization. For billing purposes, the consolidated billing feature of AWS Organizations treats all the accounts in the organization as one account. This means that all accounts in the organization can receive the hourly cost benefit of Reserved Instances that are purchased by any other account.
 - If Reserved Instance sharing is turned off for an account in an organization. Reserved Instance discounts apply only to the account that purchased the Reserved Instance.

Removing Accounts from an Organizations Master Account

Requirements for removing an account from an organization:

- The account that you want to remove must have the information that is required for it to operate as a standalone account
- The account that you want to remove must not be a delegated administrator account for any AWS service enabled for your organization
- Customers' agreements with AWS, and the rights and obligations under those agreements, cannot be assigned or transferred without AWS's prior consent. To obtain AWS's consent, contact us at <https://aws.amazon.com/contact-us/>

How to remove an account from an organization:

- Sign in as an IAM user or role in the management account with the required permissions
 - Go to the 'Accounts' tab and select 'Remove account' for the account you wish to remove
 - AWS will redirect you to the AWS Organizations console for the chosen member account, here select 'Leave organization'
 - Remove the IAM roles that grant access to your member account from the organization.
- Sign in as an IAM user or role in the member account with the required permissions
 - Go to the 'Organization' page and choose 'Leave Organization'
 - Remove the IAM roles that grant access to your account from the organization.

Effects of removing an account from an organization

- When a member account leaves an organization, that account no longer has access to cost and usage data from the time range when the account was a member of the organization. If an account rejoins an organization that it previously belonged to, the account regains access to its historical cost and usage data.
- When a member account leaves an organization, all tags attached to the account are deleted.
- The account is now responsible for paying its own charges and must have a valid payment method attached to the account.
- The principals in the account are no longer affected by any policies that applied in the organization. This means that restrictions imposed by SCPs are gone, and the users and roles in the account might have more permissions than they had before. Other organization policy types can no longer be enforced or processed.
- Integration with other services might be disabled. For example, AWS Single Sign-On (SSO) requires an organization to operate, so if you remove an account from an organization that supports AWS SSO, the users in that account can no longer use that service.

[AWS Service Catalog](#)

- AWS Service Catalog allows organizations to create and manage catalogues of IT services that are approved for use on AWS. These IT services can include everything from virtual machine images, servers, software, and databases to complete multi-tier application architectures. AWS Service Catalog allows you to centrally manage deployed IT services and your applications, resources, and metadata. This helps you achieve consistent governance and meet your compliance requirements, while enabling users to quickly deploy only the approved IT services they need.
- With AWS Service Catalog AppRegistry, organizations can understand the application context of their AWS resources. You can define and manage your applications and their metadata, to keep track of cost, performance, security, compliance and operational status at the application level.
- AWS Service Catalog Delivery Partners are APN Consulting Partners who help create catalogues of IT services that are approved by the customer's organization for use on AWS. With AWS Service Catalog, customers and partners can centrally manage commonly deployed IT services to help achieve consistent governance and meet compliance requirements while enabling users to self-provision approved services.

[AWS Systems Manager](#)

- gives you visibility and control of your infrastructure on AWS. Provides a unified user interface so you can view operational data from multiple AWS services and allows you to automate operational tasks across your AWS resources.
- you can group resources, like Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources.
- Systems Manager simplifies resource and application management, shortens the time to detect and resolve operational problems, and makes it easy to operate and manage your infrastructure securely at scale.
- AWS Systems Manager Maintenance Windows let you define a schedule for when to perform potentially disruptive actions on your instances such as patching an operating system, updating drivers, or installing software or patches.
 - Use Maintenance Windows to set up recurring schedules for managed instances to run administrative tasks like installing patches and updates without interrupting business-critical operations.

[AWS Systems Manager Session Manager](#)

Session Manager is a fully managed AWS Systems Manager capability. With Session Manager, you can manage your Amazon Elastic Compute Cloud (Amazon EC2) instances, edge devices, on-premises servers, and virtual machines (VMs). You can use either an interactive one-click browser-based shell or the AWS Command Line Interface (AWS CLI). Session Manager provides secure and auditable node management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. Session Manager also allows you to comply with corporate policies that require controlled

access to managed nodes, strict security practices, and fully auditable logs with node access details, while providing end users with simple one-click cross-platform access to your managed nodes.

AWS Systems Manager Patch Manager is designed to apply patches not only to the operating system but also to third-party software running on Amazon EC2 instances, on-premises servers, and virtual machines. It allows you to manage and automate the process of patching both operating systems and applications, including third-party applications so using the patch manager and scheduling a maintenance window, you can ensure controlled and coordinated patching of the EC2 instances. This helps in minimizing disruptions and managing the process effectively.

AWS Systems Manager Run Command allows the company to run commands or scripts on multiple EC2 instances. By using Run Command, the company can quickly and easily apply the patch to all 1,000 EC2 instances to remediate the security vulnerability.

AWS Service Health Dashboard

Displays the general status of AWS services. The dashboard provides access to current status and historical data about each and every Amazon Web Service. If there's a problem with a service, you'll be able to expand the appropriate line in the Details section. You can even subscribe to the RSS feed for any service. You can use the "Report an Issue" link to make sure that we are aware of any system-wide service issues. You will be able to see a record of service status, on a per-service basis, for the previous 35 days.

AWS Health

Provides ongoing visibility into your resource performance and the availability of your AWS services and accounts. You can use AWS Health events to learn how service and resource changes might affect your applications running on AWS. AWS Health provides relevant and timely information to help you manage events in progress. AWS Health also helps you be aware of and to prepare for planned activities. The service delivers alerts and notifications triggered by changes in the health of AWS resources, so that you get near-instant event visibility and guidance to help accelerate troubleshooting. Additionally, AWS Support customers who have a Business or Enterprise support plan can use the AWS Health API to integrate with in-house and third-party systems.

AWS Personal Health Dashboard (PHD)

All customers can use this, it is powered by the AWS Health API. A personalized view of the health of AWS services, and alerts when your resources are impacted. It provides alerts and remediation guidance when AWS is experiencing events that may impact you. Personal Health Dashboard gives you a personalized view into the performance and availability of the AWS services underlying your AWS resources. The dashboard requires no setup, and it's ready to use for authenticated AWS users.

AWS Personal Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact the performance and availability of the AWS services underlying your AWS resources.

The dashboard displays relevant and timely information to help you manage events in progress, and provides proactive notification to help you plan for scheduled activities. With Personal Health Dashboard, alerts are triggered by changes in the health of AWS resources, giving you event visibility, and guidance to help quickly diagnose and resolve issues.

Detailed troubleshooting guidance - When you get an alert, it includes remediation details and specific guidance to enable you to take immediate action to address AWS events impacting your resources. For example, in the event of an AWS hardware failure impacting one of your Amazon Elastic Block Store (EBS) volumes, your alert would include a list of your affected resources, a recommendation to restore your volume, and links to the steps to help you restore it from a snapshot. This targeted and actionable information reduces the time needed to resolve issues.

If you use AWS Organizations, AWS Health allows you to aggregate notifications from all accounts in your organization. This provides centralized and real-time access to all AWS Health events posted to individual accounts in your organization, including operational issues, scheduled maintenance, and account notifications.

Integration and automation - AWS Personal Health Dashboard can integrate with Amazon CloudWatch Events, enabling you to build custom rules and select targets such as AWS Lambda functions to define automated remediation actions. The AWS Health API, the underlying service powering Personal Health Dashboard, allows you to integrate health data and notifications with your existing in-house or third party IT Management tools.

AWS Personal Health Dashboard is available to all AWS customers, and provides status and notifications for all services across all Regions and Availability Zones. Access to the AWS Health API is included as part of the AWS Business Support and AWS Enterprise Support plans.

AWS Personal Health Dashboard gives you fine-grained access control so that you can setup permissions based on event metadata. This allows you to grant or deny access to an AWS Identity and Access Management (IAM) user based on such attributes as event types, event types of a particular service, or other role-based attributes. With fine-grained access control, you can limit access of sensitive alerts, such as those related to security, to only those users that need to see them.

Difference between AWS Service Health Dashboard and AWS Personal Health Dashboard

While the Service Health Dashboard displays the GENERAL status of AWS services, Personal Health Dashboard gives you a PERSONALIZED VIEW into the performance and availability of the AWS services underlying your AWS resources.

The difference between Personal and Health dashboards is "Health" provides the "generic status of overall AWS services and on in particular. But "Personal" provides status of services pertaining to "subscribed" AWS services. This is why it is a "Personal" health dashboard

[AWS Well-Architected Tool](#)

The AWS Well-Architected Tool (AWS WA Tool) is a service in the cloud that provides a consistent process for you to review and measure your architecture using the AWS Well-Architected Framework. The AWS WA Tool provides recommendations for making your workloads more reliable, secure, efficient, and cost-effective.

[AWS License Manager](#)

A service which helps you manage your software licenses, including Microsoft Windows Server and Microsoft SQL Server licenses. In License Manager, you can specify your licensing terms for governing license usage, as well as your Dedicated Host management preferences for host allocation and host capacity utilization. Once setup, AWS takes care of these administrative tasks on your behalf, so that you can seamlessly launch virtual machines (instances) on Dedicated Hosts just like you would launch an EC2 instance with AWS provided licenses.

[AWS CloudFormation](#)

- Speed up cloud provisioning with infrastructure as code. Gives you an easy way to model a collection of related AWS and third-party resources, provision them quickly and consistently, and manage them throughout their lifecycles, by treating infrastructure as code (IaC). A CloudFormation template describes your desired resources and their dependencies so you can launch and configure them together as a stack. You can use a template to create, update, and delete an entire stack as a single unit, as often as you need to, instead of managing resources individually. You can manage and provision stacks across multiple AWS accounts and AWS Regions. The CloudFormation template acts as a “*single source of truth*” for an AWS cloud environment.

[AWS Quick Starts](#)

AWS Quick Starts are built by AWS solutions architects and partners to help you deploy popular technologies on AWS, based on AWS best practices for security and high availability. These accelerators reduce hundreds of manual procedures into just a few steps, so you can build your production environment quickly and start using it immediately.

AWS Control Tower

- Automates the process of setting up a new baseline multi-account AWS environment that is secure, well-architected, and ready to use.
- If you're an enterprise with multiple AWS accounts and teams, cloud setup and governance can be complex and time consuming, slowing down the very innovation you're trying to speed up. AWS Control Tower provides the easiest way to set up and govern a new, secure, multi-account AWS environment based on best practices established through AWS' experience working with thousands of enterprises as they move to the cloud. With AWS Control Tower, builders can provision new AWS accounts in a few clicks, while you have peace of mind knowing your accounts conform to your company-wide policies. If you are building a new AWS environment, starting out on your journey to AWS, starting a new cloud initiative, or are completely new to AWS, Control Tower will help you get started quickly with governance and best practices built-in.
- A landing zone is a well-architected, multi-account AWS environment that's based on security and compliance best practices. AWS Control Tower automates the setup of a new landing zone using best-practices blueprints for identity, federated access, and account structure.

AWS Auto Scaling

Not to be confused with EC2 Auto Scaling.

AWS Auto Scaling – Focusses on Amazon ECS, DynamoDB and Amazon Aurora

AWS Auto Scaling monitors your services and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.

The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them.

AWS Auto Scaling is available at no additional charge. You pay only for the AWS resources needed to run your applications and Amazon CloudWatch monitoring fees.

AWS CloudTrail

- Track user activity and API usage. Helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

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 continuously monitors and records your AWS resource configurations. It can detect drift and trigger AWS Systems Manager Automation to fix it and raise alarms.

Amazon CloudWatch

- Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers.
- CloudWatch 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.
- 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.
- CloudWatch collects monitoring and operational data in the **form of logs, metrics, and events**, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers.

With Basic monitoring you get data on your Cloudwatch metrics every 5 minutes. Enabling detailed monitoring, you will get the data every one minute.

[Amazon CloudWatch Events](#)

- Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services AWS resources. Using simple rules that you can quickly set up, you can match events and route them to one or more target functions or streams. CloudWatch Events becomes aware of operational changes as they occur. CloudWatch Events responds to these operational changes and takes corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information.
- You can also use CloudWatch Events to schedule automated actions that self-trigger at certain times using cron (The origin of the name cron is from the Greek word for time, χρόνος (chronos)) or rate expressions.
- The following services are used in conjunction with CloudWatch Events: AWS CloudTrail, AWS CloudFormation, AWS Config, AWS Identity and Access Management (IAM), Amazon Kinesis Data Streams and AWS Lambda

[Amazon Eventsbridge](#)

Amazon EventBridge is a service that provides real-time access to changes in data in AWS services, your own applications and Software-as-a-Service (SaaS) applications without writing code. To get started, you can choose an event source on the Amazon EventBridge console, and select a target from a number of AWS services including AWS Lambda, Amazon SNS, and Amazon Kinesis Data Firehose. Amazon EventBridge will automatically deliver the events in near real-time.

Amazon EventBridge builds upon and extends CloudWatch Events. It uses the same service API and endpoint, and the same underlying service infrastructure. However, it has new features also that enable customers to connect data from their own apps and third-party SaaS apps

Amazon CloudWatch Logs

- You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.
- CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service. Log data can be stored and accessed indefinitely in highly durable, low-cost storage so you don't have to worry about filling up hard drives
- You can then easily view them, monitor your logs, in NEAR real-time for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis. CloudWatch Logs enables you to see all of your logs, regardless of their source, as a single and consistent flow of events ordered by time, and you can query them and sort them based on other dimensions, group them by specific fields, create custom computations with a powerful query language, and visualize log data in dashboards. You can also view the original log data to see the source of the problem
 - CloudWatch Logs Additional Features:
 - Log Retention – By default, logs are kept indefinitely and never expire. You can adjust the retention policy for each log group, keeping the indefinite retention, or choosing a retention period between 10 years and one day.
 - Archive Log Data
 - Log Route 53 DNS Queries

You can configure a CloudWatch Logs log group to stream data it receives to your Amazon OpenSearch Service cluster in NEAR REAL-TIME through a CloudWatch Logs subscription.

[Amazon CloudWatch Alarms](#)

- allow you to set a threshold on metrics and trigger an action. You can create high-resolution alarms, set a percentile as the statistic, and either specify an action or ignore as appropriate. For example, you can create alarms on a variety of factors, such as Amazon EC2 metrics or ECS clusters scaling up or down; set notifications; and take one or more actions, such as detecting and shutting down unused or underutilized instances. Real-time alarming on metrics and events enables you to minimize downtime and potential business impact.
- Alarms invoke actions for sustained state changes only. CloudWatch alarms do not invoke actions simply because they are in a particular state. The state must have changed and been maintained for a specified number of periods.
- A composite alarm includes a rule expression that takes into account the alarm states of other alarms that you have created. The composite alarm goes into ALARM state only if all conditions of the rule are met. The alarms specified in a composite alarm's rule expression can include metric alarms and other composite alarms. Using composite alarms can reduce alarm noise. You can create multiple metric alarms, and also create a composite alarm and set up alerts only for the composite alarm. For example, a composite might go into ALARM state only when all of the underlying metric alarms are in ALARM state.

AWS Directory Service

- AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft Active Directory (AD), enables your directory-aware workloads and AWS resources to use managed Active Directory (AD) in AWS.
- AWS Managed Microsoft AD is built on actual Microsoft AD and does not require you to synchronize or replicate data from your existing Active Directory to the cloud. You can use the standard AD administration tools and take advantage of the built-in AD features, such as Group Policy and single sign-on. With AWS Managed Microsoft AD, you can easily join Amazon EC2 and Amazon RDS for SQL Server instances to your domain, and use AWS End User Computing (EUC) services, such as Amazon WorkSpaces, with AD users and groups.
- AWS Managed Microsoft AD makes it easy to extend your existing Active Directory to AWS. It enables you to leverage your existing on-premises user credentials to access cloud resources such as the AWS Management Console, Amazon Workspaces, Amazon Chime, and Windows workloads in the cloud.
- Enable your users to enable single sign-on (SSO) to the AWS Console. This enables your users to sign in with their existing AD credentials, assume one of their assigned roles at sign-in, and to access and take action on the resources according to the permissions defined for the role.

AWS 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.
- You model your application as a stack, consisting of various layers. These layers are like blueprints detailing how to setup and configure a set of EC2 instances and related resources. There are prebuilt layers for common components. “Chef recipes” detail your layout and configuration. Automatically and manually scalable. Essentially opsworks automates your infrastructure deployment.
- OpsWorks comes at no additional cost, you pay only for the resources and services you use to run your applications.
- OpsWorks has three offerings, AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.
 - **AWS OpsWorks for Chef Automate** provides a fully managed Chef Automate server and suite of automation tools that give you workflow automation for continuous deployment, automated testing for compliance and security, and a user interface that gives you visibility into your nodes and their status. The Chef Automate platform gives you full stack automation by handling operational tasks such as software and operating system configurations, continuous compliance, package installations, database setups, and more. The Chef server centrally stores your configuration tasks and provides them to each node in your compute environment at any scale, from a few nodes to thousands of nodes. OpsWorks for Chef Automate is completely compatible with tooling and cookbooks from the Chef community and automatically registers new nodes with your Chef server.
 - **AWS OpsWorks for Puppet Enterprise** is a fully managed configuration management service that hosts Puppet Enterprise, a set of automation tools from Puppet for infrastructure and application management. OpsWorks also maintains your Puppet master server by automatically patching, updating, and backing up your server. OpsWorks eliminates the need to operate your own configuration management systems or worry about maintaining its infrastructure. OpsWorks gives you access to all of the Puppet Enterprise features, which you manage through the Puppet console. It also works seamlessly with your existing Puppet code.

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. You can deploy and configure Amazon EC2 instances in each layer or connect other resources such as Amazon RDS databases. OpsWorks Stacks lets you set automatic scaling for your servers based on preset schedules or in response to changing traffic levels, and it uses lifecycle hooks to orchestrate changes as your environment scales. You run Chef recipes using Chef Solo, allowing you to automate tasks such as installing packages and programming languages or frameworks, configuring software, and more.

AWS X-Ray

- Helps developers analyse and debug production, distributed applications, such as those built using a microservices architecture. With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors.
- provides an end-to-end view of requests as they travel through your application, and shows a map of your application's underlying components. You can use X-Ray to analyse both applications in development and in production, from simple three-tier applications to complex microservices applications consisting of thousands of services.

AWS Compliance

Enables customers to understand the robust controls in place at AWS to maintain security and data protection in the cloud. As systems are built on top of AWS cloud infrastructure, compliance responsibilities are shared. By tying together governance-focused, audit friendly service features with applicable compliance or audit standards, AWS Compliance enables build on traditional programs; helping customers to establish and operate in an AWS security control environment. The IT infrastructure that AWS provides to its customers is designed and managed in alignment with security best practices and a variety of IT security standards, including:

- System and Organization Controls (SOC) Reports - independent third-party examination reports that demonstrate how AWS achieves key compliance controls and objectives. The purpose of these reports is to help you and your auditors understand the AWS controls established to support operations and compliance
 - SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70)
 - SOC 2 - Security, Availability & Confidentiality Report
 - SOC 2 - Privacy Type I Report
 - SOC 3 - Security, Availability & Confidentiality Report
- FISMA, DIACAP, and FedRAMP
- DOD CSM Levels 1-5
- PCI DSS Level 1
- ISO 9001 / ISO 27001 / ISO 27017 / ISO 27018
- ITAR
- FIPS 140-2
- MTCS Level 3
- HITRUST

In addition, the flexibility and control that the AWS platform provides allows customers to deploy solutions that meet several industry-specific standards, including:

- Criminal Justice Information Services (CJIS)
- Cloud Security Alliance (CSA)
- Family Educational Rights and Privacy Act (FERPA)
- Health Insurance Portability and Accountability Act (HIPAA)
- Motion Picture Association of America (MPAA)

AWS Artifact

A no cost, self-service portal for on-demand access to for compliance-related information that matters to AWS customers. It provides on-demand access to AWS' security and compliance reports and select online agreements. Reports available in AWS Artifact include our Service Organization Control (SOC) reports, Payment Card Industry (PCI) reports, and certifications from accreditation bodies across geographies and compliance verticals that validate the implementation and operating effectiveness of AWS security controls. Agreements available in AWS Artifact include the Business Associate Addendum (BAA) and the Nondisclosure Agreement (NDA). The AWS SOC 2 report is particularly helpful for completing questionnaires because it provides a comprehensive description of the implementation and operating effectiveness of AWS security controls. Another useful document is the Executive Briefing within the AWS FedRAMP Partner Package.

Software & Application Development Services

AWS CodeStar Services

- enables you to quickly develop, build, and deploy applications on AWS.
- provides a unified user interface, enabling you to easily manage your software development activities in one place. You can set up your entire continuous delivery toolchain in minutes, allowing you to start releasing code faster. Makes it easy for your whole team to work together securely, allowing you to easily manage access and add owners, contributors, and viewers to your projects.
- Each AWS CodeStar project comes with a project management dashboard, including an integrated issue tracking capability powered by Atlassian JIRA Software. With the AWS CodeStar project dashboard, you can easily track progress across your entire software development process, from your backlog of work items to teams' recent code deployments.
- Related AWS Code services are:
 - CodeCommit - A secure and scalable source/version control service supporting Git workflows
 - CodePipeline - A service for fast and reliable continuous integration (CI) and continuous delivery (CD)
 - CodeBuild - A scalable service to compile, test, and package source code
 - CodeDeploy - A service to automate code deployments anywhere

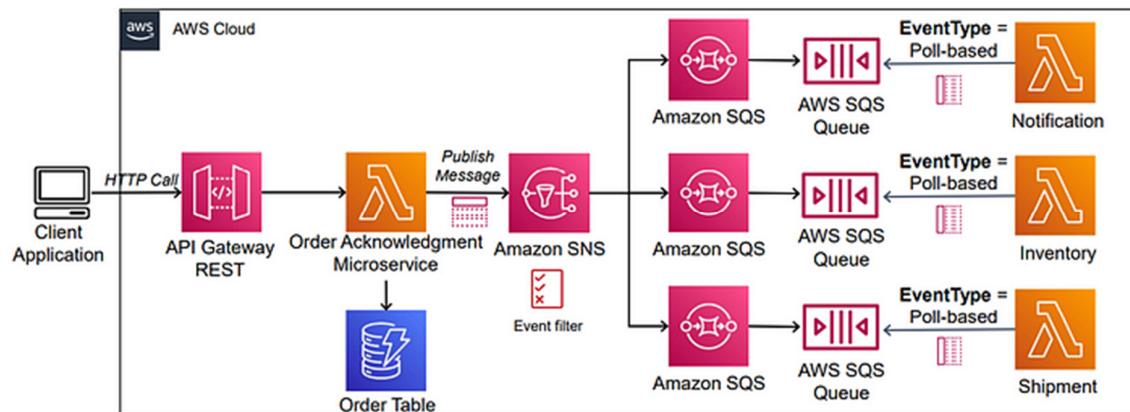
AWS Step Functions

- a serverless function orchestrator that makes it easy to sequence AWS Lambda functions and multiple AWS services into business-critical applications.
- Step Functions allow us to design and build the flow of execution of AWS serverless modules in our application in a simplified manner
- Through its visual interface, you can create and run a series of checkpointed and event-driven workflows that maintain the application state.
- The output of one step acts as an input to the next. Each step in your application executes in order, as defined by your business logic...
- This enables a developer to focus solely on ensuring that each module performs its intended task, without having to worry about connecting each module with others.

Amazon Simple Queue Service (SQS)

- is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work.
- Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.
- SQS offers two types of message queues:
 - Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery.
 - SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent

Fan-Out Pattern in Serverless Architectures Using SNS, SQS and Lambda.



Fan-out is a messaging pattern where piece of message is distributed or ‘fanned out’ to multiple destination in parallel. The main idea is each of destinations can work and process this messages in parallel.

One way to implement this messaging pattern is to use publisher/subscriber or pub/sub model. In the pub/sub model we define a topic which is logical access point to enabling message communication with asynchronously.

A publisher simply sends the message to the topic. After that this message is immediately fanned out to all subscribers of this topic. This message communication is completely decoupled and asynchronously. Each service can operate and scale independently and individually without having any dependency of other services. The publisher doesn’t need to know who is consuming this message that is broadcasting. And the subscribers don’t need to know where the message comes from. The best way to build pub/sub fan out messaging on AWS is to use Amazon SNS. Amazon SNS is fully managed reliable and secure pub/sub messaging service.

So this architectural challenges recommends by using messaging patterns, resulting in loosely coupled communication between highly cohesive components to manage complexity in serverless architectures. A common approach when one component wishes to deliver the same message to multiple receivers is to use the fanout publish/subscribe messaging pattern.

What is Pub/Sub Messaging? Publish/subscribe messaging, or pub/sub messaging, is a form of asynchronous service-to-service communication used in serverless and microservices architectures. In a pub/sub model, any message published to a topic is immediately received by all of the subscribers to the topic.

Pub/sub messaging can be used to enable event-driven architectures, or to decouple applications in order to increase performance, reliability and scalability. The Publish Subscribe model allows messages to be broadcast to different parts of a system asynchronously.

In case of SQS - multi-consumers if one consumer has already picked the message and is processing, in meantime other consumer can pick it up and process the message there by two copies are added at the end. To avoid this the message is made invisible from the time its picked and deleted after processing. This visibility timeout is increased according to max time taken to process the message.

When you publish a message to an Amazon SNS FIFO topic, the message must include a deduplication ID. This ID is included in the message that the Amazon SNS FIFO topic delivers to the subscribed Amazon SQS FIFO queues.

If a message with a particular deduplication ID is successfully published to an Amazon SNS FIFO topic, any message published with the same deduplication ID, within the five-minute deduplication interval, is accepted but not delivered. The Amazon SNS FIFO topic continues to track the message deduplication ID, even after the message is delivered to subscribed endpoints.

If the message body is guaranteed to be unique for each published message, you can enable content-based deduplication for an Amazon SNS FIFO topic and the subscribed Amazon SQS FIFO queues. Amazon SNS uses the message body to generate a unique hash value to use as the deduplication ID for each message, so you don't need to set a deduplication ID when you send each message.

Amazon API Gateway

- is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs (application programming interface) at any scale.
- APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.
- Using API Gateway, you can create:
 - REST APIs (short for representational state transfer, an architectural style for an API that uses HTTP requests to access and use data). Use HTTP as the underlying protocol for communication, which in turn follows the request and response model where a client sends a request to a service and the service responds back synchronously. This kind of model is suitable for many different kinds of applications that depend on synchronous communication. Essentially, REST is an architectural style which puts a set of constraints on HTTP to create web services.
 - WebSocket APIs (protocol which makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply). WebSocket protocol starts off over HTTP, although it is further elevated to follow the WebSockets protocol if both the server and the client are compliant. It is a bidirectional protocol, a client can send messages to a service, and services can independently send messages to clients. Websocket APIs enable real-time two-way communication, this bidirectional behavior enables richer client/service interactions because services can push data to clients without requiring clients to make an explicit request. WebSocket APIs are often used in real-time applications such as chat applications, collaboration platforms, multiplayer games, GPS location tracking, Push Notifications and stock market prices updating in realtime.
- API Gateway supports containerized and serverless workloads, as well as web applications. It has a collection of API routes that are integrated with backend HTTP endpoints, Lambda functions, or other AWS services.
- API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, CORS support, authorization and access control, throttling, monitoring, and API version management.
- API Gateway has no minimum fees or startup costs. You pay for the API calls you receive and the amount of data transferred out.

Miscellaneous

Cluster Placement Group

Cluster placement group packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance.

Amazon Kinesis Data Firehose & Amazon Kinesis Data Analytics

This solution will meet the requirements with the least operational overhead as it uses Amazon Kinesis Data Firehose, which is a fully managed service that can automatically handle the data collection, data transformation, encryption, and data storage in near-real time. Kinesis Data Firehose can automatically store the data in Amazon S3 in Apache Parquet format for further processing. Additionally, it allows you to create an Amazon Kinesis Data Analytics application to analyze the data in near real-time, with no need to manage any infrastructure or invoke any Lambda function. This way you can process a large amount of data with the least operational overhead.

AWS Transfer Family

AWS Transfer Family securely scales your recurring business-to-business file transfers to AWS Storage services using SFTP, FTPS, FTP, and AS2 protocols.

Trust Policy

A trust policy is attached to an IAM (Identity and Access Management) role.

It specifies the trusted entities or services that are allowed to assume the role.

The trust policy defines the "principals" (entities) that can assume the role, such as IAM users, IAM roles, AWS services, or external identity providers.

It establishes the trust relationship between the role and the entities that are allowed to assume it.

Trust policies are written in JSON format and are part of IAM role configuration.

Amazon MQ

Message brokers allow software systems, which often use different programming languages on various platforms, to communicate and exchange information. Amazon MQ is a managed message broker service for Apache ActiveMQ and RabbitMQ that streamlines setup, operation, and management of message brokers on AWS. With a few steps, Amazon MQ can provision your message broker with support for software version upgrades.

Amazon OpenSearch Service

Amazon OpenSearch Service makes it easy for you to perform interactive log analytics, real-time application monitoring, website search, and more. OpenSearch is an open source, distributed search and analytics suite derived from Elasticsearch.

AWS ParallelCluster

AWS ParallelCluster is an open source cluster management tool that makes it easy for you to deploy and manage High Performance Computing (HPC) clusters on AWS. ParallelCluster uses a simple graphical user interface (GUI) or text file to model and provision the resources needed for your HPC applications in an automated and secure manner. It also supports multiple instance types and job submission queues, and job schedulers like AWS Batch and Slurm.

Amazon Textract

Amazon Textract is a machine learning (ML) service that automatically extracts text, handwriting, and data from scanned documents.

Amazon Comprehend Medical

Amazon Comprehend Medical is a HIPAA-eligible natural language processing (NLP) service that uses machine learning that has been pre-trained to understand and extract health data from medical text, such as prescriptions, procedures, or diagnoses.

Amazon AppFlow

With Amazon AppFlow automate bi-directional data flows between SaaS applications and AWS services in just a few clicks. Run the data flows at the frequency you choose, whether on a schedule, in response to a business event, or on demand. Simplify data preparation with transformations, partitioning, and aggregation. Automate preparation and registration of your schema with the AWS Glue Data Catalog so you can discover and share data with AWS analytics and machine learning services.

Amazon Lex

- is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. With Amazon Lex, the same deep learning technologies that power Amazon Alexa are now available to any developer, enabling you to quickly and easily build sophisticated, natural language, conversational bots (“chatbots”).
- With Amazon Lex, you can build bots to increase contact center productivity, automate simple tasks, and drive operational efficiencies across the enterprise. As a fully managed service, Amazon Lex scales automatically, so you don’t need to worry about managing infrastructure.



Figure 3: An example Amazon Lex implementation

Amazon Polly

- is a service that turns text into lifelike speech, allowing you to create applications that talk, and build entirely new categories of speech-enabled products. Polly's Text-to-Speech (TTS) service uses advanced deep learning technologies to synthesize natural sounding human speech. With dozens of lifelike voices across a broad set of languages, you can build speech-enabled applications that work in many different countries.
- In addition to Standard TTS voices, Amazon Polly offers Neural Text-to-Speech (NTTS) voices that deliver advanced improvements in speech quality through a new machine learning approach. Polly's Neural TTS technology also supports two speaking styles that allow you to better match the delivery style of the speaker to the application: a Newscaster reading style that is tailored to news narration use cases, and a Conversational speaking style that is ideal for two-way communication like telephony applications. Finally, Amazon Polly Brand Voice can create a custom voice for your organization. This is a custom engagement where you will work with the Amazon Polly team to build an NTTS voice for the exclusive use of your organization.

Amazon Rekognition

- image and video analysis for your applications using highly scalable, deep learning technology that requires no machine learning expertise to use. With Amazon Rekognition, you can identify objects, people, text, scenes, and activities in images and videos, as well as detect any inappropriate content.
- Amazon Rekognition also provides highly accurate facial analysis and facial search capabilities that you can use to detect, analyze, and compare faces for a wide variety of user verification, people counting, and public safety use cases.

Amazon Transcribe

- makes it easy for developers to add speech to text capabilities to their applications. Audio data is virtually impossible for computers to search and analyze. Therefore, recorded speech needs to be converted to text before it can be used in applications. Historically, customers had to work with transcription providers that required them to sign expensive contracts and were hard to integrate into their technology stacks to accomplish this task. Many of these providers use outdated technology that does not adapt well to different scenarios, like low-fidelity phone audio common in contact centers, which results in poor accuracy.
- Amazon Transcribe uses a deep learning process called automatic speech recognition (ASR) to convert speech to text quickly and accurately. Amazon Transcribe can be used to transcribe customer service calls, to automate closed captioning and subtitling, and to generate metadata for media assets to create a fully searchable archive. You can use Amazon Transcribe Medical to add medical speech to text capabilities to clinical documentation applications.

Amazon Sagemaker

- Service which provides ability to build, train and deploy machine learning (ML) models quickly

AWS Lake Formation

- a service that makes it easy to set up a secure data lake in days.
- A data lake is a centralized, curated, and secured repository that stores all your data, both in its original form and prepared for analysis. A data lake enables you to break down data silos and combine different types of analytics to gain insights and guide better business decisions.
- However, setting up and managing data lakes today involves a lot of manual, complicated, and time-consuming tasks. This work includes loading data from diverse sources, monitoring those data flows, setting up partitions, turning on encryption and managing keys, defining transformation jobs and monitoring their operation, re-organizing data into a columnar format, configuring access control settings, deduplicating redundant data, matching linked records, granting access to data sets, and auditing access over time.
- Creating a data lake with Lake Formation is as simple as defining data sources and what data access and security policies you want to apply. Lake Formation then helps you collect and catalog data from databases and object storage, move the data into your new Amazon S3 data lake, clean and classify your data using machine learning algorithms, and secure access to your sensitive data. Your users can access a centralized data catalog which describes available data sets and their appropriate usage. Your users then leverage these data sets with their choice of analytics and machine learning services, like Amazon Redshift, Amazon Athena, and (in beta) Amazon Elastic MapReduce (EMR) for Apache Spark. Lake Formation builds on the capabilities available in AWS Glue.

Amazon Comprehend

Derive and understand valuable insights from text within documents. Amazon Comprehend is a natural-language processing (NLP) service that uses machine learning to uncover valuable insights and connections in text.

Amazon EMR

Amazon EMR is the industry-leading cloud big data solution for petabyte-scale data processing, interactive analytics, and machine learning using open-source frameworks such as Apache Spark, Apache Hive, and Presto.

AWS Glue

AWS Glue is a serverless data integration service that makes it easier to discover, prepare, move, and integrate data from multiple sources for analytics, machine learning (ML), and application development.

AWS Glue tracks data that has already been processed during a previous run of an ETL job by persisting state information from the job run. This persisted state information is called a job bookmark. Job bookmarks help AWS Glue maintain state information and prevent the reprocessing of old data.

AWS Glue Data Catalog

The AWS Glue Data Catalog is a centralized metadata repository for all your data assets across various data sources. It provides a unified interface to store and query information about data formats, schemas, and sources. When an AWS Glue ETL job runs, it uses this catalog to understand information about the data and ensure that it is transformed correctly.

Amazon QuickSight

- is a fast, cloud-powered business intelligence service that makes it easy to deliver insights to everyone in your organization.
- as a fully managed service, QuickSight lets you easily create and publish interactive dashboards that include ML Insights. Dashboards can then be accessed from any device, and embedded into your applications, portals, and websites.
- with our Pay-per-Session pricing, QuickSight allows you to give everyone access to the data they need, while only paying for what you use.

Amazon Simple Notification Service

- is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.
 - The A2A pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems including Amazon SQS queues, AWS Lambda functions and HTTPS endpoints, for parallel processing.
 - The A2P functionality enables you to send messages to users at scale via SMS, mobile push, and email.

Amazon Pinpoint

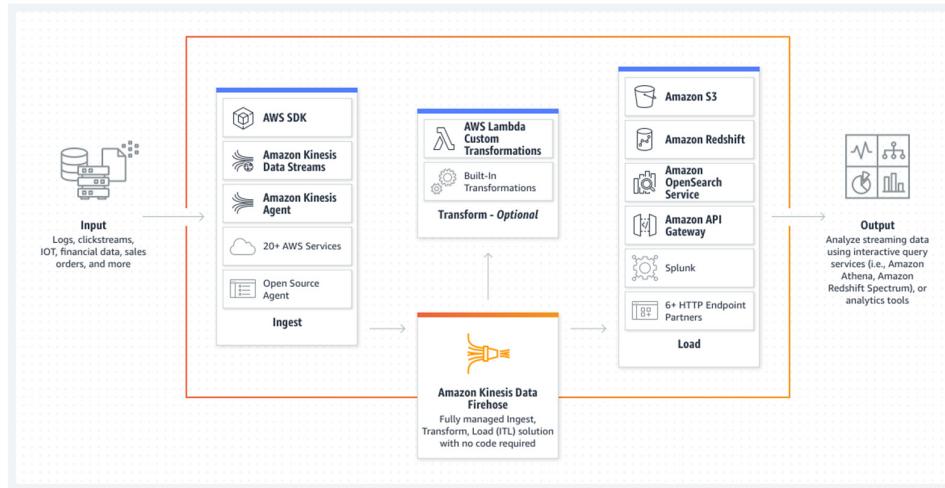
- a flexible and scalable outbound and inbound multichannel marketing communications service. You can connect with customers over channels like email, SMS, push, or voice. Segment your campaign audience for the right customer and personalize your messages with the right content. Delivery and campaign metrics in Amazon Pinpoint measure the success of your communications. Amazon Pinpoint can grow and scales globally

Amazon Kinesis

- 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.
- Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application.
- 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.

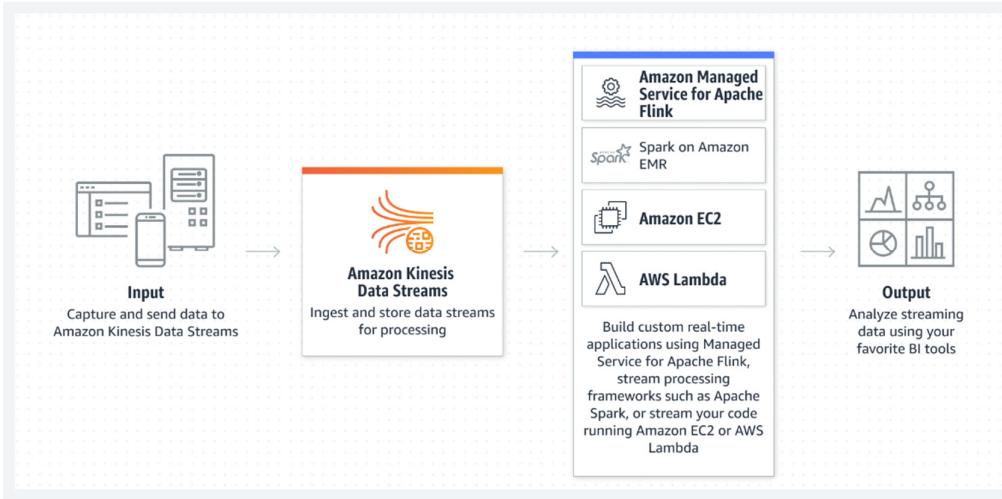
Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose is an extract, transform, and load (ETL) service that reliably captures, transforms, and delivers streaming data to data lakes, data stores, and analytics services.



Amazon Kinesis Data Streams

Amazon Kinesis Data Streams is a fully managed, serverless data streaming service that stores and ingests various streaming data in real time at any scale.



Data Streams vs Firehose

Kinesis Data Streams focuses on ingesting and storing data streams.

Kinesis Data Firehose focuses on delivering data streams to select destinations.

Both can ingest data streams but the deciding factor in which to use depends on where your streamed data should go to.

If you already have a data lake or data store (e.g. S3, RedShift, ElasticSearch) where you want your data stream to be delivered to, Kinesis Data Firehose might be for you. Firehose has been built to remove the admin work that comes with delivering data streams. Do note that data latency for Firehose is 60 seconds or higher.

If you have your own data processing pipeline and just need to be able to stream data there, then Kinesis Data Streams will be for you. It's for custom processing. It also has sub-second processing latency which Firehose doesn't have.

If you're not sure which to use, you can start with Kinesis Data Streams. You will be able to attach Firehose to it without much additional work. However if all you want is to just park the data somewhere, then you can go right ahead with Firehose.

AWS IoT Greengrass

AWS IoT Greengrass is a service that extends Amazon Web Services functionality to Internet of Things (IoT) devices, allowing a business to perform data collection and analysis closer to its origin.

AWS IoT Greengrass seamlessly extends AWS to edge devices so they can act locally on the data they generate, while still using the cloud for management, analytics, and durable storage. With AWS IoT Greengrass, connected devices can run AWS Lambda functions, Docker containers, or both, execute predictions based on machine learning models, keep device data in sync, and communicate with other devices securely – even when not connected to the Internet.

AWS Data Pipeline

- is a web service that you can use to automate the movement and transformation of data. With AWS Data Pipeline, you can define data-driven workflows, so that tasks can be dependent on the successful completion of previous tasks. You define the parameters of your data transformations and AWS Data Pipeline enforces the logic that you've set up.

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals. With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon EMR.

Amazon AppStream 2.0

- is a fully managed non-persistent application and desktop streaming service. You centrally manage your desktop applications on AppStream 2.0 and securely deliver them to any computer. You can easily scale to any number of users across the globe without acquiring, provisioning, and operating hardware or infrastructure. AppStream 2.0 is built on AWS, so you benefit from a data center and network architecture designed for the most security-sensitive organizations. Each end user has a fluid and responsive experience because your applications run on virtual machines optimized for specific use cases and each streaming session automatically adjusts to network conditions.

Amazon Cognito

- Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0

Amazon Connect

- Amazon Connect is an easy to use omnichannel cloud contact center that helps you provide superior customer service at a lower cost. Over 10 years ago, Amazon's retail business needed a contact center that would give our customers personal, dynamic, and natural experiences. We couldn't find one that met our needs, so we built it. We've now made this available for all businesses, and today thousands of companies ranging from 10 to tens of thousands of agents use Amazon Connect to serve millions of customers daily.
- Designed from the ground up to be omnichannel, Amazon Connect provides a seamless experience across voice and chat for your customers and agents. This includes one set of tools for skills-based routing, task management, powerful real-time and historical analytics, and intuitive management tools – all with pay-as-you-go pricing, which means Amazon Connect simplifies contact center operations, improves agent efficiency, and lowers costs. You can set up a contact center in minutes that can scale to support millions of customers from the office or as a virtual contact center.

Glossary

A

- Alias Record – See CNAME
- Amazon Athena
 - Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.
 - Queries use standard SQL. Most results are delivered within seconds.
-

C

- CDMB (Configuration Management Database) –
 - A CMDB is a repository that acts as a data warehouse – storing information about your IT environment, the components that are used to deliver IT services. The data stored in a CMDB include lists of assets (referred to as configuration items) and the relationships among them.
- CNAME –
 - is a Canonical Name Record or Alias Record. A type of resource record in the Domain Name System (DNS), that specifies that one domain name is an alias of another canonical domain name. Any system hosting a Web site must have an IP address in order to be connected to the World Wide Web.
 - CNAME records must always point to another domain name, never directly to an IP address.
 - A common example is when you have both example.com and www.example.com pointing to the same application and hosted by the same server. To avoid maintaining two different records, it's common to create: (1) An A record for example.com pointing to the server IP address (2) A CNAME record for www.example.com pointing to example.com. As a result, example.com points to the server IP address, and www.example.com points to the same address via example.com. If the IP address changes, you only need to update it in one place: just edit the A record for example.com, and www.example.com automatically inherits the changes.
- Colocation center –
 - Also known as a "carrier hotel". Colocation is the practice of housing privately-owned servers and networking equipment in a third-party data center. It can also extend to renting equipment, bandwidth and other resources. It is a shared facility generally with other paying tenants. It is good for businesses that require full control over their equipment. When compared with traditional datacentre there is access to higher levels of bandwidth, higher reliability and higher levels of physical protection. This is different to AWS which manages the entire datacentre themselves and instead provides products packaged as services e.g. EC2 for compute power or S3 for object storage.
- Container –
 - A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A container is a uniform structure in which any application can be stored, transported and run.
 - It is named for and often compared to the standardised intermodal containers used in the shipping industry for efficient transportation.
 - In the software world, containerisation is an efficient method for deploying applications. A container encapsulates an application with its own operating environment. It can be placed on any host machine without special configuration.
 - Virtual machines (VMs) and containers are not the same. Deploying VMs is hardware virtualisation whereas containerisation is OS virtualisation. An application on a VM requires a guest OS and thus an underlying hypervisor to run. By contrast, an application in a container, doesn't require a guest OS or hypervisor. It allows an application to run in the Userspace of the OS – a segment of the computer memory that is kept separate from the critical processes of the OS kernel. This leads to

improved performance, as an application's instructions do not have to pass through the guest OS and the hypervisor to reach the CPU. It also means that applications in containers are smaller and can be started up in seconds, compared to minutes for VMs. Significantly, container applications offer much more stability - they never hang on the host OS, like VM applications can do, which takes all VMs on the host offline.

- One of the appeals of using containers is their ability to die gracefully and respawn upon demand. Whether a container's demise is caused by a crash or because it's simply no longer needed when server traffic is low, containers are cheap to start, and they're designed to seamlessly appear and disappear.
- Containers can be thought of as necessitating three categories of software:
 - Builder: technology used to build a container (e.g. Docker)
 - Engine: technology used to run a container (e.g. Docker)
 - Orchestration: technology used to manage many container (e.g. Kubernetes)
- Containers are used in PaaS where customer is responsible for app and data and the rest is taken care of by the cloud provider.
- CI/CD (continuous integration, continuous delivery/deployment) –
 - CI - is a software development practice in which all developers merge code changes in a central repository multiple times a day
 - With CI, each change in code triggers an automated build-and-test sequence for the given project, providing feedback to the developer(s) who made the change. The entire CI feedback loop should run in less than 10 minutes.
 - CD adds the practice of automating the entire software release process. The purpose of continuous delivery is to ensure that it takes minimal effort to deploy new code.
 - Continuous Delivery includes infrastructure provisioning and deployment, which may be manual and consist of multiple stages. What's important is that all these processes are fully automated, with each run fully logged and visible to the entire team.
 - Continuous deployment (the other possible "CD") can refer to automatically releasing a developer's changes from the repository to production, where it is usable by customers. It addresses the problem of overloading operations teams with manual processes that slow down app delivery. It builds on the benefits of continuous delivery by automating the next stage in the pipeline.

D

- Docker –
 - an open source project launched in 2013
 - Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. It has became the de facto standard program in this area.
 - The container will hold everything the application requires to run from within its container package, this can include system libraries, code, tools, etc... Does not include a VM. Containers are decoupled from the OS, making container applications very portable. They should run as expected, regardless of their deployment location.
 - It uses OS-level virtualization to deliver software in packages called containers.
 - Docker allows applications to use the same OS kernel as the system leading to more efficient operation
- DNS (Domain Name Service)
 - Translates human readable domain names into numeric IP addresses

- [IaaS \(Infrastructure as a Service\)](#)
- [IAM](#)
 - [Users](#)
 - [Roles](#)
- IdP –
 - an identity provider, that manages your user identities outside of AWS, such as Login with Amazon, Facebook, or Google
- IOPS –
 - Input/Output operations per second. The operations are measured in KiB.
 - It is a performance metric used to distinguish one storage type from another.
 - The underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O.
 - SSD volumes generally I/O much more efficiently than HDD volumes

J

- JSON –
 - stands for JavaScript Object Notation
 - is an lightweight open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value).
 - It is a very common data format, with a diverse range of applications

K

- Kubernetes -
 - is an open-source container-orchestration system for automating computer application deployment, networking, load-balancing, security, scaling, and management
 - It will orchestrate the running of containers across a potentially large number of hosts (also called nodes), these can be Docker hosts, bare-metal servers or virtual machines.
 - A collection of nodes that is managed by a single Kubernetes instance is referred to as a Kubernetes cluster.
 - Kubernetes can control its clusters from a single command line or dashboard.
 - The name Kubernetes originates from Greek, meaning helmsman or pilot.
 - It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation
 -
- KPI (Key performance indicator) –
 - a type of performance measurement. KPIs evaluate the success of an organization or of a particular activity in which it engages

N

- NAS –
 - Network attached storage – single storage devices that provides file-level storage to clients on the network
- NAT (Network Address Translation)
 - You can use a NAT instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the internet.

O

- OpenStack –
 - is a free open standard cloud computing platform, mostly deployed as infrastructure-as-a-service in both public and private clouds where virtual servers and other resources are made available to users

P

- [PaaS \(Platform as a Service\)](#)
- [Principle of Least Privilege](#)

R

- [Root Account](#)

S

- [SaaS \(Software as a Service\)](#)
- SAN –
 - a specialized, high-speed network that provides block-level network access to storage. Used to improve application availability (e.g., multiple data paths), enhance application performance, increase storage effectiveness and improve data protection and security.
- SSH (Secure Shell)
 - is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network
 - Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH
- SSL (Secure Sockets Layer) – See TLS

T

- (TLS) Transport Layer Security –
 - a cryptographic protocol designed to provide communications security over a computer network. When sender and recipient computers send data they both agree to encrypt the information in a way they both understand. If either machine cannot support an encrypted connection, both services will default to a less secure Secure Sockets Layer (SSL) connection, a non-encrypted connection or may simply refuse to connect, all depending on the rules in place.
 - A public encryption key is used to encrypt data while a private key only held by the data recipient is used to decrypt the data
 - Uses include secure web browsing (and in particular the padlock icon that appears in web browsers when a secure session is established) and sending and receiving emails securely.

U

- Unified SAN –
 - NAS & SAN used together

V

- VMware –
 - is an American publicly traded software company from California, USA. It provides cloud computing and virtualization software and services. Based in Palo Alto, California. Founded in 1998, VMware is a subsidiary of Dell Technologies

W

Y

- YAML –
 - is a recursive acronym for "YAML Ain't Markup Language"
 - is a human-readable data-serialization language. It is commonly used for configuration files and in applications where data is being stored or transmitted
 - uses key-value pairs to store data

