AWS Solution Architect Associate

- EBS Volume
  - RAID 0
  - RAID 1
  - Provisioned IOPS SSD `io1` volumes
  - Encryption
- Amazon RDS
  - RDS Multi-AZ
  - Read Replicas
  - Enhanced Monitoring
- Amazon Aurora
  - Replicas
  - Failover
- DynamoDB
  - DynamoDB auto scaling
  - Endpoints
  - Backups
  - DynamoDB Accelerator (DAX)
  - Streams
- Amazon Neptune
- AWS Database Migration Service (AWS DMS)
- AWS Elastic Disaster Recovery (DRS)
  - Disaster Recovery strategies
    - Backup and Restore
    - Pilot Light
    - Warm Standby
    - Multi-Site
- Amazon Redshift
  - Redshift Spectrum
- Amazon S3
  - Performance limits
  - S3 Static Website
  - S3 Object Lock
    - Legal Hold vs. Retention Period
  - S3 Access Point
  - Data protection
  - Glacier
  - Notification feature
- Amazon FSx for Lustre
- Amazon FSx for Windows File Server
  - Amazon FSx File Gateway
- AWS Transfer (for SFTP)

AWS Identity and Access Management (IAM)

As a best practice, **use IAM roles instead of sharing access keys.**

Policy Elements: Include `Version`, `Statement`, `Effect`, `Action`, `Resource`, and optional `Condition` keys.

Utilize service control policies (SCPs) in AWS Organizations to set boundaries for account permissions. A permissions boundary is an advanced feature for using a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity.

Trust policies define which principal entities (accounts, users, roles, and federated users) can assume the role. An IAM role is both an identity and a resource that supports resource-based policies. For this reason, you must attach both a trust policy and an identity-based policy to an IAM role. **The IAM service supports only one type of resource-based policy called a role trust policy, which is attached to an IAM role.**

IAM Role

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user, in that it **is an AWS identity with permission policies that determine what the identity can and cannot do in AWS**. However, instead of being uniquely associated with one person, **a role is intended to be assumable by anyone who needs it**. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. For example, you might want to **grant users in your AWS account access to resources they don't usually have**, or **grant users in one AWS account access to resources in another**

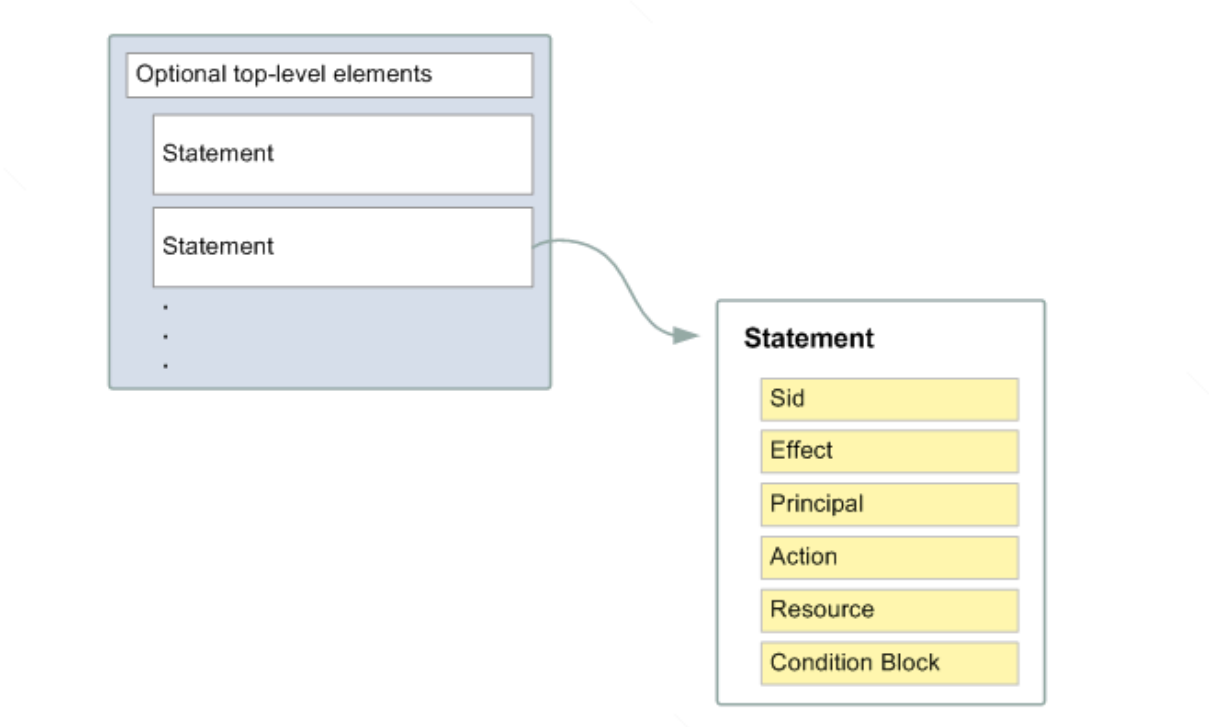
account. Or you might want to allow a mobile app to use AWS resources, but not want to embed AWS keys within the app (where they can be difficult to update and where users can potentially extract them). Sometimes you want to give AWS access to users who already have identities defined outside of AWS, such as in your corporate directory. Or, you might want to grant access to your account to third parties so that they can perform an audit on your resources.

For these scenarios, you can delegate access to AWS resources using an IAM role. This section introduces roles and the different ways you can use them, when and how to choose among approaches, and how to create, manage, switch to (or assume), and delete roles.

IAM 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.

A policy default effect is `Deny`



IAM DB Authentication

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

An **authentication token** is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.

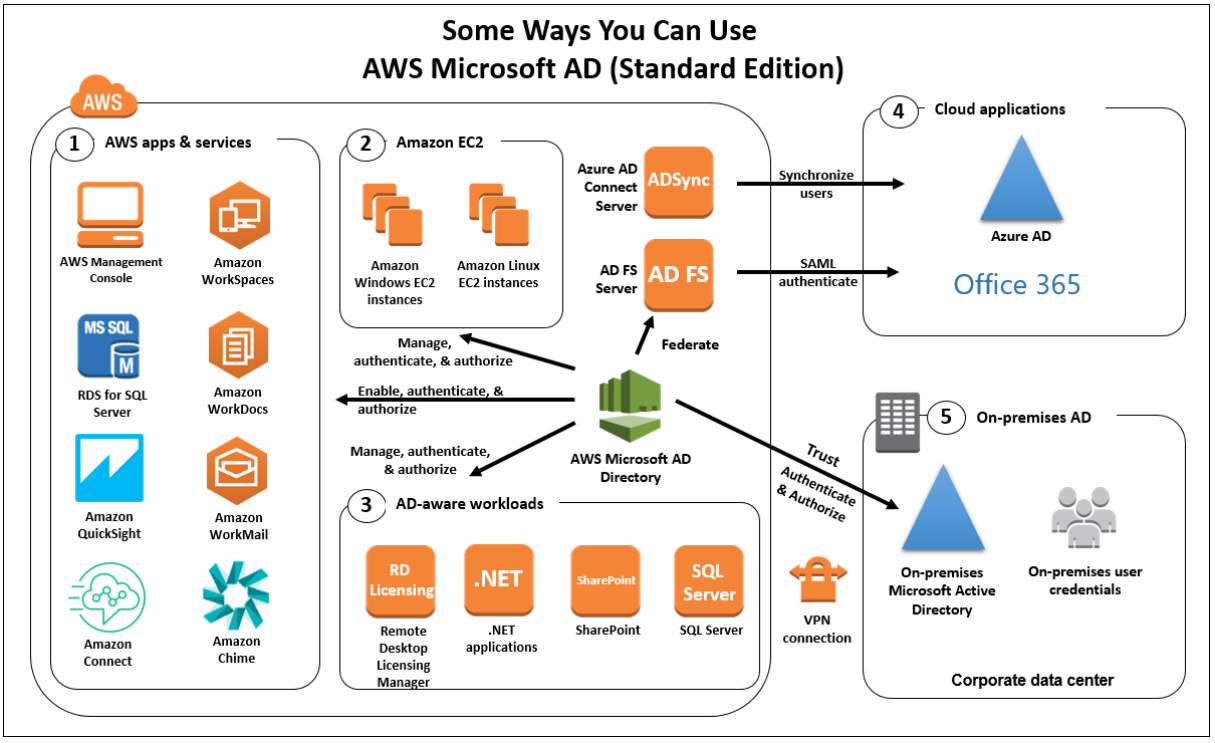
AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD)

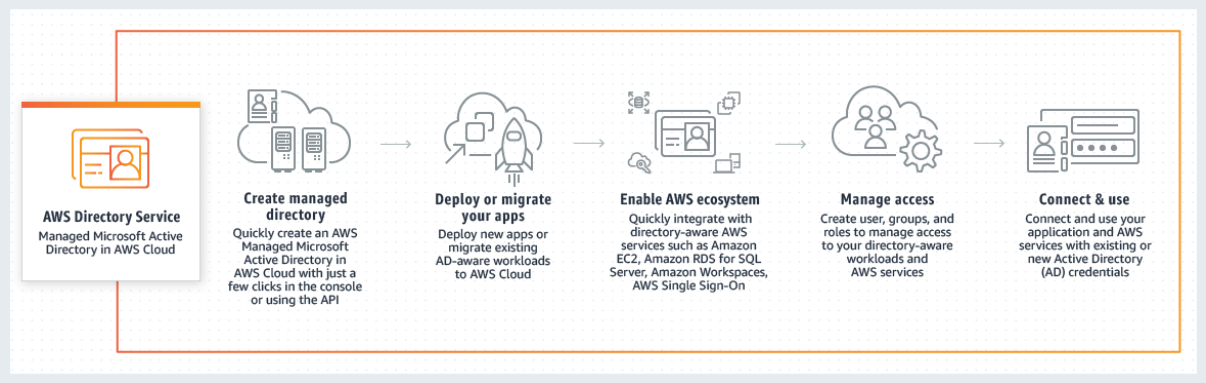
AWS Directory Service lets you run Microsoft Active Directory (AD) as a managed service.

You should use it when the engineering team at the company wants to run directory-aware workloads on AWS for a SQL Server-based application. The team also wants to configure a trust relationship to enable single sign-on (SSO) for its users to access resources in either domain.

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

If the company is using a corporate Active Directory, it is best to use AWS Directory Service AD Connector for easier integration. If the roles are already assigned using groups in the corporate Active Directory, it would be better to also use IAM Roles. Take note that you can assign an IAM Role to the users or groups from your Active Directory once it is integrated with your VPC via the AWS Directory Service AD Connector.





Identity store not compatible with SAML

If your identity store is not compatible with SAML 2.0 then you can **build a custom identity broker application** to perform a similar function. **The broker application authenticates users, requests temporary credentials for users from AWS, and then provides them to the user to access AWS resources**.

The application verifies that employees are signed into the existing corporate network's identity and authentication system, which might use LDAP, Active Directory, or another system. The identity broker application then obtains temporary security credentials for the employees.

To get temporary security credentials, the identity broker application calls either `AssumeRole` or `GetFederationToken` to obtain temporary security credentials, depending on how you want to manage the policies for users and when the temporary

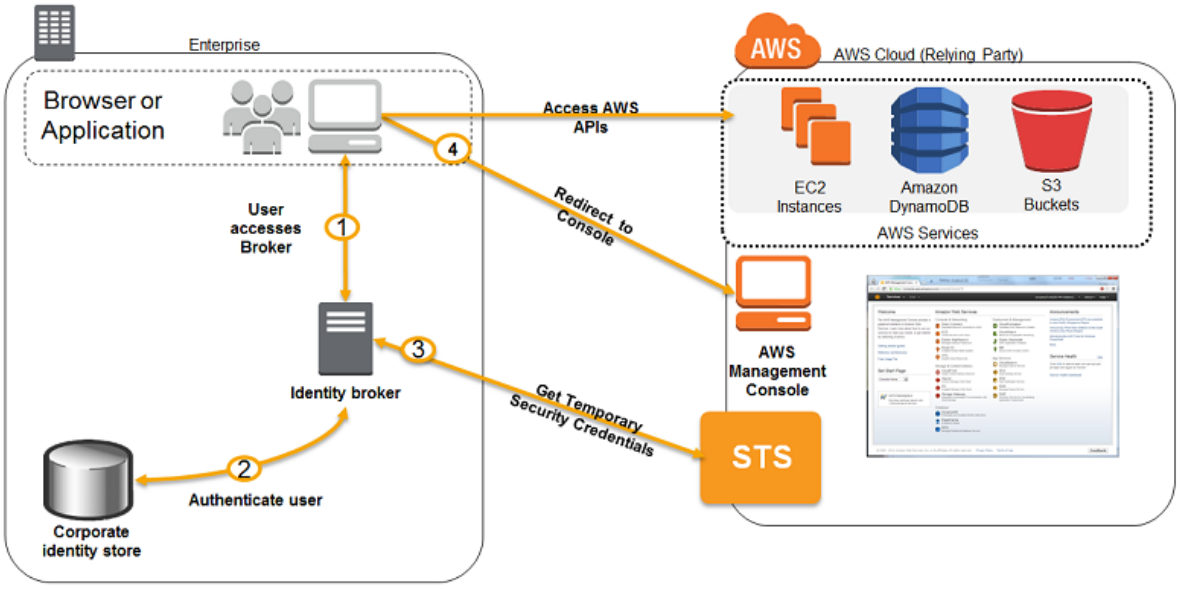
credentials should expire. The call returns temporary security credentials consisting of an AWS access key ID, a secret access key, and a session token. The identity broker application makes these temporary security credentials available to the internal company application. The app can then use the temporary credentials to make calls to AWS directly. The app caches the credentials until they expire, and then requests a new set of temporary credentials.

AWS Security Token Service (STS)

AWS provides AWS Security Token Service (AWS STS) as a web service that enables you to request temporary, limited-privilege credentials for users.

Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles.

In an enterprise identity federation, you can authenticate users in your organization's network, and then provide those users access to AWS without creating new AWS identities for them and requiring them to sign in with a separate username and password. This is known as the single sign-on (SSO) approach to temporary access. AWS STS supports open standards like Security Assertion Markup Language (SAML) 2.0, with which you can use Microsoft AD FS to leverage your Microsoft Active Directory. You can also use SAML 2.0 to manage your own solution for federating user identities.



AWS IAM Identity Center

AWS IAM Identity Center (successor to AWS Single Sign-On) **provides single sign-on access for all of your AWS accounts and cloud applications. It connects with Microsoft Active Directory through AWS Directory Service** to allow users in that directory to sign in to a personalized AWS access portal using their existing Active Directory user names and passwords. From the AWS access portal, users have access to all the AWS accounts and cloud applications that they have permission for.

Users in your self-managed directory in Active Directory (AD) can also have single sign-on access to AWS accounts and cloud applications in the AWS access portal.

IAM certificate store

To enable HTTPS connections to your website or application in AWS, you need an SSL/TLS server certificate. **For certificates in a Region supported by AWS Certificate Manager (ACM), we recommend that you use ACM to provision, manage, and deploy your server certificates. In unsupported Regions, you must use IAM as a certificate manager**. To learn which Regions ACM supports, see AWS Certificate Manager endpoints and quotas in the AWS General Reference.

AWS Single Sign-On (SSO)

Single sign-on (SSO) is an authentication solution that allows users to **log in to multiple applications and websites with one-time user authentication**. Given that users today frequently access applications directly from their browsers, organizations are prioritizing access management strategies that improve both security and the user experience. SSO delivers both aspects, as users can access all password-protected resources without repeated logins once their identity is validated.

Amazon Cognito

User Pools

Built-in user management or integrate with external identity providers, such as Facebook, Twitter, Google+, and Amazon.

Can be integrated with Application Load Balancer.

Cannot be integrated with CloudFront (without a custom Lambda@Edge).

Identity Pools

Identity pools provide AWS credentials to grant your users access to other AWS services. To enable users in your user pool to access AWS resources, you can configure an identity pool to exchange user pool tokens for AWS credentials. So, identity pools aren't an authentication mechanism in themselves.

EC2

Only EBS-backed instances can be stopped and restarted. Remember that **an instance store-backed instance can only be rebooted or terminated, and its data will be erased if the EC2 instance is either stopped or terminated.**

If you stopped an EBS-backed EC2 instance, the volume is preserved, but the data in any attached instance store volume will be erased. Keep in mind that an EC2 instance has an underlying physical host computer. If the instance is stopped, AWS usually moves the instance to a new host computer. Your instance may stay on the same host computer if there are no problems with the host computer. In addition, its Elastic IP address is disassociated from the instance if it is an EC2-Classic instance. Otherwise, if it is an EC2-VPC instance, the Elastic IP address remains associated.

Hibernation provides the convenience of pausing and resuming the instances, saves time by reducing the startup time taken by applications, and saves effort in setting up the environment or applications all over again. Instead of having to rebuild the memory footprint, hibernation allows applications to pick up exactly where they left off. **It is not possible to enable or disable hibernation for an instance after it has been launched**.

While the instance is in hibernation, you pay only for the EBS volumes and Elastic IP Addresses attached to it; there are no other hourly charges (just like any other stopped instance).

You are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region. If you need more instances, complete the Amazon EC2 limit increase request form with your use case, and your limit increase will be considered. Limit increases are tied to the region they were requested for.

Dedicated Instances vs Dedicated Host

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.

A Dedicated Host is also a physical server that's dedicated for your use. With a Dedicated Host, you have visibility and control over how instances are placed on the server. Dedicated Hosts allow you to use your existing software licenses on EC2 instances. Costlier.

You can change the tenancy of an instance from dedicated to host.

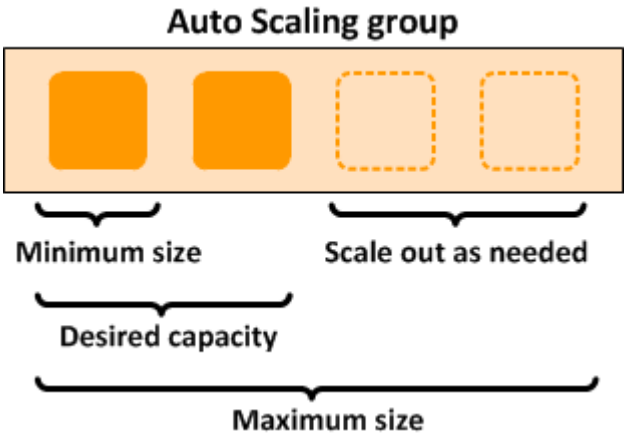
You can change the tenancy of an instance from host to dedicated.

EC2 Auto Scaling

Amazon EC2 Auto Scaling helps ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. **You create collections of EC2 instances, called Auto Scaling groups**. **You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size**. **You can also specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size**.

Amazon EC2 Auto Scaling does not immediately terminate instances with an Impaired status.

By default, Amazon EC2 Auto Scaling doesn't use the results of ELB health checks to determine an instance's health status when the group's health check configuration is set to EC2. As a result, Amazon EC2 Auto Scaling doesn't terminate instances that fail ELB health checks.



Auto Scaling Group

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service. The size of an Auto Scaling group depends on the number of instances that you set as the desired

capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

Step scaling policies and **simple scaling** policies are two of the dynamic scaling options available for you to use. **Both require you to create CloudWatch alarms for the scaling policies. Both require you to specify the high and low thresholds for the alarms. Both require you to define whether to add or remove instances, and how many, or set the group to an exact size**. The main difference between the policy types is the step adjustments that you get with step scaling policies. **When step adjustments are applied the adjustments vary based on the size of the alarm breach. When you create a step scaling policy, you can also specify the number of seconds that it takes for a newly launched instance to warm up.**

The primary issue with **simple scaling** is that after a scaling activity is started, the policy must wait for the scaling activity or health check replacement to complete and the **cooldown** period to expire before responding to additional alarms. Cooldown periods help to prevent the initiation of additional scaling activities before the effects of previous activities are visible. The default cooldown value is 300 seconds.

With a target tracking scaling policy, you can increase or decrease the current capacity of the group based on a target value for a specific metric. This policy will help **resolve the over-provisioning** of your resources. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to changes in the metric due to a changing load pattern.

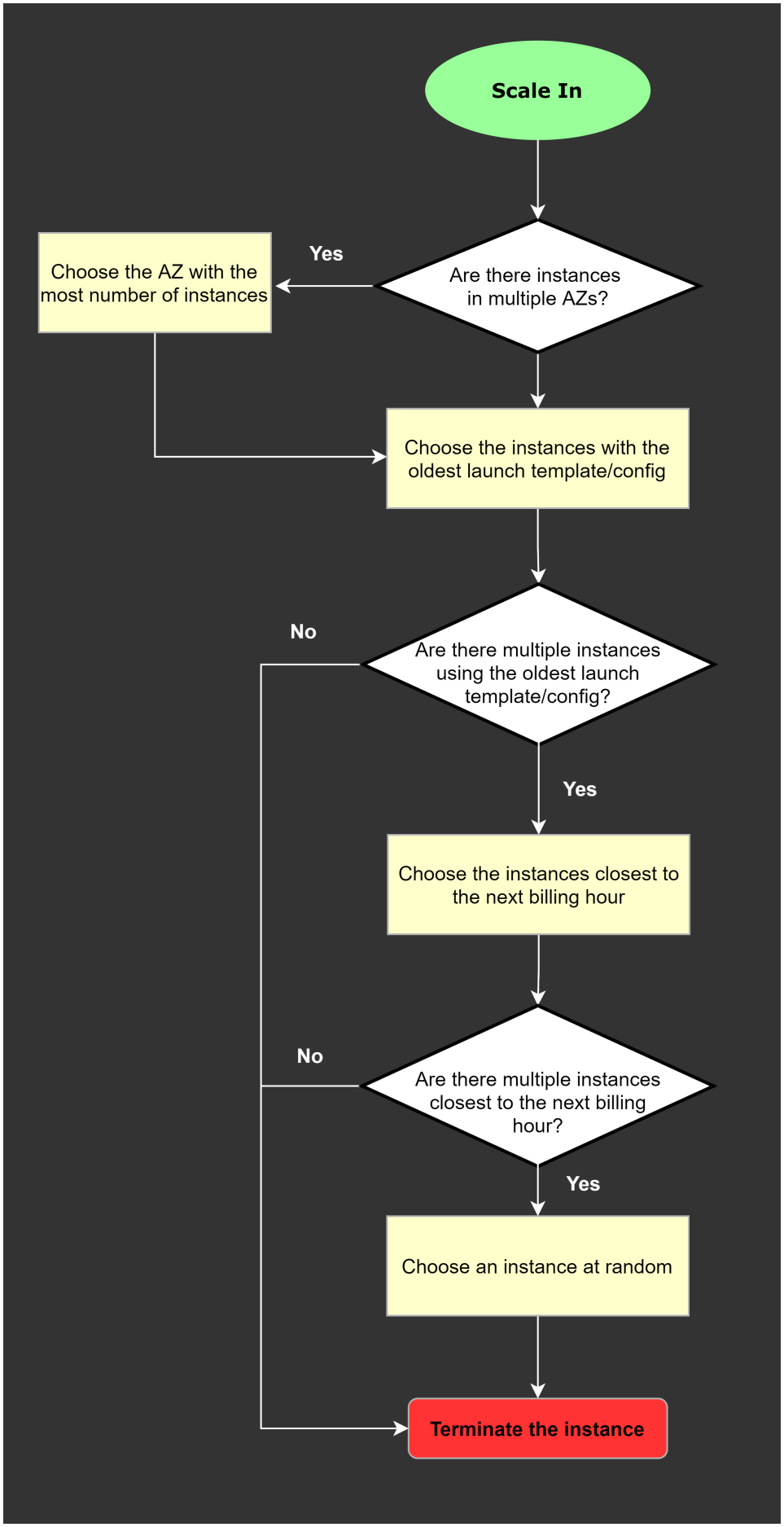
If the resources are unbalanced in the Availability Zones, Amazon EC2 Auto Scaling will compensate by rebalancing the Availability Zones. When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application.

Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance.

The default termination policy is designed to help ensure that your network architecture spans Availability Zones evenly. With the default termination policy, the behavior of the Auto Scaling group is as follows:

1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that use the oldest launch configuration.
2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.
3. If there are multiple instances to terminate based on the above criteria, determine which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances and manage your Amazon EC2 usage costs.) If there is one such instance, terminate it.
4. If there is more than one unprotected instance closest to the next billing hour, choose one of these instances at random.

The following flow diagram illustrates how the default termination policy works:


Scale In
Are there instances in multiple AZs?
Yes
Choose the AZ with the most number of instances
Choose the instances with the oldest launch template/config
Are there multiple instances using the oldest launch template/config?
No
Yes
Choose the instances closest to the next billing hour
Are there multiple instances closest to the next billing hour?
No
Yes
Choose an instance at random
Terminate the instance

Launch configuration

A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances, such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.

You can specify your launch configuration with multiple Auto Scaling groups. However, you can only specify **one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it**. Therefore, **if you want to change the launch configuration for an Auto Scaling group, you must create a launch configuration and then update your Auto Scaling group with the new launch configuration.**

Launch template

A launch template is similar to a launch configuration, in that it specifies instance configuration information. It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances. **However, defining a launch template instead of a launch configuration allows you to have multiple versions of a launch template**.

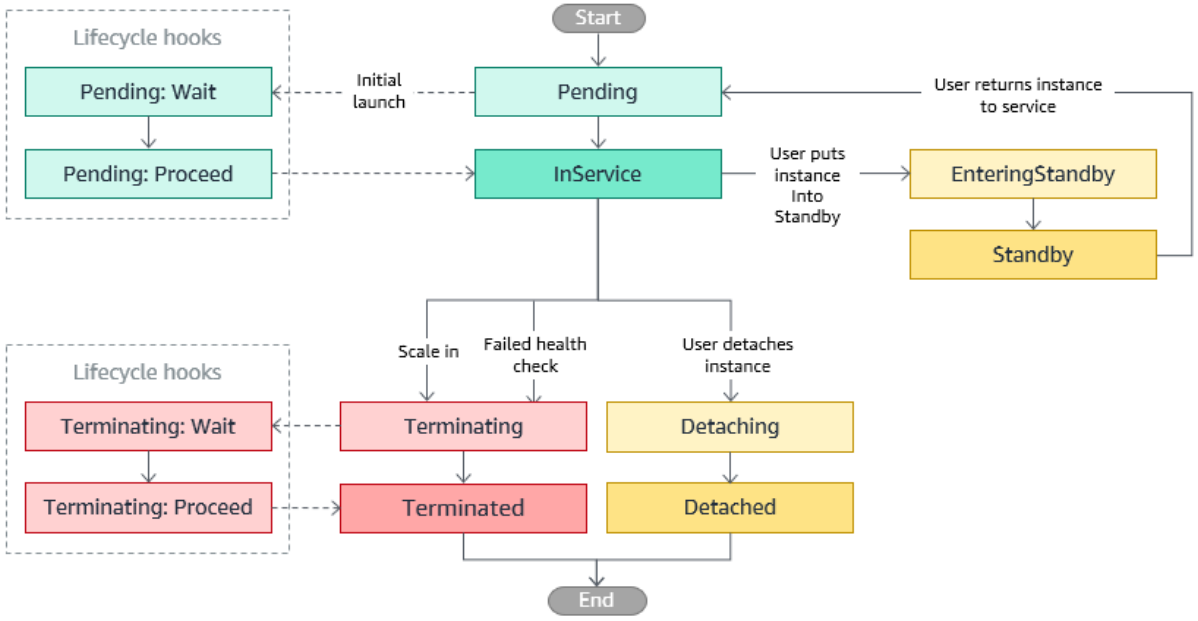
With launch templates, you can provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost.

With versioning of launch templates, you can create a subset of the full set of parameters.

Lifecycle

The EC2 instances in an Auto Scaling group have a path, or lifecycle, that differs from that of other EC2 instances. The lifecycle starts when the Auto Scaling group launches an instance and puts it into service. The lifecycle ends when you terminate the instance, or the Auto Scaling group takes the instance out of service and terminates it.

The `ReplaceUnhealthy` process terminates instances that are marked as unhealthy and then creates new instances to replace them.



Cluster vs Partition vs Spread placement group

A cluster placement group packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of high-performance computing (HPC) applications.

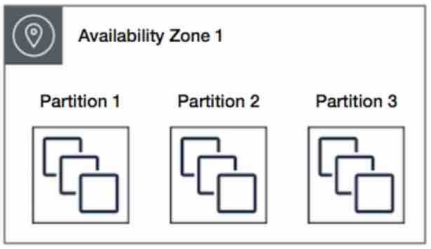
Partition placement group – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as **Hadoop, Cassandra, and Kafka.** Therefore, this is the correct option for the given use-case.

A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group. Therefore, to deploy 15 Amazon EC2 instances in a single Spread placement group, the company needs to use 3 Availability Zones.

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

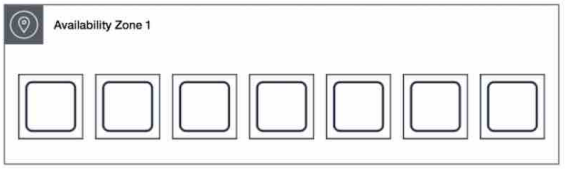
The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—**Partition 1**, **Partition 2**, and **Partition 3**. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.



Spread placement groups

A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.

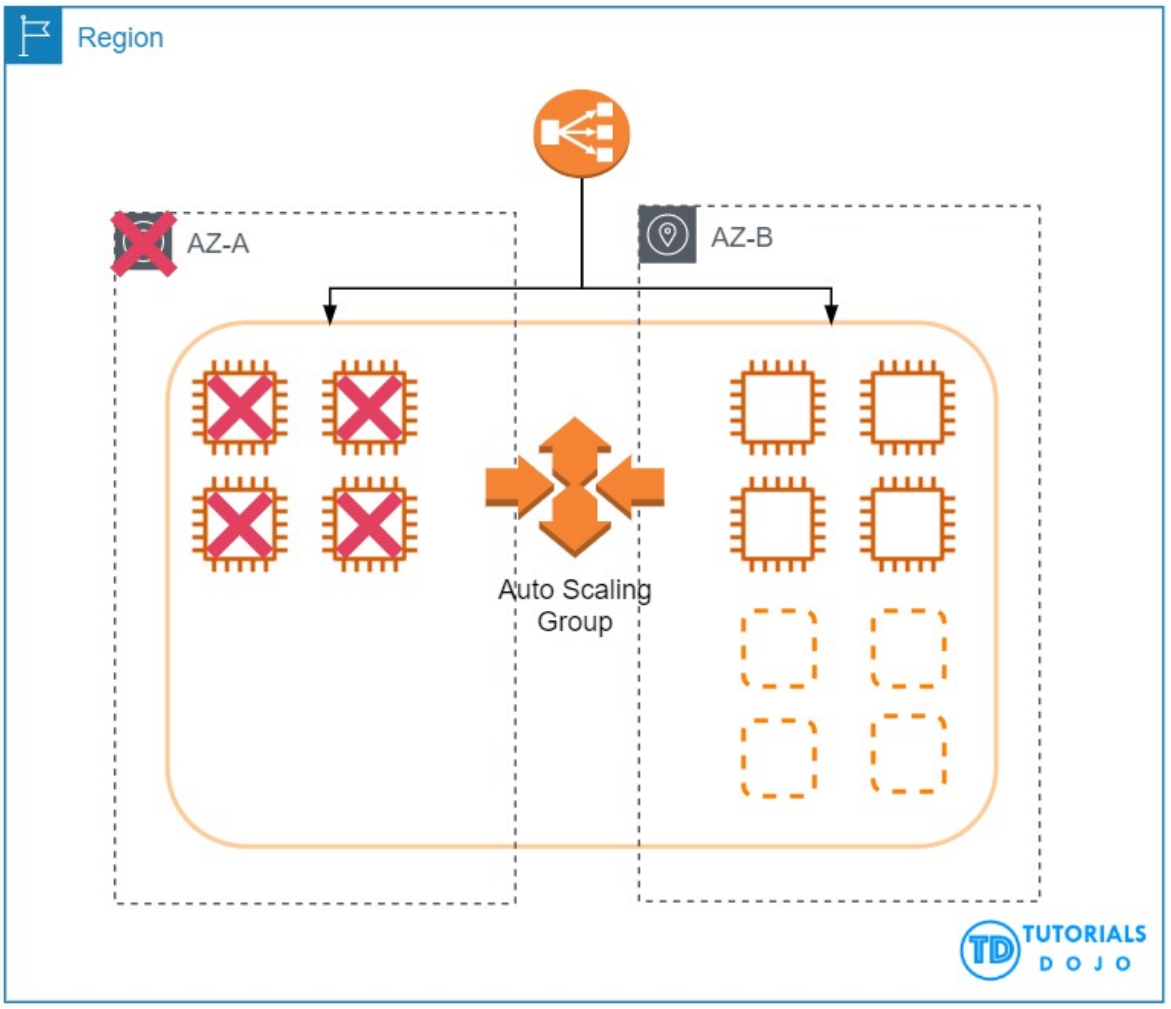


AZ Redundancy

The best option to take is to deploy four EC2 instances in one Availability Zone and four in another availability zone in the same region behind an Amazon Elastic Load Balancer. In this way, if one availability zone goes down, there is still another available zone that can accommodate traffic.

When the first AZ goes down, the second AZ will only have an initial 4 EC2 instances. This will eventually be scaled up to 8 instances since the solution is using Auto Scaling.

The 110% compute capacity for the 4 servers might cause some degradation of the service but not a total outage since there are still some instances that handle the requests. Depending on your scale-up configuration in your Auto Scaling group, the additional 4 EC2 instances can be launched in a matter of minutes.



Instance metadata

Instance metadata is data about your EC2 instance that you can use to configure or manage the running instance. Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

To view the private IPv4 address, public IPv4 address, and all other categories of instance metadata from within a running instance, use the following URL:

`http://169.254.169.254/latest/meta-data/`

Instance store

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host instance.

Instance store is ideal for the temporary storage of information that changes frequently such as buffers, caches, scratch data, and other temporary content, **or for data that is replicated across a fleet of instances**, such as a load-balanced pool of web servers. Instance store volumes are included as part of the instance's usage cost.

As **Instance Store based volumes provide high random I/O performance at low cost** (as the storage is part of the instance's usage cost).

If you create an Amazon Machine Image (AMI) from an instance, the data on its instance store volumes isn't preserved.

Elastic Fabric Adapter (EFA)

An Elastic Fabric Adapter (EFA) is simply an Elastic Network Adapter (ENA) with added capabilities. It provides all of the functionality of an ENA, with additional OS-bypass functionality. **OS-bypass is an access model that allows HPC and machine learning applications to communicate directly with the network interface hardware to provide low-latency, reliable transport functionality.**

An Elastic Fabric Adapter (EFA) is a **network device that you can attach to your Amazon EC2 instance to accelerate High Performance Computing (HPC) and machine learning applications**. EFA enables you to achieve the application performance of an on-premises HPC cluster with the scalability, flexibility, and elasticity provided by the AWS Cloud.

EFA provides lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems. It enhances the performance of inter-instance communication which is critical for scaling HPC and machine learning applications. It is optimized to work on the existing AWS network infrastructure, and it can scale depending on application requirements.

The OS-bypass capabilities of EFAs are not supported on Windows instances. If you attach an EFA to a Windows instance, the instance functions as an Elastic Network Adapter without the added EFA capabilities.

Elastic Network Adapters (ENAs) provide traditional IP networking features that are required to support VPC networking. EFAs provide all of the same traditional IP networking features as ENAs, and they also support OS-bypass capabilities. OS-bypass enables HPC and machine learning applications to bypass the operating system kernel and communicate directly with the EFA device.

AWS Lambda

AWS Lambda supports the synchronous and asynchronous invocation of a Lambda function. You can control the invocation type only when you invoke a Lambda function. When you use an AWS service as a trigger, the invocation type is predetermined for each service. You have no control over the invocation type that these event sources use when they invoke your Lambda function. Since processing only takes 5 minutes, Lambda is also a cost-effective choice.

You can use an AWS Lambda function to process messages in an Amazon Simple Queue Service (Amazon SQS) queue. Lambda event source mappings support standard queues and first-in, first-out (FIFO) queues. With Amazon SQS, you can offload tasks from one component of your application by sending them to a queue and processing them asynchronously.

You can configure a function to connect to a virtual private cloud (VPC) in your account. Use **Amazon Virtual Private Cloud (Amazon VPC)** to create a private network for resources such as databases, cache instances, or internal services. Connect your function to the VPC to access private resources during execution.

AWS Lambda runs your function code securely within a VPC by default. However, to enable your Lambda function to access resources inside your private VPC, you must provide additional VPC-specific configuration information that includes VPC subnet IDs and security group IDs. AWS Lambda uses this information to set up elastic network interfaces (ENIs) that enable your function to connect securely to other resources within your private VPC.

By default, AWS Lambda functions always operate from an AWS-owned VPC and hence have access to any public internet address or public AWS APIs. Once an AWS Lambda function is VPC-enabled, it will need a route through a Network Address Translation gateway (NAT gateway) in a public subnet to access public resources

Lambda functions cannot connect directly to a VPC with dedicated instance tenancy. To connect to resources in a dedicated VPC, peer it to a second VPC with default tenancy.

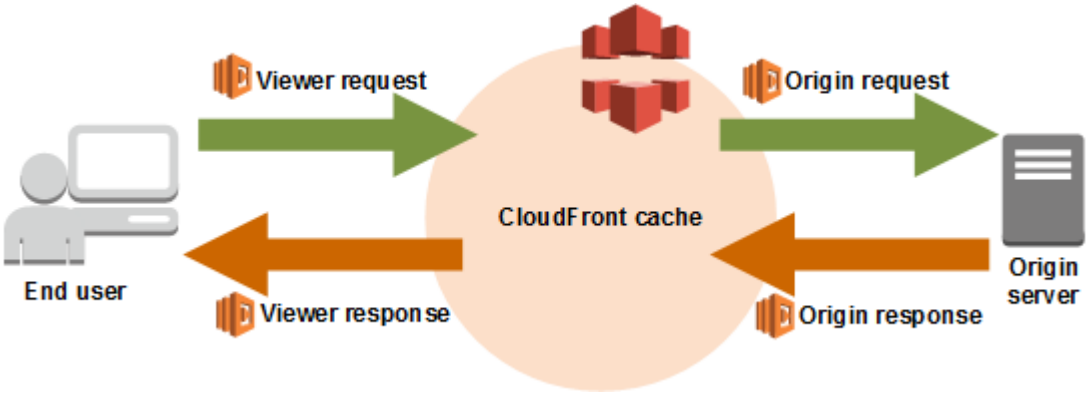
Your Lambda function automatically scales based on the number of events it processes. **If your Lambda function accesses a VPC, you must make sure that your VPC has sufficient ENI capacity** to support the scale requirements of your Lambda function. It is also recommended that you specify at least one subnet in each Availability Zone in your Lambda function configuration.

AWS Lambda currently supports 1000 concurrent executions per AWS account per region. You need to contact AWS support to raise the account limit. Therefore this option is correct.

Lambda@Edge

Lambda@Edge lets you run Lambda functions to customize the content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests and responses at the following points:

- After CloudFront receives a request from a viewer (viewer request)
- Before CloudFront forwards the request to the origin (origin request)
- After CloudFront receives the response from the origin (origin response)
- Before CloudFront forwards the response to the viewer (viewer response)



Elasticache

Redis and Memcached are popular, open-source, in-memory data stores. Although they are both easy to use and offer high performance, there are important differences to consider when choosing an engine. Memcached is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases.

	Memcached	Redis
Sub-millisecond latency	Yes	Yes
Developer ease of use	Yes	Yes
Data partitioning	Yes	Yes
Support for a broad set of programming languages	Yes	Yes
Advanced data structures	-	Yes
Multithreaded architecture	Yes	-
Snapshots	-	Yes
Replication	-	Yes
Transactions	-	Yes
Pub/Sub	-	Yes
Lua scripting	-	Yes
Geospatial support	-	Yes

Elasticache for Redis

Supports geospatial data.

Supports archivial snapshots.

Elasticache for Memcached

Designed for simplicity.

Supports auto-discovery.

Elastic Load Balancing (ELB)

With cross-zone load balancing enabled, one instance in Availability Zone A receives 20% traffic and four instances in Availability Zone B receive 20% traffic each. With cross-zone load balancing disabled, one instance in Availability Zone A receives 50% traffic and four instances in Availability Zone B receive 12.5% traffic each.

By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer.

You can't assign an Elastic IP address to an Application Load Balancer. The alternative method you can do is assign an Elastic IP address to a Network Load Balancer in front of the Application Load Balancer.

A Network Load balancer doesn't have Weighted Target Groups.

Configure the DNS zone apex record to point to the load balancer by creating an A record aliased to the load balancer DNS name.

AWS Elastic Beanstalk

AWS Elastic Beanstalk is an **easy-to-use service for deploying and scaling web applications and services** developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can **simply upload your code and Elastic Beanstalk automatically handles the deployment**, from capacity provisioning, load balancing, and auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

When you create an AWS Elastic Beanstalk environment, you can specify an Amazon Machine Image (AMI) to use instead of the standard Elastic Beanstalk AMI included in your platform version. A custom AMI can improve provisioning times when instances are launched in your environment if you need to install a lot of software that isn't included in the standard AMIs.

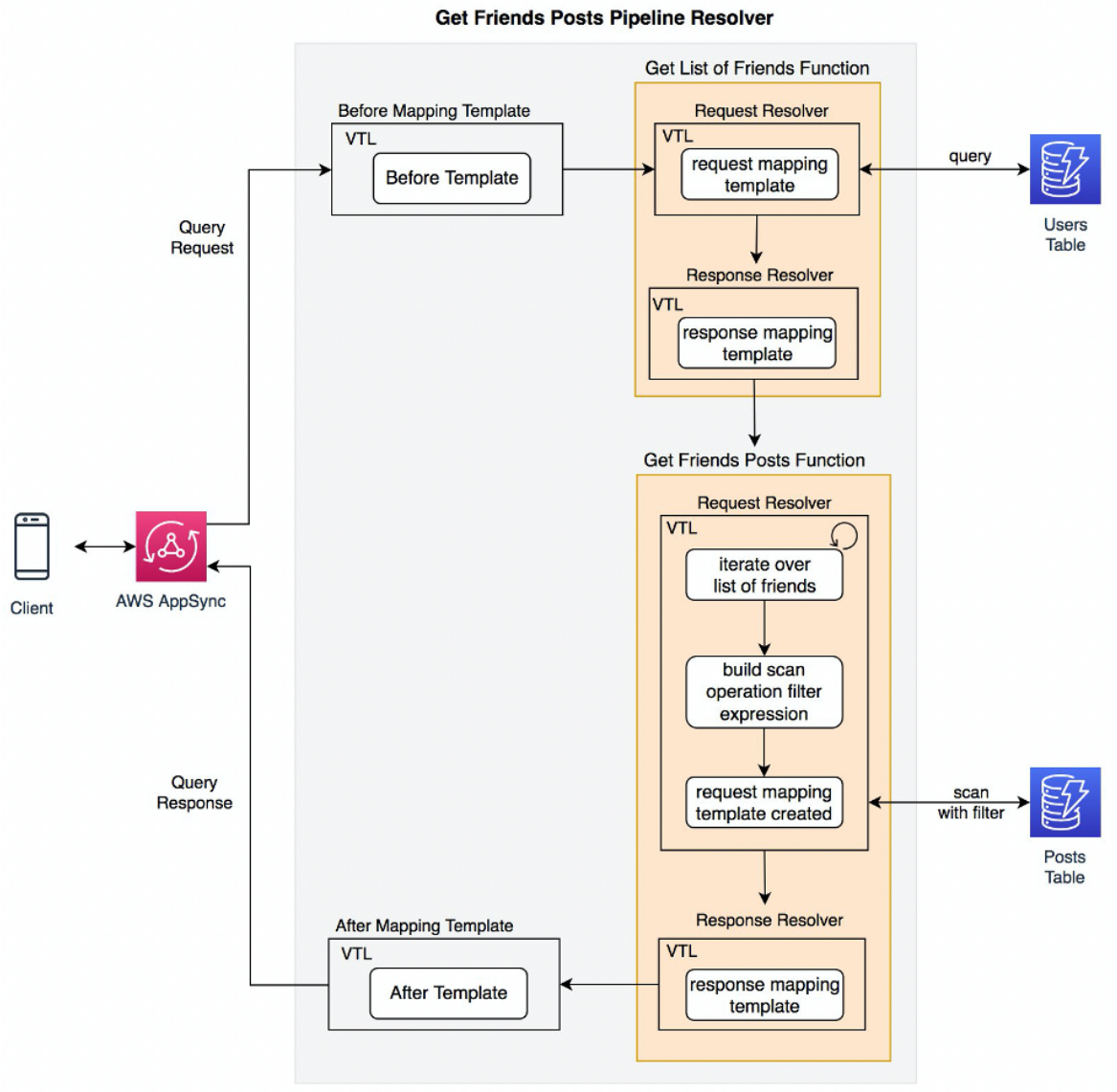
AWS AppSync

AWS AppSync is a serverless GraphQL and Pub/Sub API service that simplifies building modern web and mobile applications. It provides a robust, scalable GraphQL interface for application developers to combine data from multiple sources, including Amazon DynamoDB, AWS Lambda, and HTTP APIs.

AWS AppSync is a managed service that **makes it easy to build scalable APIs that connect applications to data**. Developers use AppSync every day to build GraphQL APIs that interact with data sources like Amazon DynamoDB, AWS Lambda, and HTTP APIs. With AppSync, developers can write their resolvers using JavaScript, and run their code on AppSync’s APPSYNC_JS runtime.

AppSync Pipeline

AppSync pipeline resolvers offer an elegant server-side solution to address the common challenge faced in web applications—aggregating data from multiple database tables. Instead of invoking multiple API calls across different data sources, which can degrade application performance and user experience, AppSync pipeline resolvers enable easy retrieval of data from multiple sources with just a single call. By leveraging Pipeline functions, these resolvers streamline the process of consolidating and presenting data to end-users.



Amazon AppFlow

Amazon AppFlow is a **fully-managed integration service that enables you to securely exchange data between software as a service (SaaS) applications**,

such as Salesforce, and AWS services, such as Amazon Simple Storage Service (Amazon S3) and Amazon Redshift.

Amazon Simple Workflow Service

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to **build applications that coordinate work across distributed components**. In Amazon SWF, a task represents a logical unit of work that is performed by a component of your application. Coordinating tasks across the application involves managing intertask dependencies, scheduling, and concurrency in accordance with the logical flow of the application. Amazon SWF gives you full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state.

When using Amazon SWF, you implement workers to perform tasks. These workers can run either on cloud infrastructure, such as Amazon Elastic Compute Cloud (Amazon EC2), or on your own premises. You can create tasks that are long-running, or that may fail, time out, or require restarts—or that may complete with varying throughput and latency. Amazon SWF stores tasks and assigns them to workers when they are ready, tracks their progress, and maintains their state, including details on their completion. To coordinate tasks, you write a program that gets the latest state of each task from Amazon SWF and uses it to initiate subsequent tasks. Amazon SWF maintains an application's execution state durably so that the application is resilient to failures in individual components. With Amazon SWF, you can implement, deploy, scale, and modify these application components independently.

Amazon SWF provides useful guarantees around task assignments. It **ensures that a task is never duplicated and is assigned only once**. Thus, even though you may have multiple workers for a particular activity type (or a number of instances of a decider), Amazon SWF will give a specific task to only one worker (or one decider instance). Additionally, Amazon SWF keeps at most one decision task outstanding at a time for workflow execution. Thus, you can run multiple decider instances without worrying about two instances operating on the same execution simultaneously. These facilities enable you to coordinate your workflow without worrying about duplicate, lost, or conflicting tasks.

EBS Volume

An Amazon EBS volume is a durable, block-level storage device that you can attach to a **single EC2 instance in the same AZ**. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

Amazon EBS provides three volume types to best meet the needs of your workloads: **General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic**.

Baseline I/O performance for General Purpose SSD storage is 3 IOPS for each GiB. For 334 GiB of storage, the baseline performance would be 1,002 IOPS. Additionally, General Purpose SSD storage is more cost-effective than Provisioned IOPS storage.

General Purpose (SSD) is the new, SSD-backed, general purpose EBS volume type that is recommended as the default choice for customers. General Purpose (SSD) volumes are suitable for a broad range of workloads, including small to medium-sized databases, development and test environments, and boot volumes.

Provisioned IOPS (SSD) volumes offer storage with consistent and low-latency performance and are designed for I/O intensive applications such as large relational or NoSQL databases.

Magnetic volumes are ideal for workloads where data are accessed infrequently, and applications where the lowest storage cost is important. Take note that this is a Previous Generation Volume. The latest low-cost magnetic storage types are Cold HDD (sc1) and Throughput Optimized HDD (st1) volumes.

The multi-attach feature can only be enabled on EBS Provisioned IOPS io2 or io1 volumes. In addition, **multi-attach won't offer multi-az resiliency** because this feature only allows an EBS volume to be attached on multiple instances **within an availability zone**.

You can set the `DeleteOnTermination` attribute to False when you launch a new instance. It is not possible to update this attribute of a running instance from the AWS console. Must be done with the AWS CLIe

Encryption by default is a Region-specific setting. If you enable it for a Region, you cannot disable it for individual volumes or snapshots in that Region.

Here is a list of important information about EBS Volumes:

- When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to a failure of any single

hardware component.

- An EBS volume can only be attached to one EC2 instance at a time.
- After you create a volume, you can attach it to any EC2 instance in the same Availability Zone
- An EBS volume is off-instance storage that can persist independently from the life of an instance. You can specify not to terminate the EBS volume when you terminate the EC2 instance during instance creation.
- EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.
- Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256)
- EBS Volumes offer 99.999% SLA.

RAID 0

RAID 0 configuration enables you to improve your storage volumes performance by distributing the I/O across the volumes in a stripe. Therefore, if you add a storage volume, you get the straight addition of throughput and IOPS. This configuration can be implemented on both EBS or instance store volumes. Since the main requirement in the scenario is storage performance, you need to use an instance store volume. It uses NVMe or SATA-based SSD to deliver high random I/O performance. This type of storage is a good option when you need storage with very low latency and you don't need the data to persist when the instance terminates.

RAID 1

Mirroring

Provisioned IOPS SSD `io1` volumes

Provisioned IOPS SSD (`io1`) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike `gp2`, which uses a bucket and credit model to calculate performance, an `io1` volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

An `io1` volume can range in size **from 4 GiB to 16 TiB**. You can provision from 100 IOPS **up to 64,000 IOPS per volume on Nitro** system instance families and **up to 32,000 on other instance families**. The maximum ratio of provisioned IOPS to the requested volume size **(in GiB) is 50:1**.

For example, a 100 GiB volume can be provisioned with up to 5,000 IOPS. On a supported instance type, any volume 1,280 GiB in size or greater allows provisioning up to the 64,000 IOPS maximum (50 × 1,280 GiB = 64,000).

An io1 volume provisioned with up to 32,000 IOPS supports a maximum I/O size of 256 KiB and yields as much as 500 MiB/s of throughput. With the I/O size at the maximum, peak throughput is reached at 2,000 IOPS. A volume provisioned with more than 32,000 IOPS (up to the cap of 64,000 IOPS) supports a maximum I/O size of 16 KiB and yields as much as 1,000 MiB/s of throughput.

Therefore, for instance, a 10 GiB volume can be provisioned with up to 500 IOPS. Any volume 640 GiB in size or greater allows provisioning up to a maximum of 32,000 IOPS (50 × 640 GiB = 32,000).

The volume queue length is the number of pending I/O requests for a device. Latency is the true end-to-end client time of an I/O operation, in other words, the time elapsed between sending an I/O to EBS and receiving an acknowledgment from EBS that the I/O read or write is complete. Queue length must be correctly calibrated with I/O size and latency to avoid creating bottlenecks either on the guest operating system or on the network link to EBS.

Optimal queue length varies for each workload, depending on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to fully use the performance available to your EBS volume, then your volume might not deliver the IOPS or throughput that you have provisioned.

Transaction-intensive applications are sensitive to increased I/O latency and are well-suited for SSD-backed io1 and gp2 volumes. You can maintain high IOPS while keeping latency down by maintaining a low queue length and a high number of IOPS available to the volume. Consistently driving more IOPS to a volume than it has available can cause increased I/O latency.

Throughput-intensive applications are less sensitive to increased I/O latency and are well-suited for HDD-backed st1 and sc1 volumes. You can maintain high throughput to HDD-backed volumes by maintaining a high queue length when performing large, sequential I/O.

Encryption

Amazon EBS encryption offers seamless encryption of EBS data volumes, boot volumes, and snapshots, eliminating the need to build and maintain a secure key management infrastructure. **EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys, or keys you create and manage using the AWS Key Management Service (KMS)**. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage.

EBS volumes are not encrypted by default. To encrypt an unencrypted EBS volume you must create an EBS snapshot, encrypt it, create a new EBS volume from the snapshot.

For an encrypted volume, the snapshots are automatically encrypted and all the data moving between the volume and the instance is encrypted.

Amazon RDS

Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from CloudWatch Logs in a monitoring system of your choice. By default, Enhanced Monitoring metrics are stored in the CloudWatch Logs for 30 days. To modify the amount of time the metrics are stored in the CloudWatch Logs, change the retention for the RDSOSMetrics log group in the CloudWatch console.

You can use **Secure Socket Layer / Transport Layer Security (SSL/TLS) connections to encrypt data in transit**. Amazon RDS creates an SSL certificate and installs the certificate on the DB instance when the instance is provisioned. For MySQL, you launch the MySQL client using the --ssl_ca parameter to reference the public key to encrypt connections. Using SSL, you can encrypt a PostgreSQL connection between your applications and your PostgreSQL DB instances. You can also force all connections to your PostgreSQL DB instance to use SSL.

Snapshots are the cheapest way to avoid incurring costs for a DB that is not used all the time (e.g. once a month). A stopped database has storage costs.

Snapshot cannot be directly accessed through S3, Amazon stores these in your behalf. You can still share a snapshot with another AWS Account.

You can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created. However, because you can encrypt a copy of an unencrypted DB snapshot, you can effectively add

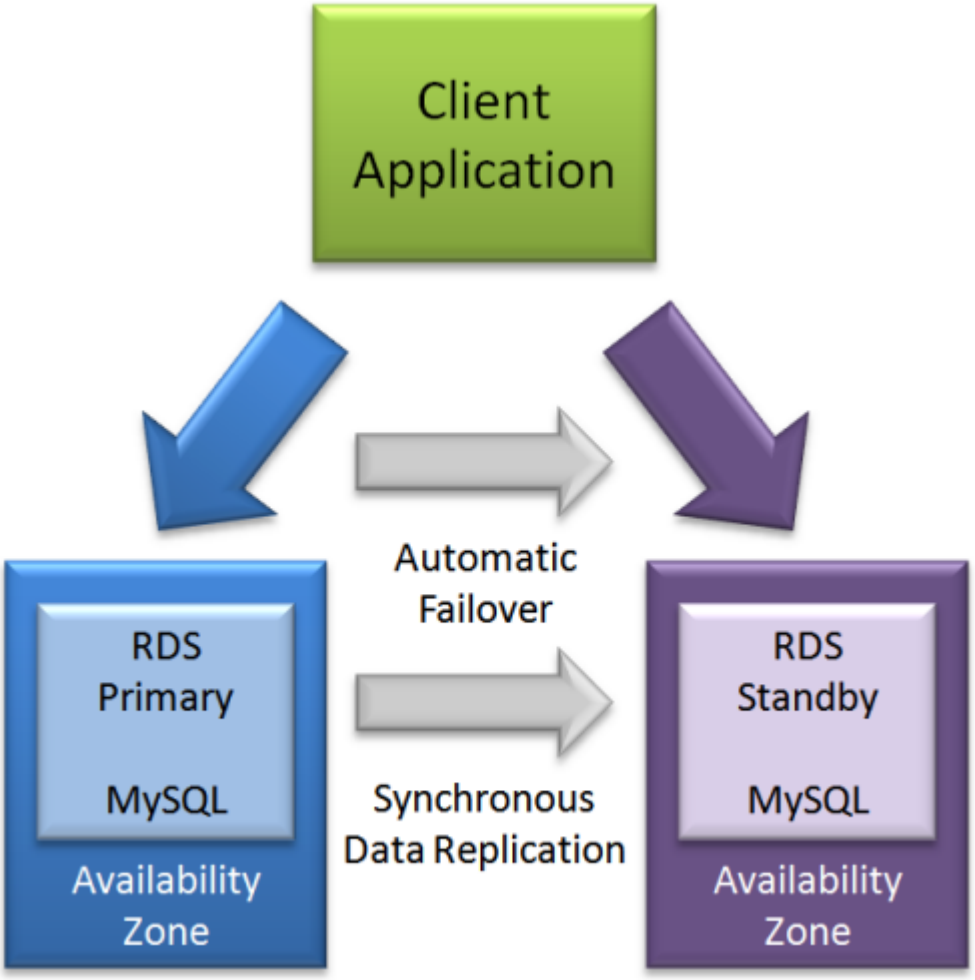
encryption to an unencrypted DB instance. That is, you can create a snapshot of your DB instance, and then create an encrypted copy of that snapshot. And then you restore the database.

If your workload is unpredictable, you can enable storage autoscaling for an Amazon RDS DB instance.

RDS Multi-AZ

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) Instances, making them a natural fit for production database workloads. When you provision a **Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ)**. Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.

In Amazon RDS, failover is automatically handled so that you can resume database operations as quickly as possible without administrative intervention in the event that your primary database instance goes down. **When failing over, Amazon RDS simply flips the canonical name record (CNAME) for your DB instance to point at the standby, which is in turn promoted to become the new primary.**



It also provides increased database availability in the case of system upgrades like OS patching or DB Instance scaling.

For example, with automated backups, I/O activity is no longer suspended on your primary during your preferred backup window, since **backups are taken from the standby**. In the case of **patching or DB instance class scaling, these operations occur first on the standby, prior to automatic failover**. As a result, your availability impact is limited to the time required for automatic failover to complete.

Read Replicas

Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This feature makes 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. Read replicas can also be promoted when needed to become standalone DB instances.

You can create up to five read replicas from one DB instance. For replication to operate effectively, each read replica should have the same amount of compute and storage resources as the source database instance. If you scale the source database instance, also scale the read replicas.

RDS Read Replicas can only provide asynchronous replication in seconds and not in milliseconds.

For the MySQL, MariaDB, PostgreSQL, and Oracle database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance. **It then uses the engines' native asynchronous replication to update the read replica** whenever there is a change to the source DB instance. The read replica operates as a DB instance that allows only read-only connections; applications can connect to a read replica just as they would to any DB instance. Amazon RDS replicates all databases in the source DB instance.

When you create a read replica for Amazon RDS for MySQL, MariaDB, PostgreSQL, and Oracle, Amazon RDS sets up a secure communications channel using public-key encryption between the source DB instance and the read replica, even when replicating across regions. Amazon RDS establishes any AWS security configurations, such as adding security group entries needed to enable the secure channel.

You can also create read replicas within a Region or between Regions for your Amazon RDS for MySQL, MariaDB, PostgreSQL, and Oracle database instances encrypted at rest with AWS Key Management Service (KMS).

Multi-AZ deployments	Multi-Region deployments	Read replicas
Main purpose is high availability	Main purpose is disaster recovery and local performance	Main purpose is scalability
Non-Aurora: synchronous replication; Aurora: asynchronous replication	Asynchronous replication	Asynchronous replication
Non-Aurora: only the primary instance is active; Aurora: all instances are active	All regions are accessible and can be used for reads	All read replicas are accessible and can be used for readscaling
Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer	Automated backups can be taken in each region	No backups configured by default
Always span at least two Availability Zones within a single region	Each region can have a Multi-AZ deployment	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent from source instance; Aurora: all instances are updated together
Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected	Aurora allows promotion of a secondary region to be the master	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)

Enhanced Monitoring

Enhanced monitoring gets data from an **agent on the instance**.

In RDS, the Enhanced Monitoring metrics shown in the **Process List** view are organized as follows:

RDS child processes – Shows a summary of the RDS processes that support the DB instance, for example aurora for Amazon Aurora DB clusters and mysqld for MySQL DB instances. Process threads appear nested beneath the parent process. Process threads show CPU utilization only as other metrics are the same for all threads for the process. The console displays a maximum of 100 processes and threads. The results are a combination of the top CPU-consuming and memory-consuming processes and threads. If there are more than 50 processes and more than 50 threads, the console displays the top 50 consumers in each category. This display helps you identify which processes are having the greatest impact on performance.

RDS processes – Shows a summary of the resources used by the RDS management agent, diagnostics monitoring processes, and other AWS processes that are required to support RDS DB instances.

OS processes – Shows a summary of the kernel and system processes, which generally have minimal impact on performance.

Amazon Aurora

Amazon Aurora typically involves a **cluster** of DB instances instead of a single instance. Each connection is handled by a specific DB instance. When you connect to an Aurora cluster, the host name and port that you specify point to an intermediate handler called an endpoint. **Aurora uses the endpoint mechanism to abstract these connections**. Thus, you don't have to hardcode all the hostnames or write your own logic for load-balancing and rerouting connections when some DB instances aren't available.

For certain Aurora tasks, different instances or groups of instances perform different roles. For example, t**he primary instance handles all data definition language (DDL) and data manipulation language (DML) statements**. Up to 15 Aurora Replicas handle read-only query traffic.

Using endpoints, you can map each connection to the appropriate instance or group of instances based on your use case. For example, to perform DDL statements you can connect to whichever instance is the primary instance. **To perform queries, you can connect to the reader endpoint, with Aurora automatically performing load-balancing among all the Aurora Replicas**. For clusters with DB instances of different capacities or configurations, you can connect to custom endpoints associated with different subsets of DB instances. For diagnosis or tuning, you can connect to a specific instance endpoint to examine details about a specific DB instance.

The custom endpoint provides load-balanced database connections based on criteria other than the read-only or read-write capability of the DB instances. For example, you might define a custom endpoint to connect to instances that use a particular AWS instance class or a particular DB parameter group. Then you might tell particular groups of users about this custom endpoint. For example, you might direct internal users to low-capacity instances for report generation or ad hoc (one-time) querying, and direct production traffic to high-capacity instances.

You can invoke an AWS Lambda function from an Amazon Aurora MySQL-Compatible Edition DB cluster with a native function or a stored procedure. This approach can be useful when you want to integrate your database running on Aurora MySQL with other AWS services. For example, you might want to capture data changes whenever a row in a table is modified in your database.

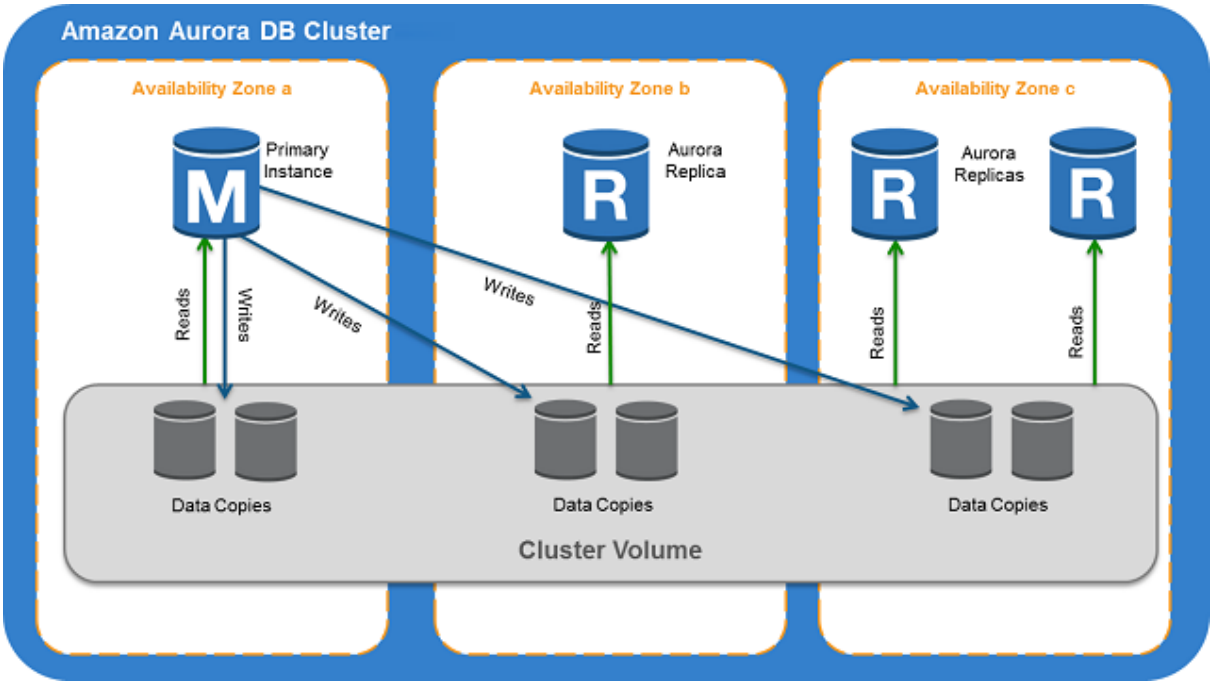
Two types of DB instances make up an Aurora DB cluster:

- **Primary DB instance** – Supports read and write operations, and performs all of the data modifications to the cluster volume. Each Aurora DB cluster has one primary DB instance.
- **Aurora Replica** – Connects to the same storage volume as the primary DB instance and supports only read operations. Each Aurora DB cluster can have up to 15 Aurora Replicas in addition to the primary DB instance. **Aurora automatically fails over to an Aurora Replica in case the primary DB instance becomes unavailable**. You can specify the failover priority for Aurora Replicas. Aurora Replicas can also offload read workloads from the primary DB instance.

Multi-AZ deployments	Multi-Region deployments	Read replicas
Main purpose is high availability	Main purpose is disaster recovery and local performance	Main purpose is scalability
Non-Aurora: synchronous replication; Aurora: asynchronous replication	Asynchronous replication	Asynchronous replication
Non-Aurora: only the primary instance is active; Aurora: all instances are active	All regions are accessible and can be used for reads	All read replicas are accessible and can be used for read scaling
Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer	Automated backups can be taken in each region	No backups configured by default
Always span at least two Availability Zones within a single region	Each region can have a Multi-AZ deployment	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent from source instance; Aurora: all instances are updated together
Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected	Aurora allows promotion of a secondary region to be the primary	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)

Replicas

Amazon Aurora Replicas have two main purposes. You can issue queries to them to scale the read operations for your application. You typically do so by connecting to the **reader endpoint of the cluster**. That way, Aurora can spread the load for read-only connections across as many Aurora Replicas as you have in the cluster. Amazon Aurora Replicas also help to increase availability. **If the writer instance in a cluster becomes unavailable, Aurora automatically promotes one of the reader instances to take its place as the new writer**. **Up to 15 Aurora Replicas can be distributed across the Availability Zones (AZs) that a DB cluster spans within an AWS Region.**



Failover

Failover is automatically handled by Amazon Aurora so that your applications can resume database operations as quickly as possible without manual administrative intervention.

If you have an Amazon Aurora Replica in the same or a different Availability Zone, **when failing over, Amazon Aurora flips the canonical name record (CNAME) for your DB Instance to point at the healthy replica, which in turn is promoted to become the new primary**. Start-to-finish failover typically completes within 30 seconds.

If you are running Aurora Serverless and the DB instance or AZ becomes unavailable, Aurora will automatically recreate the DB instance in a different AZ.

If you do not have an Amazon Aurora Replica (i.e., single instance) and are not running Aurora Serverless, Aurora will attempt to create a new DB Instance in the same Availability Zone as the original instance. This replacement of the original instance is done on a best-effort basis and may not succeed, for example, if there is an issue that is broadly affecting the Availability Zone.

DynamoDB

DynamoDB is durable, scalable, and highly available data store which can be used for real-time tabulation. You can also use **AppSync** with DynamoDB to make it easy

for you to build collaborative apps that keep shared data updated in real-time. You just specify the data for your app with simple code statements and AWS AppSync manages everything needed to keep the app data updated in real-time. This will allow your app to access data in Amazon DynamoDB, trigger AWS Lambda functions, or run Amazon OpenSearch Service queries and combine data from these services to provide the exact data you need for your app.

Amazon DynamoDB has two read/write capacity modes for processing reads and writes on your tables:
- On-demand
- Provisioned (default, free-tier eligible)

Amazon DynamoDB on-demand is a flexible billing option capable of serving thousands of requests per second without capacity planning. DynamoDB on-demand offers pay-per-request pricing for read and write requests so that you pay only for what you use.

The on-demand mode is a good option if any of the following are true:
- You create new tables with unknown workloads.
- You have unpredictable application traffic.
- You prefer the ease of paying for only what you use.

If you choose provisioned mode, you specify the number of reads and writes per second that you require for your application. You can use auto-scaling to adjust your table's provisioned capacity automatically in response to traffic changes. This helps you govern your DynamoDB use to stay at or below a defined request rate to obtain cost predictability.

Provisioned mode is a good option if any of the following are true:
- You have predictable application traffic.
- You run applications whose traffic is consistent or ramps gradually.
- You can forecast capacity requirements to control costs.

By default, all Amazon DynamoDB tables are encrypted using AWS owned keys, which do not write to AWS CloudTrail logs.

DynamoDB auto scaling

DynamoDB auto scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns. This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic, without

throttling. When the workload decreases, Application Auto Scaling decreases the throughput so that you don't pay for unused provisioned capacity.

Endpoints

Since DynamoDB tables are public resources, applications within a VPC rely on an Internet Gateway to route traffic to/from Amazon DynamoDB. You can use a Gateway endpoint if you want to keep the traffic between your VPC and Amazon DynamoDB within the Amazon network. This way, resources residing in your VPC can use their private IP addresses to access DynamoDB with no exposure to the public internet.

When you create a DynamoDB Gateway endpoint, you specify the VPC where it will be deployed as well as the route table that will be associated with the endpoint. The route table will be updated with an Amazon DynamoDB prefix list (list of CIDR blocks) as the destination and the endpoint's ID as the target.

Backups

DynamoDB on-demand backups are available at no additional cost beyond the normal pricing that's associated with backup storage size. DynamoDB on-demand backups cannot be copied to a different account or Region. To create backup copies across AWS accounts and Regions and for other advanced features, you should use AWS Backup.

With **AWS Backup**, you can configure backup policies and monitor activity for your AWS resources and on-premises workloads in one place. Using DynamoDB with AWS Backup, you can copy your on-demand backups across AWS accounts and Regions, add cost allocation tags to on-demand backups, and transition on-demand backups to cold storage for lower costs. To use these advanced features, you must opt into AWS Backup. Opt-in choices apply to the specific account and AWS Region, so you might have to opt into multiple Regions using the same account.

DynamoDB Accelerator (DAX)

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement – from milliseconds to microseconds – even at millions of requests per second. DAX does all the heavy lifting required to add in-memory acceleration to your DynamoDB tables, without requiring developers to manage cache invalidation, data population, or cluster management.

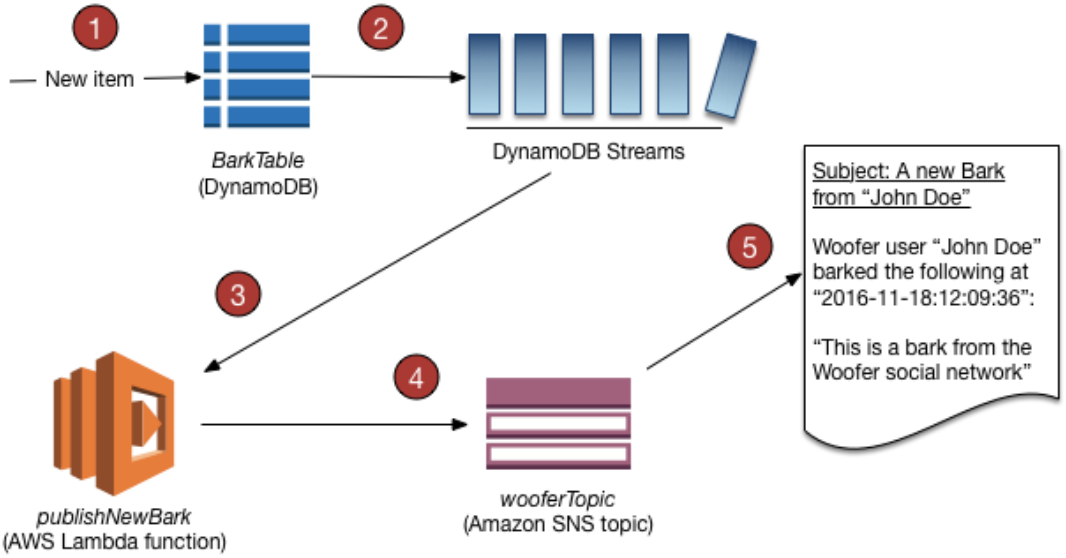
Streams

A DynamoDB stream is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attribute(s) of the items that were modified. A stream record contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the "before" and "after" images of modified items.

Amazon DynamoDB is integrated with AWS Lambda so that you can create triggers—pieces of code that automatically respond to events in DynamoDB Streams. With triggers, you can build applications that react to data modifications in DynamoDB tables.

If you enable DynamoDB Streams on a table, you can **associate the stream ARN with a Lambda function that you write**. Immediately after an item in the table is modified, a new record appears in the table's stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records. The Lambda function can perform any actions you specify, such as sending a notification or initiating a workflow.



Amazon Neptune

Amazon Neptune is a fast, fully managed database service powering graph use cases such as identity graphs, knowledge graphs, and fraud detection.

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) is a cloud service that makes it easy to migrate relational databases, data warehouses, NoSQL databases, and other types of data stores. You can use AWS DMS to migrate your data into the AWS Cloud or between combinations of cloud and on-premises setups.

You can also leverage AWS Database Migration Service (AWS DMS) as a bridge between Amazon S3 and Amazon Kinesis Data Streams.

AWS Database Migration Service helps you migrate your databases to AWS with **virtually no downtime**. **All data changes to the source database that occur during the migration are continuously replicated to the target, allowing the source database to be fully operational during the migration process.**

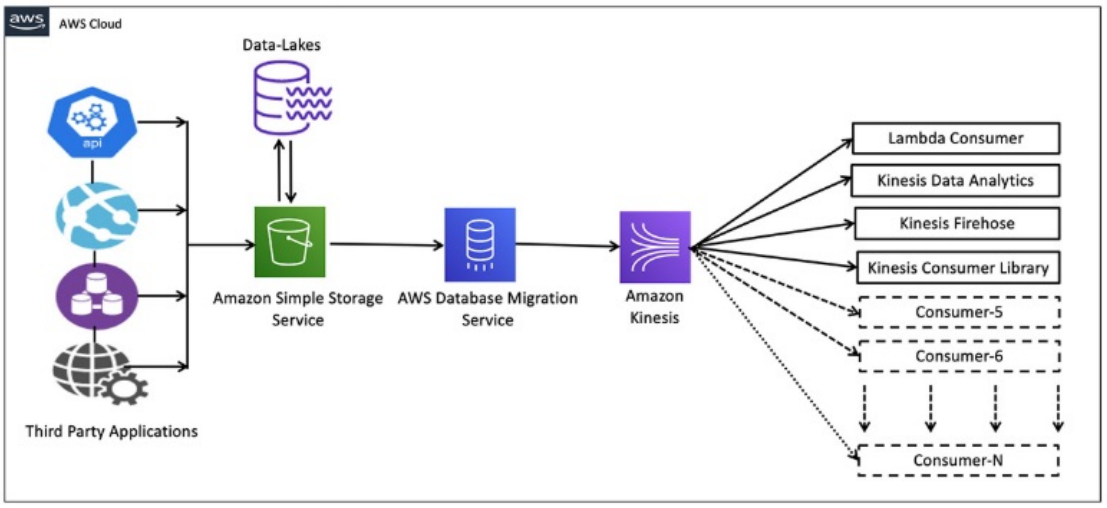
With AWS DMS, you can perform one-time migrations, and you can replicate ongoing changes to keep sources and targets in sync. **If you want to migrate to a different database engine, you can use the AWS Schema Conversion Tool (AWS SCT)** to translate your database schema to the new platform. You then use AWS DMS to migrate the data. Because AWS DMS is a part of the AWS Cloud, you get the cost efficiency, speed to market, security, and flexibility that AWS services offer.

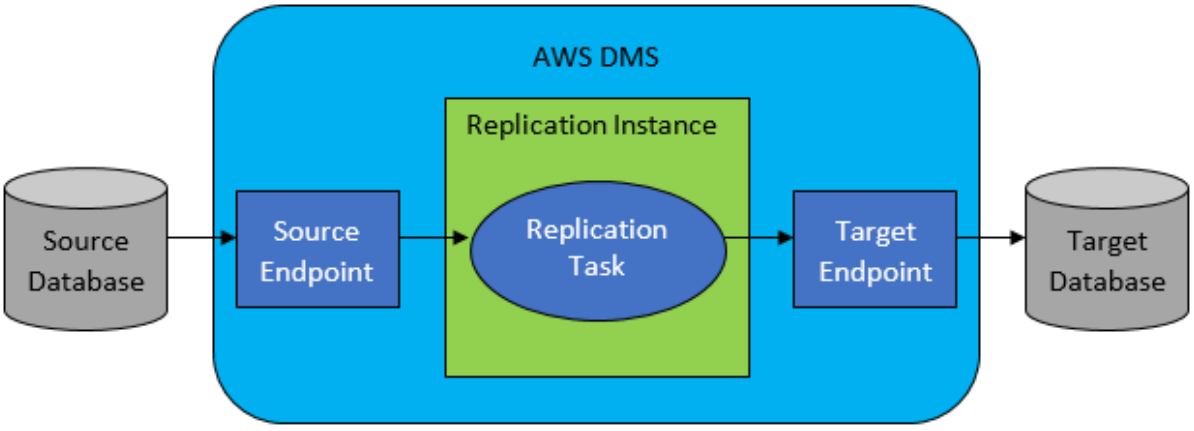
You can migrate data to Amazon S3 using AWS DMS from any of the supported database sources. When using Amazon S3 as a target in an AWS DMS task, both **full load and change data capture (CDC)** data is written to comma-separated value (.csv) format by default.

You can encrypt connections for source and target endpoints by using Secure Sockets Layer (SSL). To do so, you can use the AWS DMS Management Console or AWS DMS API to assign a certificate to an endpoint. You can also use the AWS DMS console to manage your certificates.

Not all databases use SSL in the same way. Amazon Aurora MySQL-Compatible Edition uses the server name, the endpoint of the primary instance in the cluster, as the endpoint for SSL. An Amazon Redshift endpoint already uses an SSL connection and does not require an SSL connection set up by AWS DMS.

The following diagram shows the architecture of this solution.





AWS Elastic Disaster Recovery (DRS)

AWS Elastic Disaster Recovery (AWS DRS) provides continuous block-level replication, recovery orchestration, and automated server conversion capabilities. These allow customers to achieve a crash-consistent recovery point objective (RPO) of seconds, and a recovery time objective (RTO) typically ranging between 5–20 minutes.

Disaster Recovery strategies

- **Backup and restore** (RPO in hours, RTO in 24 hours or less): Back up your data and applications using point-in-time backups into the DR Region. Restore this data when necessary to recover from a disaster.
- **Pilot light** (RPO in minutes, RTO in hours): Replicate your data from one region to another and provision a copy of your core workload infrastructure. Resources required to support data replication and backup such as databases and object storage are always on. Other elements such as application servers are loaded with application code and configurations, but are switched off and are only used during testing or when Disaster Recovery failover is invoked.
- **Warm standby** (RPO in seconds, RTO in minutes): Maintain a scaled-down but fully functional version of your workload always running in the DR Region. Business-critical systems are fully duplicated and are always on, but with a scaled down fleet. When the time comes for recovery, the system is scaled up quickly to handle the production load. The more scaled-up the Warm Standby is, the lower RTO and control plane reliance will be. When scaled up to full scale this is known as a **Hot Standby**.
- **Multi-region (multi-site) active-active** (RPO near zero, RTO potentially zero): Your workload is deployed to, and actively serving traffic from, multiple AWS Regions. This strategy requires you to synchronize data across Regions. Possible conflicts caused by writes to the same record in two different regional replicas must be avoided or handled. Data replication is useful for data synchronization and will protect you against some types of disaster, but it will not protect you against data corruption or destruction unless your solution also includes options for point-in-time recovery. Use services like Amazon Route 53 or AWS Global Accelerator to route your user traffic to where your workload is healthy. For more details on AWS services you can use for active-active architectures see the **AWS Regions** section of Use Fault Isolation to Protect Your Workload.

Backup and Restore

In most traditional environments, data is backed up to tape and sent off-site regularly. If you use this method, it can take a long time to restore your system in the event of a disruption or disaster. Amazon S3 is an ideal destination for backup data that might be needed quickly to perform a restore. Transferring data to and from Amazon S3 is typically done through the network and is therefore accessible from any location. There are many commercial and open-source backup solutions that integrate with Amazon S3. You can use AWS Import/Export to transfer very large data sets by shipping storage devices directly to AWS. For longer-term data storage where retrieval times of several hours are adequate, there is Amazon Glacier, which has the same durability model as Amazon S3. Amazon Glacier is a low-cost alternative starting from $0.01/GB per month. Amazon Glacier and Amazon S3 can be used in conjunction to produce a tiered backup solution. Even though Backup and Restore method is cheaper, it has an RPO in hours.

Pilot Light

The term pilot light is often used to describe a DR scenario in which a minimal version of an environment is always running in the cloud. The idea of the pilot light is an analogy that comes from the gas heater. In a gas heater, a small flame that’s always on can quickly ignite the entire furnace to heat up a house. This scenario is similar to a backup-and-restore scenario. For example, with AWS you can maintain a pilot light by configuring and running the most critical core elements of your system in AWS. For the given use-case, a small part of the backup infrastructure is always running simultaneously syncing mutable data (such as databases or documents) so

that there is no loss of critical data. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core. For Pilot light, RPO/RTO is in 10s of minutes.

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 AWS and have them always on. This option is costlier compared to Pilot Light.

Multi-Site

A multi-site solution runs on AWS as well as on your existing on-site infrastructure in an active-active configuration. The data replication method that you employ will be determined by the recovery point that you choose, either Recovery Time Objective (the maximum allowable downtime before degraded operations are restored) or Recovery Point Objective (the maximum allowable time window whereby you will accept the loss of transactions during the DR process). This option is more costly compared to Pilot Light or Warm Standby.

Amazon Redshift

Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large-scale data set storage and analysis.

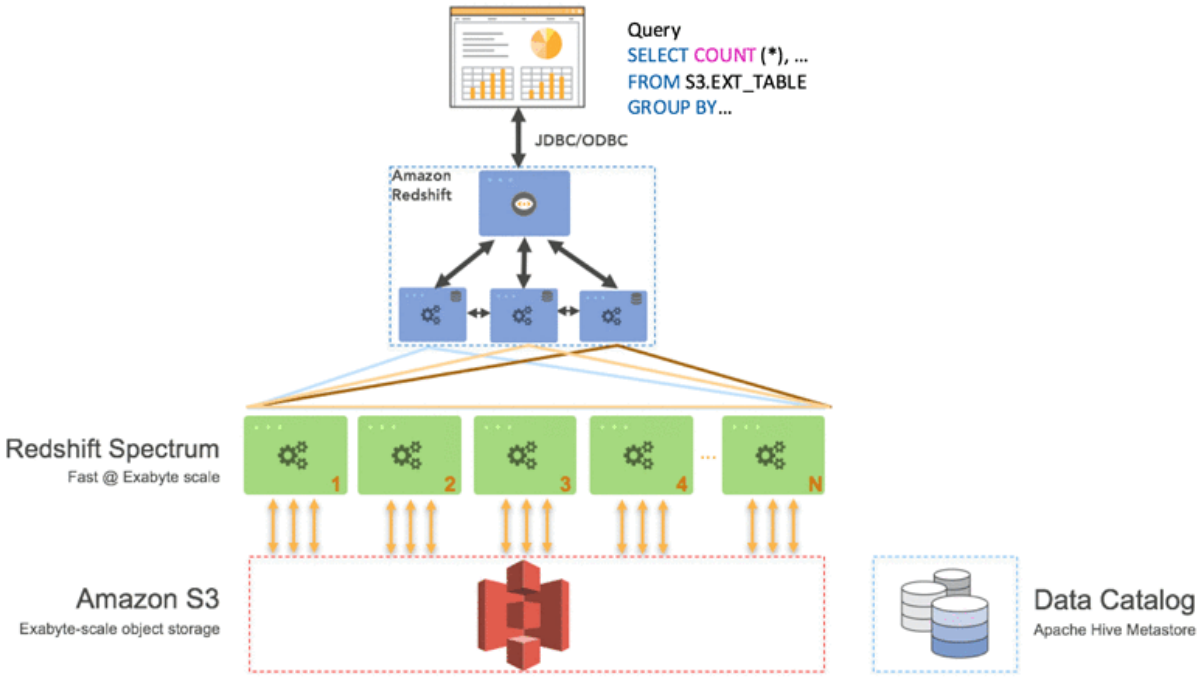
You can configure Amazon Redshift to copy snapshots for a cluster to another region. To configure **cross-region snapshot copy**, you need to enable this copy feature for each cluster and configure where to copy snapshots and how long to keep copied automated snapshots in the destination region. When a cross-region copy is enabled for a cluster, all new manual and automatic snapshots are copied to the specified region.

Redshift Spectrum

Use Amazon Redshift Spectrum to create Amazon Redshift cluster tables pointing to the underlying historical data in Amazon S3. Using Amazon Redshift Spectrum, you can efficiently query and retrieve structured and

semistructured data from files in Amazon S3 without having to load the data into Amazon Redshift tables.

Amazon Redshift Spectrum resides on dedicated Amazon Redshift servers that are independent of your cluster. Redshift Spectrum pushes many compute-intensive tasks, such as predicate filtering and aggregation, down to the Redshift Spectrum layer. Thus, Amazon Redshift Spectrum queries use much less of your cluster's processing capacity than other queries.



Amazon S3

Amazon S3 stores data as objects within buckets. An object is a file and any optional metadata that describes the file. To store a file in Amazon S3, you upload it to a bucket. When you upload a file as an object, you can set permissions on the object and any metadata. Buckets are containers for objects. You can have one or more buckets. You can control access for each bucket, deciding who can create, delete, and list objects in it. You can also choose the geographical region where Amazon S3 will store the bucket and its contents and view access logs for the bucket and its objects.

There are no S3 data transfer charges when data is transferred in from the internet.

Metadata, which can be included with the object, is not encrypted while being stored on Amazon S3.

Amazon S3 can protect data at rest using Client-Side Encryption or Server-Side Encryption.

Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. **Amazon S3TA takes advantage of Amazon CloudFront's globally distributed edge locations.** As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path. **You only pay for trasfers that are accelerated**.

Multipart upload allows you to upload a single object as a set of parts. Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order. If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. In general, when your object size reaches 100 MB, you should consider using multipart uploads instead of uploading the object in a single operation. Multipart upload provides improved throughput, therefore it facilitates faster file uploads.

Bucket policies in Amazon S3 can be used to add or deny permissions across some or all of the objects within a single bucket. Policies can be attached to users, groups, or Amazon S3 buckets, enabling centralized management of permissions. With bucket policies, you can grant users within your AWS Account or other AWS Accounts access to your Amazon S3 resources.

If you need server-side encryption for all of the objects that are stored in a bucket, use a bucket policy. You can create a bucket policy that denies permissions to upload an object unless the request includes the x-amz-server-side-encryption header to request server-side encryption.

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. **Once you version-enable a bucket, it can never return to an unversioned state. Versioning can only be suspended once it has been enabled. When you apply a retention period to an object version explicitly, you specify a `Retain Until Date` for the object version. Different versions of a single object can have different retention modes and periods.**

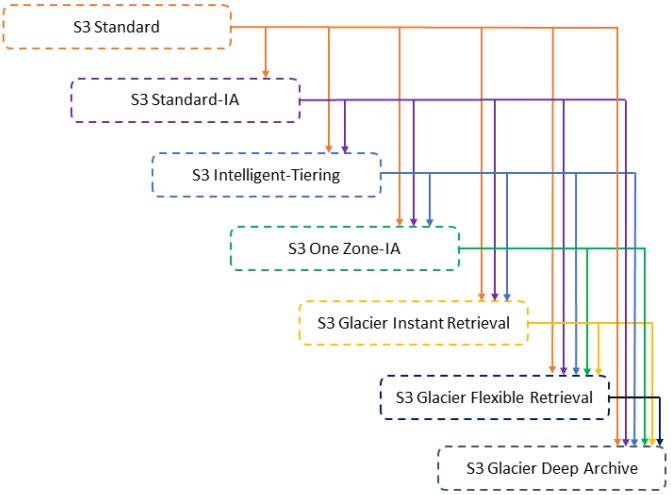
Amazon S3 delivers **strong read-after-write consistency automatically**, without changes to performance or availability, without sacrificing regional isolation for applications, and at no additional cost.

Your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per partitioned Amazon S3 prefix.

The Amazon S3 notification feature enables you to receive notifications when certain events happen in your bucket. To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

Amazon S3 supports the following destinations where it can publish events:
- Amazon Simple Notification Service (Amazon SNS) topic
- Amazon Simple Queue Service (Amazon SQS) queue
- AWS Lambda

SQS FIFO queue are not supported.



	Storage class	Designed for	Availability Zones	Min storage duration
◉	Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-
○	Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-
○	Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days
○	One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days
○	Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days
○	Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days
○	Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days
○	Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)	S3 Intelligent Tiering
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier	S3 Intelligent to S3 One Zone-IA or S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads

Performance limits

A key prefix is a string of characters that can be the complete path in front of the object name. This includes the bucket name. For example, if an object (**123.txt**) is stored as **BucketName/Project/WordFiles/123.txt**, then **the prefix might be BucketName/Project/WordFiles/123.txt**. The prefix can be any length, such as the entire object key name.

Amazon S3 now provides increased performance to support **at least 3,500 requests per second to add data and 5,500 requests per second to retrieve data**, which can save significant processing time for no additional charge. **Each S3 prefix can support these request rates**, making it simple to increase performance significantly.

Applications running on Amazon S3 today will enjoy this performance improvement with no changes, and customers building new applications on S3 do not have to make any application customizations to achieve this performance. Amazon S3's support for parallel requests means you can scale your S3 performance by the factor of your compute cluster without making any customizations to your application. **Performance scales per prefix, so you can use as many prefixes as you need in parallel to achieve the required throughput. There are no limits to the number of prefixes.**

This S3 request rate performance increase removes any previous guidance to randomize object prefixes to achieve faster performance. That means **you can now use logical or sequential naming patterns in S3 object naming without any performance implications**. This improvement is now available in all AWS Regions.

S3 Static Website

Here are the prerequisites for routing traffic to a website that is hosted in an Amazon S3 Bucket:

- An S3 bucket that is configured to host a static website. **The bucket must have the same name as your domain or subdomain.** For example, if you want to use the subdomain portal.tutorialsdojo.com, the name of the bucket must be portal.tutorialsdojo.com.
- A registered domain name. You can use Route 53 as your domain registrar, or you can use a different registrar.
- Route 53 as the DNS service for the domain. If you register your domain name by using Route 53, we automatically configure Route 53 as the DNS service for the domain.

S3 website endpoints names:

http://bucket-name.s3-website.Region.amazonaws.com

http://bucket-name.s3-website-Region.amazonaws.com

S3 Object Lock

With S3 Object Lock, you can store objects using a write-once-read-many (**WORM**) model. Object Lock can help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. You can use Object Lock to help meet regulatory requirements that require WORM storage or to simply add another layer of protection against object changes and deletion.

Before you lock any objects, you have to enable a bucket to use S3 Object Lock. You enable Object Lock when you create a bucket. After you enable Object Lock on a bucket, you can lock objects in that bucket. When you create a bucket with Object Lock enabled, you can't disable Object Lock or suspend versioning for that bucket.

In governance mode, users can't overwrite or delete an object version or alter its lock settings unless they have special permissions. With governance mode, you protect objects against being deleted by most users, but you can still grant some users permission to alter the retention settings or delete the object if necessary. You can also use governance mode to test retention-period settings before creating a compliance-mode retention period.

In compliance mode, a protected object version can't be overwritten or deleted by any user, including the root user in your AWS account. When an object is locked in compliance mode, its retention mode can't be changed, and its retention period can't be shortened. Compliance mode helps ensure that an object version can't be overwritten or deleted for the duration of the retention period.

Legal Hold vs. Retention Period

With Object Lock, you can also place a **legal hold** on an object version. Like a retention period, **a legal hold prevents an object version from being overwritten or deleted**. However, **a legal hold doesn't have an associated retention period and remains in effect until removed**. Legal holds can be freely placed and removed by any user who has the s3:PutObjectLegalHold permission.

Legal holds are independent from retention periods. As long as the bucket that contains the object has Object Lock enabled, you can place and remove legal holds regardless of whether the specified object version has a retention period set. Placing

a legal hold on an object version doesn't affect the retention mode or retention period for that object version.

For example, suppose that you place a legal hold on an object version while the object version is also protected by a retention period. If the retention period expires, the object doesn't lose its WORM protection. Rather, the legal hold continues to protect the object until an authorized user explicitly removes it. Similarly, if you remove a legal hold while an object version has a retention period in effect, the object version remains protected until the retention period expires.

S3 Access Point

Amazon S3 access points simplify data access for any AWS service or customer application that stores data in S3. **Access points are named network endpoints that are attached to buckets that you can use to perform S3 object operations**, such as GetObject and PutObject.

Each access point has distinct permissions and network controls that S3 applies for any request that is made through that access point. Each access point enforces a customized access point policy that works in conjunction with the bucket policy that is attached to the underlying bucket. You can configure any access point to accept requests only from a virtual private cloud (VPC) to restrict Amazon S3 data access to a private network. You can also configure custom block public access settings for each access point.

You can also use Amazon S3 **Multi-Region Access Points** to provide a global endpoint that applications can use to fulfill requests from S3 buckets located in multiple AWS Regions. You can use Multi-Region Access Points to build multi-Region applications with the same simple architecture used in a single Region, and then run those applications anywhere in the world. Instead of sending requests over the congested public internet, Multi-Region Access Points provide built-in network resilience with acceleration of internet-based requests to Amazon S3. Application requests made to a Multi-Region Access Point global endpoint use AWS Global Accelerator to automatically route over the AWS global network to the S3 bucket with the lowest network latency.

Data protection

Data protection refers to protecting data while in-transit (as it travels to and from Amazon S3) and at rest (while it is stored on disks in Amazon S3 data centers). You

can protect data in transit by using SSL or by using client-side encryption. You have the following options for protecting data at rest in Amazon S3:

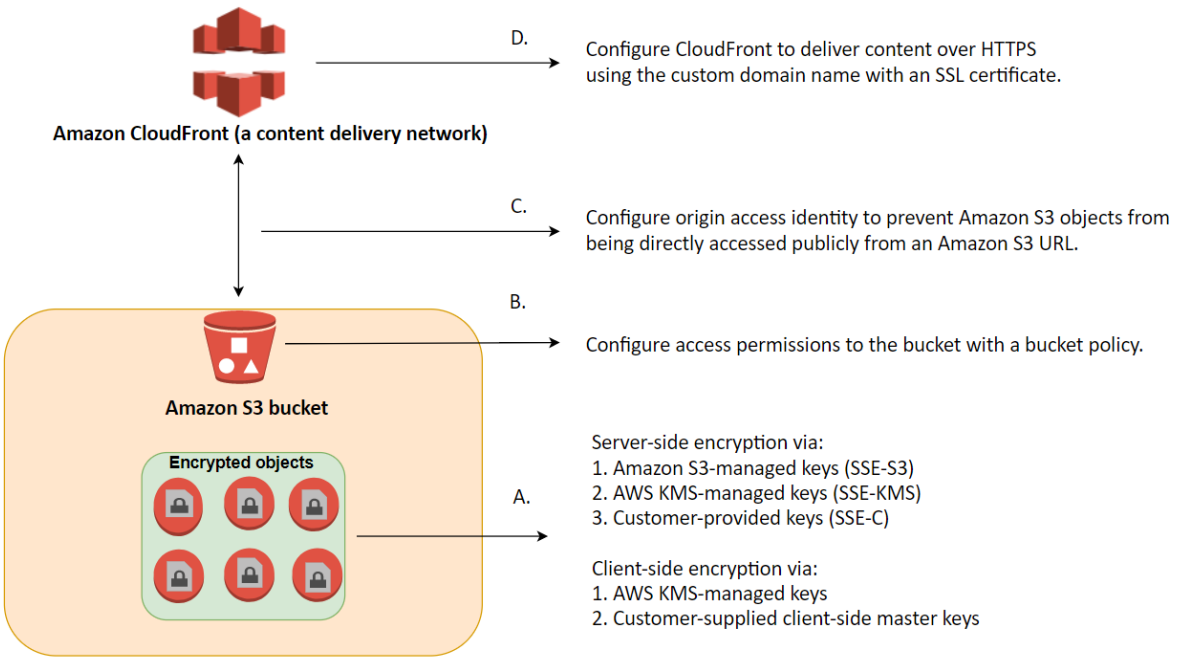
When you use server-side encryption with Amazon S3 managed keys (SSE-S3), each object is encrypted with a unique key. As an additional safeguard, it encrypts the key itself with a root key that it regularly rotates.

Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) (AES-256)

Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

Use Server-Side Encryption with Customer-Provided Keys (SSE-C)

Use Client-Side Encryption – You can 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.



Glacier

Expedited retrievals allow you to quickly access your data when occasional urgent requests for a subset of archives are required. **For all but the largest archives (250 MB+), data accessed using Expedited retrievals are typically made available within 1–5 minutes. Provisioned Capacity ensures that retrieval capacity for Expedited retrievals is available when you need it.**

Provisioned capacity ensures that your retrieval capacity for expedited retrievals is available when you need it. Each unit of capacity provides that at

least three expedited retrievals can be performed every five minutes and provides up to 150 MB/s of retrieval throughput. You should purchase provisioned retrieval capacity if your workload requires highly reliable and predictable access to a subset of your data in minutes. Without provisioned capacity Expedited retrievals are accepted, except for rare situations of unusually high demand. However, if you require access to Expedited retrievals under all circumstances, you must purchase provisioned retrieval capacity.

You can't directly copy data from AWS Snowball Edge devices into Amazon S3 Glacier.

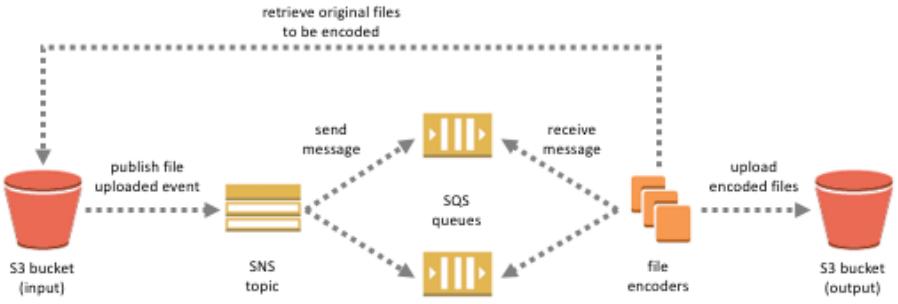
Notification feature

The Amazon S3 notification feature enables you to receive notifications when certain events happen in your bucket. To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications. You store this configuration in the notification subresource that is associated with a bucket.

Amazon S3 supports the following destinations where it can publish events:

- Amazon Simple Notification Service (Amazon SNS) topic
- Amazon Simple Queue Service (Amazon SQS) queue
- AWS Lambda

In Amazon SNS, **the fanout scenario is when a message published to an SNS topic is replicated and pushed to multiple endpoints, such as Amazon SQS queues, HTTP(S) endpoints, and Lambda functions**. This allows for parallel asynchronous processing.



Amazon FSx for Lustre

Amazon FSx for Lustre makes it easy and cost-effective to launch and run the world's most popular **high-performance file system**. **It is used for workloads such as machine learning, high-performance computing (HPC), video processing, and financial modeling**. The open-source Lustre file system is designed for applications that require fast storage – where you want your storage to keep up with your compute. FSx for Lustre integrates with Amazon S3, making it easy to process data sets with the Lustre file system. When linked to an S3 bucket, an FSx for Lustre file system transparently presents S3 objects as files and allows you to write changed data back to S3.

FSx for Lustre provides the ability to both process the '**hot data**' in a parallel and distributed fashion as well as easily store the '**cold data**' on Amazon S3.

Amazon FSx for Windows File Server

Provides fully managed, highly reliable file storage that is accessible over the industry-standard Service Message Block (SMB) protocol. It is built on Windows Server, delivering a wide range of administrative features such as user quotas, end-user file restore, and Microsoft Active Directory (AD) integration. FSx for Windows does not allow you to present S3 objects as files and does not allow you to write changed data back to S3. Therefore you cannot reference the "cold data" with quick access for reads and updates at low cost.

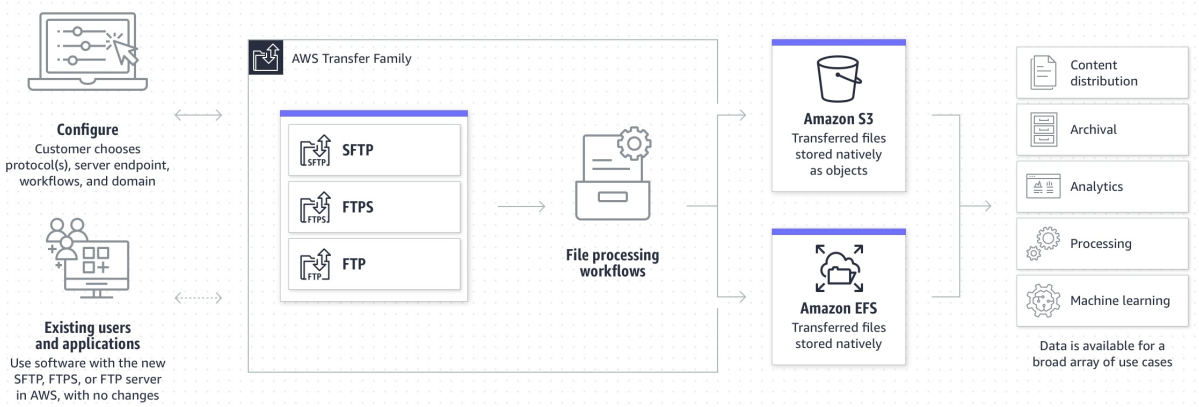
Amazon FSx File Gateway

For user or team file shares, and file-based application migrations, **Amazon FSx File Gateway provides low-latency, on-premises access to fully managed file shares in Amazon FSx for Windows File Server**. For applications deployed on AWS, you may access your file shares directly from Amazon FSx in AWS. **Supports native Windows workloads.**

AWS Transfer (for SFTP)

AWS Transfer for SFTP enables you to easily move your file transfer workloads that use the Secure Shell File Transfer Protocol (SFTP) to AWS without needing to modify your applications or manage any SFTP servers.

To get started with AWS Transfer for SFTP (AWS SFTP) you create an SFTP server and map your domain to the server endpoint, select authentication for your SFTP clients using service-managed identities, or integrate your own identity provider, and select your Amazon S3 buckets to store the transferred data. Your existing users can continue to operate with their existing SFTP clients or applications. Data uploaded or downloaded using SFTP is available in your Amazon S3 bucket, and can be used for archiving or processing in AWS.



AWS Storage Gateway

AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. 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.

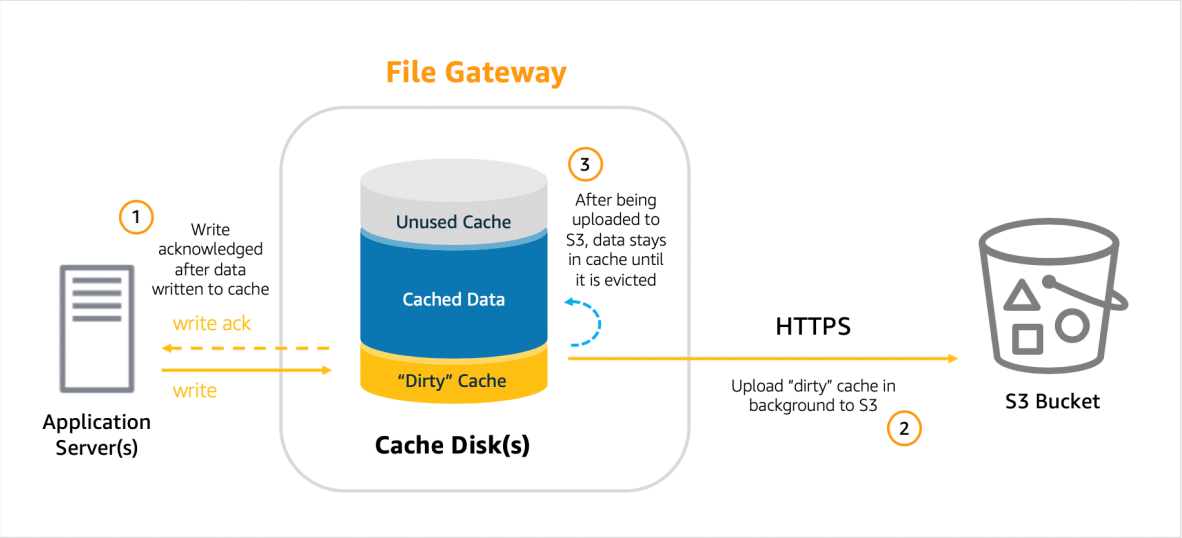
Storage Gateway is mainly used in providing low-latency access to data by caching frequently accessed data on-premises while storing archive data securely and durably in Amazon cloud storage services. Storage Gateway optimizes data transfer to AWS by sending only changed data and compressing data.

Tape Gateway enables you to replace physical tapes on-premises with virtual tapes in AWS without changing existing backup workflows. Tape Gateway supports all leading backup applications and caches virtual tapes on-premises for low-latency data access. Tape Gateway encrypts data between the gateway and AWS for secure data transfer and compresses data and transitions virtual tapes between Amazon S3 and Amazon S3 Glacier, or Amazon S3 Glacier Deep Archive, to minimize storage costs.

File gateway

AWS File Gateway acts as a bridge between on-premises environments and Amazon S3, enabling you to store and retrieve files as objects in S3 using standard file protocols such as NFS, SMB, and iSCSI. Begin by deploying the gateway on a virtual machine or hardware appliance and connect it to your AWS account. Ensure proper IAM roles and security groups are in place for secure access. Configure the gateway to cache frequently accessed data locally for low-latency access or store all data directly on S3 based on your use case. Use the AWS Management Console for monitoring and management tasks, set up S3 lifecycle policies for cost-effective data management, and implement regular backups and snapshots for data durability and recovery. Always encrypt sensitive data both at rest and in transit, and review AWS CloudTrail logs to audit access and changes to your file gateway.

Doesn’t support native Windows workloads.



Volume Gateway

The Volume Gateway is a cloud-based iSCSI block storage volume for your on-premises applications. The Volume Gateway provides either a local cache or full volumes on-premises while also storing full copies of your volumes in the AWS cloud.

There are two options for Volume Gateway:

Cached Volumes - you store volume data in AWS, with a small portion of recently accessed data in the cache on-premises.

Stored Volumes - you store the entire set of volume data on-premises and store periodic point-in-time backups (snapshots) in AWS.

AWS Data Sync

AWS DataSync makes it simple and fast to move large amounts of data online between on-premises storage and Amazon S3, Amazon Elastic File System (Amazon EFS), or Amazon FSx for Windows File Server.

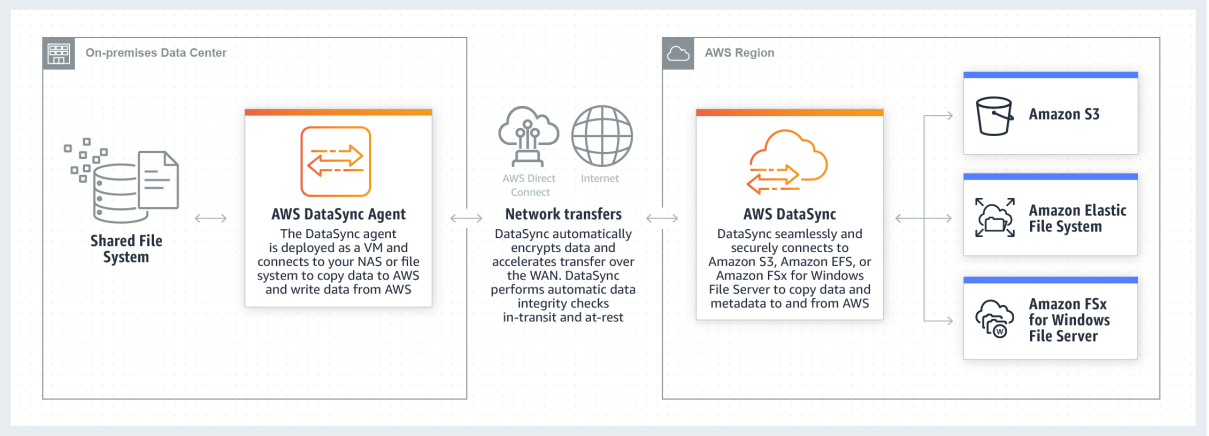
With AWS DataSync, you can transfer data from on-premises directly to Amazon S3 Glacier Deep Archive. You don't have to configure the S3 lifecycle policy and wait for 30 days to move the data to Glacier Deep Archive.

Manual tasks related to data transfers can slow down migrations and burden IT operations. DataSync eliminates or automatically handles many of these tasks, including scripting copy jobs, scheduling, and monitoring transfers, validating data, and optimizing network utilization. The DataSync software agent connects to your

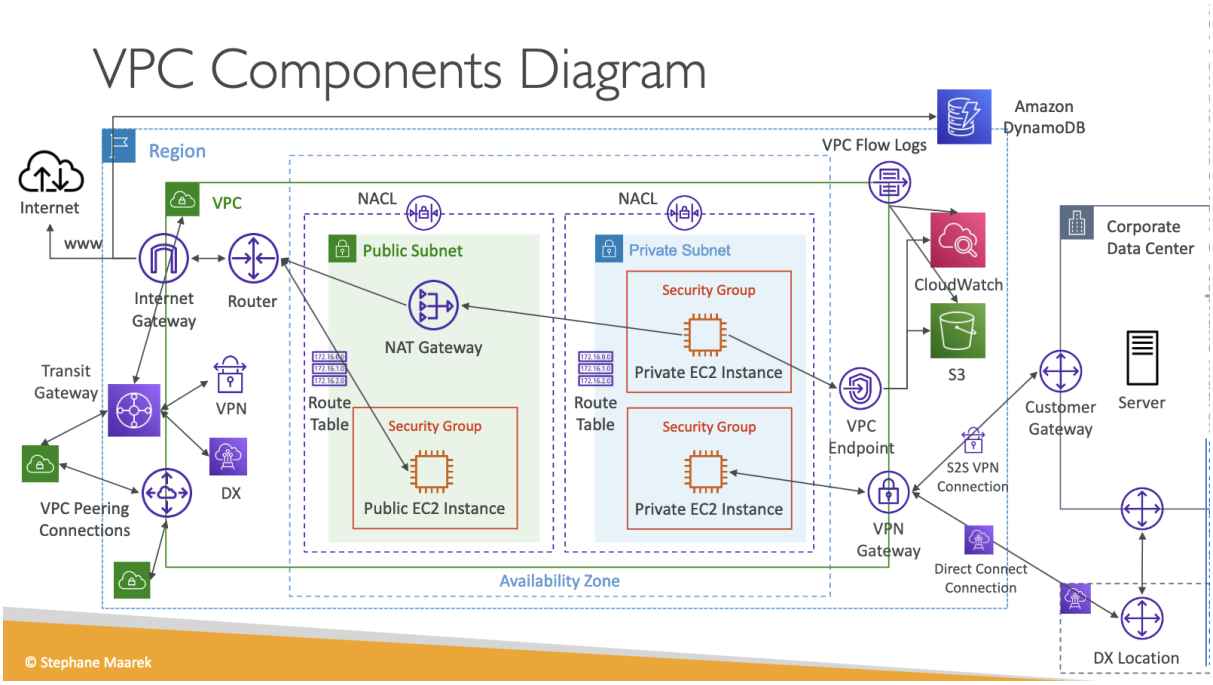
Network File System (NFS), Server Message Block (SMB) storage, and your self-managed object storage, so you don’t have to modify your applications.

DataSync can transfer hundreds of terabytes and millions of files at speeds up to 10 times faster than open-source tools, over the Internet or AWS Direct Connect links. You can use DataSync to migrate active data sets or archives to AWS, transfer data to the cloud for timely analysis and processing, or replicate data to AWS for business continuity. Getting started with DataSync is easy: deploy the DataSync agent, connect it to your file system, select your AWS storage resources, and start moving data between them. You pay only for the data you move.

To migrate from NFS on-premises with DirectConnect: configure an AWS DataSync agent on the on-premises server that has access to the NFS file system. Transfer data over the AWS Direct Connect connection to an AWS PrivateLink interface VPC endpoint for Amazon EFS by using a private VIF. Set up an AWS DataSync scheduled task to send the files to the Amazon EFS file system every 24 hours.



Amazon VPC



A VPC **spans all the Availability Zones in the region**. After creating a VPC, you can add one or more subnets in each Availability Zone. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block. **Each subnet must reside entirely within one Availability Zone and cannot span zones**. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

VPC Endpoint

A VPC endpoint enables you to **privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection**. Instances in your VPC do not require public IP addresses to communicate with resources in the service. **AWS PrivateLink simplifies the security of data shared with cloud-based applications by eliminating the exposure of data to the public Internet**. **When you use VPC endpoint, the traffic between your VPC and the other AWS service does not leave the Amazon network,** therefore this option cannot be used to send and receive data between the remote branch offices of the company.

There are two types of VPC endpoints: *interface endpoints* and *gateway endpoints*. You have to create the type of VPC endpoint required by the supported service.

Interface Endpoint

An interface endpoint **is an elastic network interface with a private IP address** that serves as an entry point for traffic destined to a supported service.

Gateway Endpoint

A Gateway endpoint is a type of VPC endpoint that provides reliable connectivity to Amazon S3 and DynamoDB without requiring an internet gateway or a NAT device for your VPC. Instances in your VPC do not require public IP addresses to communicate with resources in the service.

Connect to Amazon S3 and DynamoDB without passing through the internet.

When you create a Gateway endpoint, you can attach an endpoint policy that controls access to the service to which you are connecting. You can modify the endpoint policy attached to your endpoint and add or remove the route tables used by the endpoint. An endpoint policy does not override or replace IAM user policies or service-specific policies (such as S3 bucket policies). It is a separate policy for controlling access from the endpoint to the specified service.

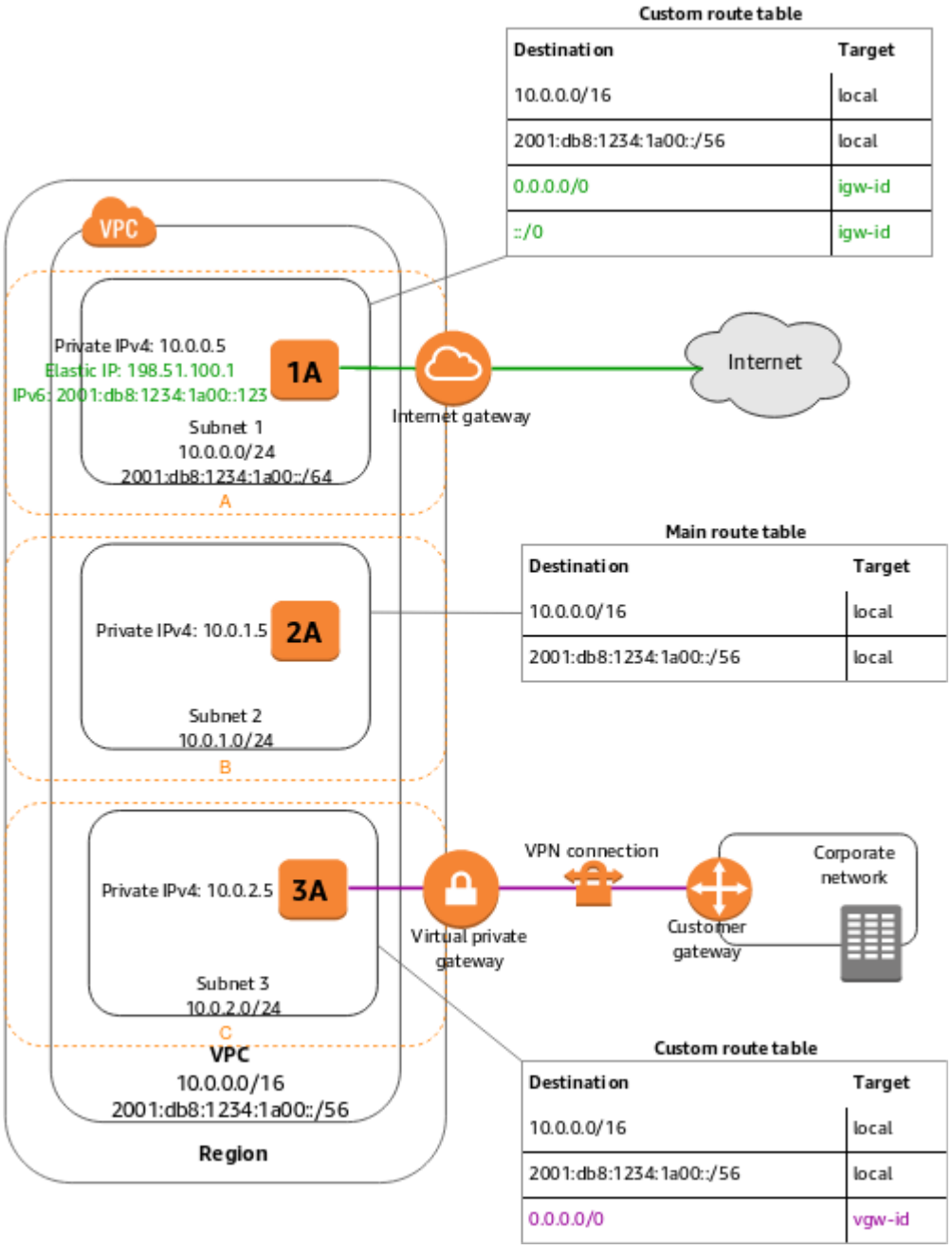
Gateway endpoints for Amazon S3	Interface endpoints for Amazon S3
In both cases, your network traffic remains on the AWS network.	
Use Amazon S3 public IP addresses	Use private IP addresses from your VPC to access Amazon S3
Does not allow access from on premises	Allow access from on premises
Does not allow access from another AWS Region	Allow access from a VPC in another AWS Region using VPC peering or AWS Transit Gateway
Not billed	Billed

VPC Sharing

VPC sharing (part of Resource Access Manager) allows multiple AWS accounts to create their application resources into shared and centrally-managed Amazon Virtual Private Clouds (VPCs).

To set this up, the account that owns the VPC (owner) shares one or more subnets with other accounts (participants) that belong to the same organization from AWS Organizations. After a subnet is shared, the participants can view, create, modify, and delete their application resources in the subnets shared with them. Participants cannot view, modify, or delete resources that belong to other participants or the VPC owner.

You can share Amazon VPCs to leverage the implicit routing within a VPC for applications that require a high degree of interconnectivity and are within the same trust boundaries. This reduces the number of VPCs that you create and manage while using separate accounts for billing and access control.

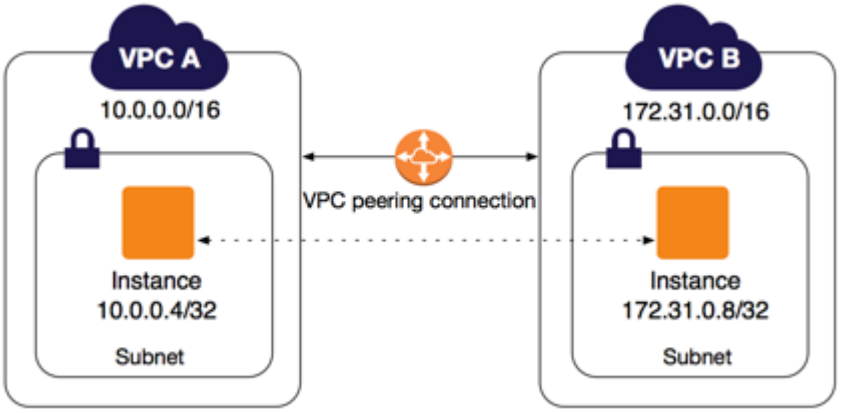


VPC Peering

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).

To make two EC2 instance communicate through VPC peering you need to **re-configure the route table's target and destination of the instances' subnet.**

Inter-Region VPC Peering provides a simple and cost-effective way to share resources between regions or replicate data for geographic redundancy. Built on the same horizontally scaled, redundant, and highly available technology that powers VPC today, Inter-Region VPC Peering encrypts inter-region traffic with no single point of failure or bandwidth bottleneck. Traffic using Inter-Region VPC Peering always stays on the global AWS backbone and never traverses the public internet, thereby reducing threat vectors, such as common exploits and DDoS attacks.



AWS Network Firewall

AWS Network Firewall is a **stateful, managed network firewall and intrusion detection and prevention service for your virtual private cloud (VPC)** that you created in Amazon Virtual Private Cloud (Amazon VPC). With Network Firewall, you can **filter traffic** at the perimeter of your VPC. This includes filtering traffic going to and coming from an internet gateway, NAT gateway, or over VPN or AWS Direct Connect. Network Firewall uses the open source intrusion prevention system (IPS), Suricata, for stateful inspection. Network Firewall supports Suricata compatible rules.

AWS Network Firewall's stateful firewall can incorporate context from traffic flows, like tracking connections and protocol identification, to enforce policies such as preventing your VPCs from accessing domains using an unauthorized protocol. **AWS Network Firewall's intrusion prevention system (IPS) provides active traffic flow inspection** so you can identify and block vulnerability exploits using signature-based detection. AWS Network Firewall also offers web filtering that can stop traffic to known bad URLs and monitor fully qualified domain names.

Traffic Mirroring

Traffic Mirroring is an Amazon VPC feature that you can use to **copy network traffic from an elastic network interface of type** `interface`. You can then send the traffic to out-of-band security and monitoring appliances for:
- Content inspection
- Threat monitoring
- Troubleshooting

The security and monitoring appliances can be deployed as individual instances, or as a fleet of instances behind either a Network Load Balancer or a Gateway Load Balancer with a UDP listener. Traffic Mirroring supports filters and packet truncation, so that you can extract only the traffic of interest, using the monitoring tools of your choice.

Subnet

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.

Below are the important points you have to remember about subnets:

- **Each subnet maps to a single Availability Zone.**
- **Every subnet that you create is automatically associated with the main route table for the VPC.**
- **If a subnet's traffic is routed to an Internet gateway, the subnet is known as a public subnet.**

IPV4 are required. If you receive an error saying that there is no IP address available on the subnet then set up a new IPv4 subnet with a larger CIDR range.

AWS Transit Gateway

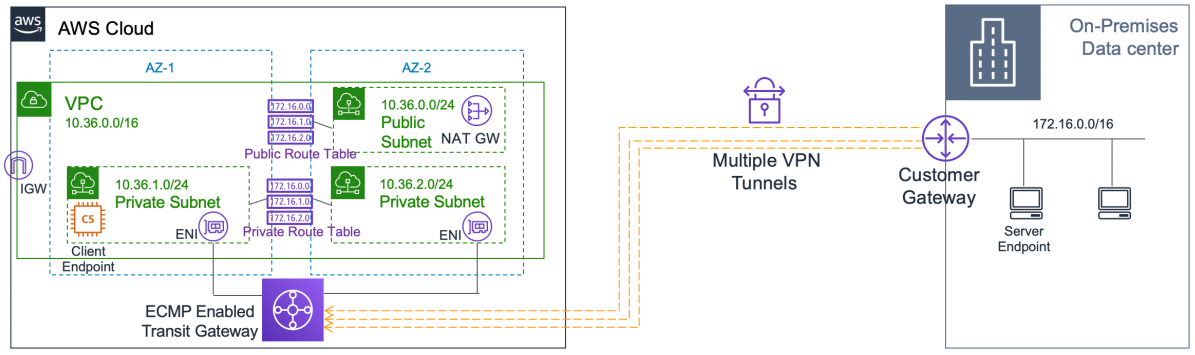
AWS Transit Gateway provides a hub and spoke design for connecting VPCs and on-premises networks. You can attach all your hybrid connectivity (VPN and Direct Connect connections) to a single Transit Gateway consolidating and controlling your organization's entire AWS routing configuration in one place. It also controls how traffic is routed among all the connected spoke networks using route tables. This hub and spoke model simplifies management and reduces

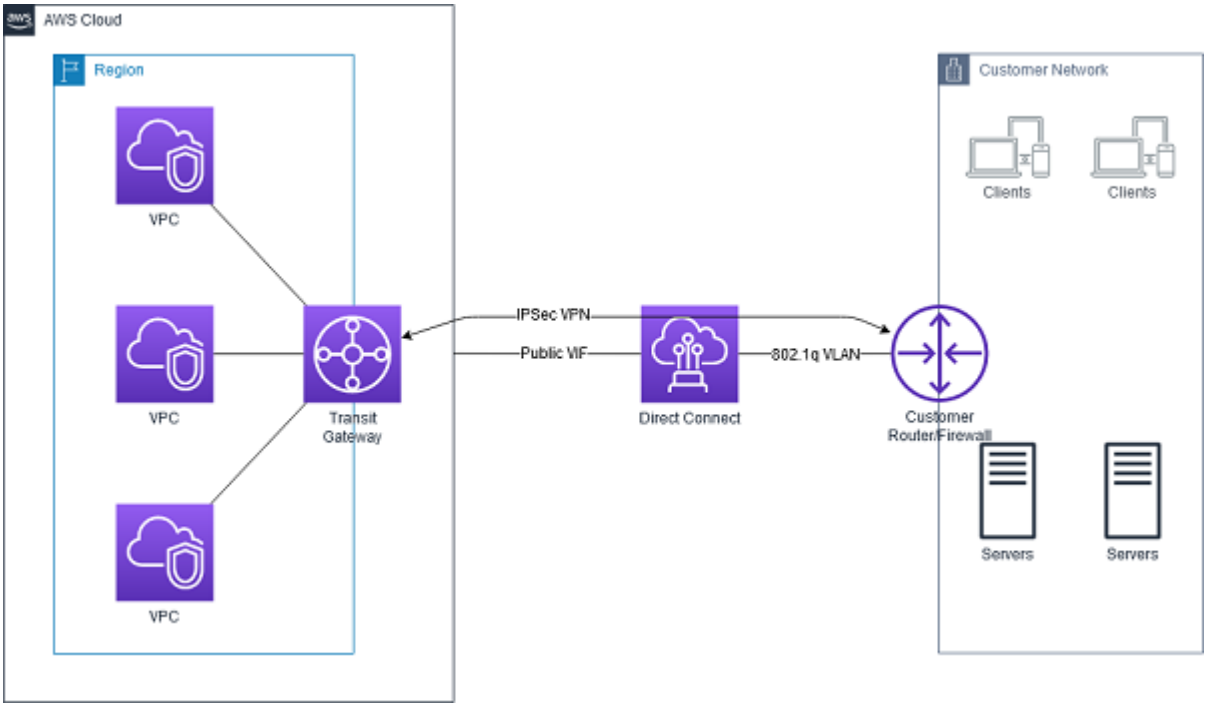
operational costs because VPCs only connect to the Transit Gateway to gain access to the connected networks.

By attaching a transit gateway to a Direct Connect gateway using a transit virtual interface, you can manage a single connection for multiple VPCs or VPNs that are in the same AWS Region. You can also advertise prefixes from on-premises to AWS and from AWS to on-premises.

The AWS Transit Gateway and AWS Direct Connect solution simplify the management of connections between an Amazon VPC and your networks over a private connection. It can also minimize network costs, improve bandwidth throughput, and provide a more reliable network experience than Internet-based connections.

AWS Transit Gateway also enables you to scale the IPsec VPN throughput with equal-cost multi-path (ECMP) routing support over multiple VPN tunnels. A single VPN tunnel still has a maximum throughput of 1.25 Gbps. If you establish multiple VPN tunnels to an ECMP-enabled transit gateway, it can scale beyond the default limit of 1.25 Gbps.





NAT Gateway

A NAT Gateway is a highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet. NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.

You must create a NAT gateway on a public subnet to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the Internet from initiating a connection with those instances.

If you have resources in multiple Availability Zones and they share one NAT gateway, and if the NAT gateway's Availability Zone is down, resources in the other Availability Zones lose Internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

Gateway Load Balancer

Gateway Load Balancer is an AWS service tailored for **distributing network traffic across multiple virtual appliances**, such as firewalls and intrusion detection systems. It **operates at the network layer, handling both TCP and UDP traffic, and can be deployed within a VPC to span multiple Availability Zones**. The service simplifies the integration of third-party virtual appliances by providing a

seamless way to route traffic for inspection without altering existing network architectures.

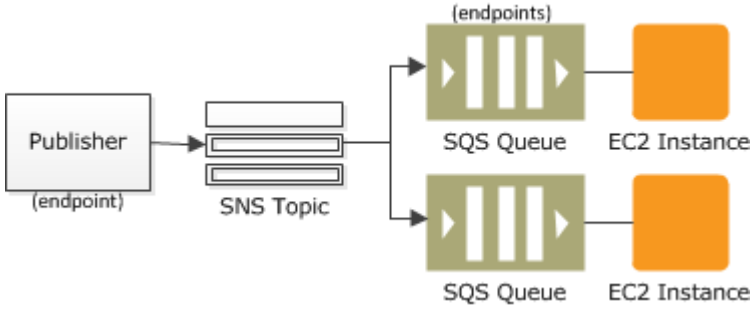
Through the use of Gateway Load Balancer endpoints, traffic is directed to the appropriate appliances for processing. These endpoints can be equipped with health checks to ensure traffic is only sent to operational appliances. The service ensures high availability and fault tolerance by rerouting traffic away from failed instances to healthy ones.

Gateway Load Balancer offers detailed monitoring and logging for effective tracking and troubleshooting. It also supports scalability, allowing for the dynamic addition or removal of appliances based on demand. In terms of billing, charges are based on the volume of data processed and the number of endpoints, making it a cost-efficient choice for enhancing network security and resilience.

Simple Notification Service (SNS)

Amazon Simple Notification Service (Amazon SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.

Example when two SQS queues need to consume the same data (fanout):



Simple Queue Service (SQS)

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.

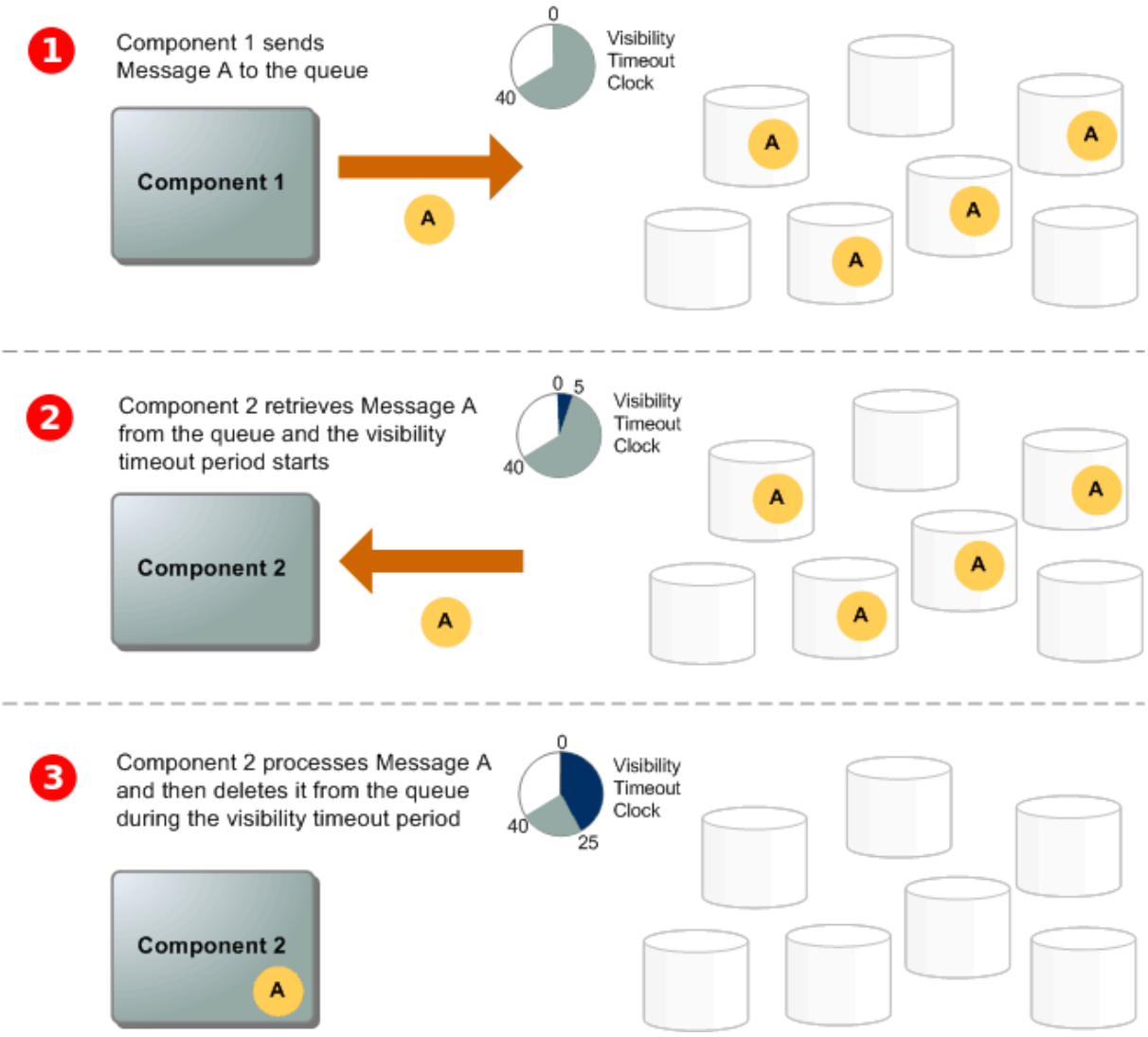
You **cannot set a priority to individual items** in the SQS queue.

Always remember that the **messages in the SQS queue will continue to exist even after the EC2 instance has processed it, until you delete that message**. You have to ensure that you delete the message after processing to prevent the message from being received and processed again once the visibility timeout expires.

Long polling helps reduce the cost of using SQS by eliminating the number of empty responses (when there are no messages available for a ReceiveMessage request) and false empty responses.

In Amazon SQS, you can configure the message **retention period to a value from 1 minute to 14 days**. The default is 4 days. Once the message retention limit is reached, your messages are automatically deleted.

Only Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed.

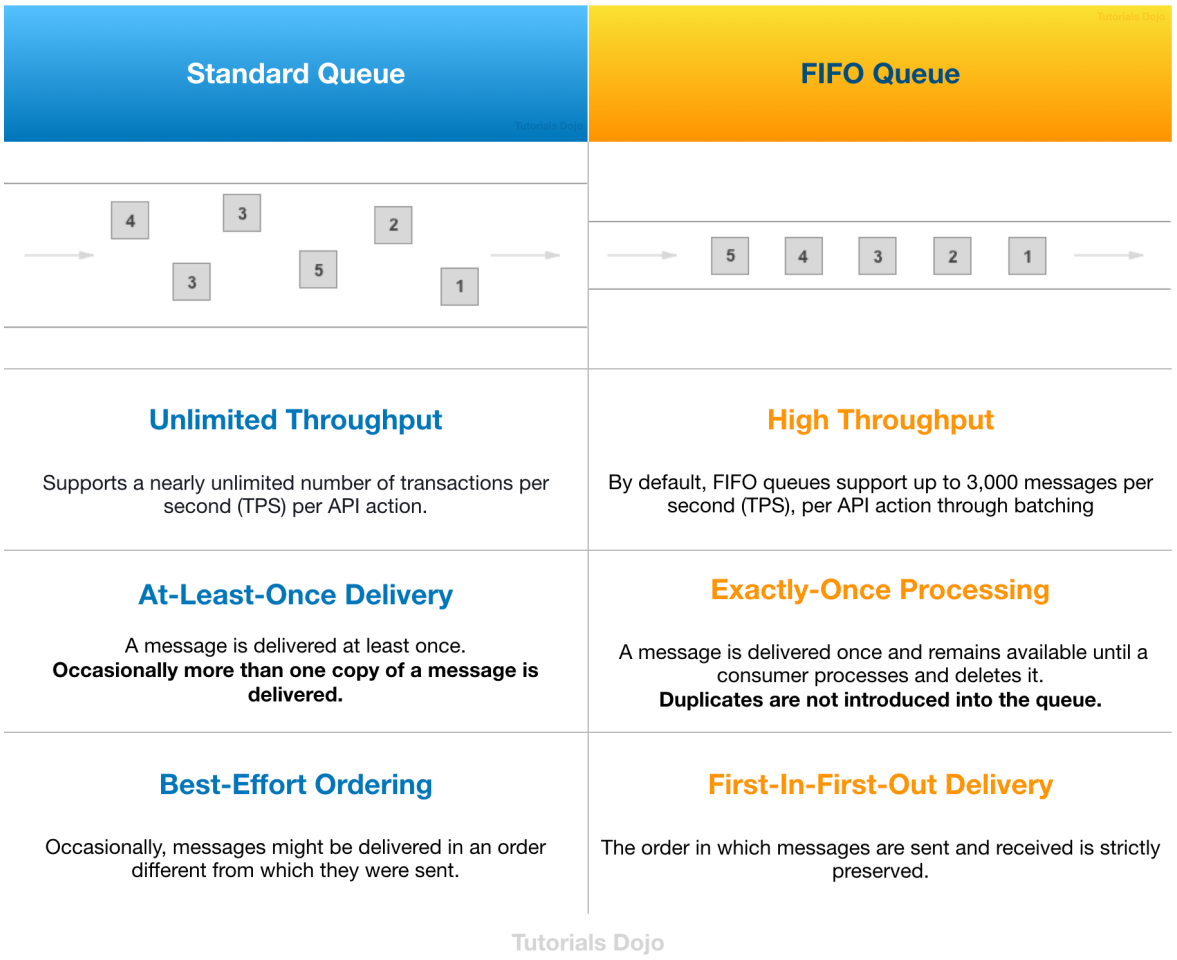


FIFO Queues

Amazon 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.

By default, FIFO queues support **up to 3,000 messages per second with batching**, or up to **300 messages per second (300 send, receive, or delete operations per second) without batching**. Therefore, using batching you can meet a throughput requirement of up to 3,000 messages per second.

You can't convert an existing standard queue into a FIFO queue. To make the move, you must either create a new FIFO queue for your application or delete your existing standard queue and recreate it as a FIFO queue.



Temporary Queues

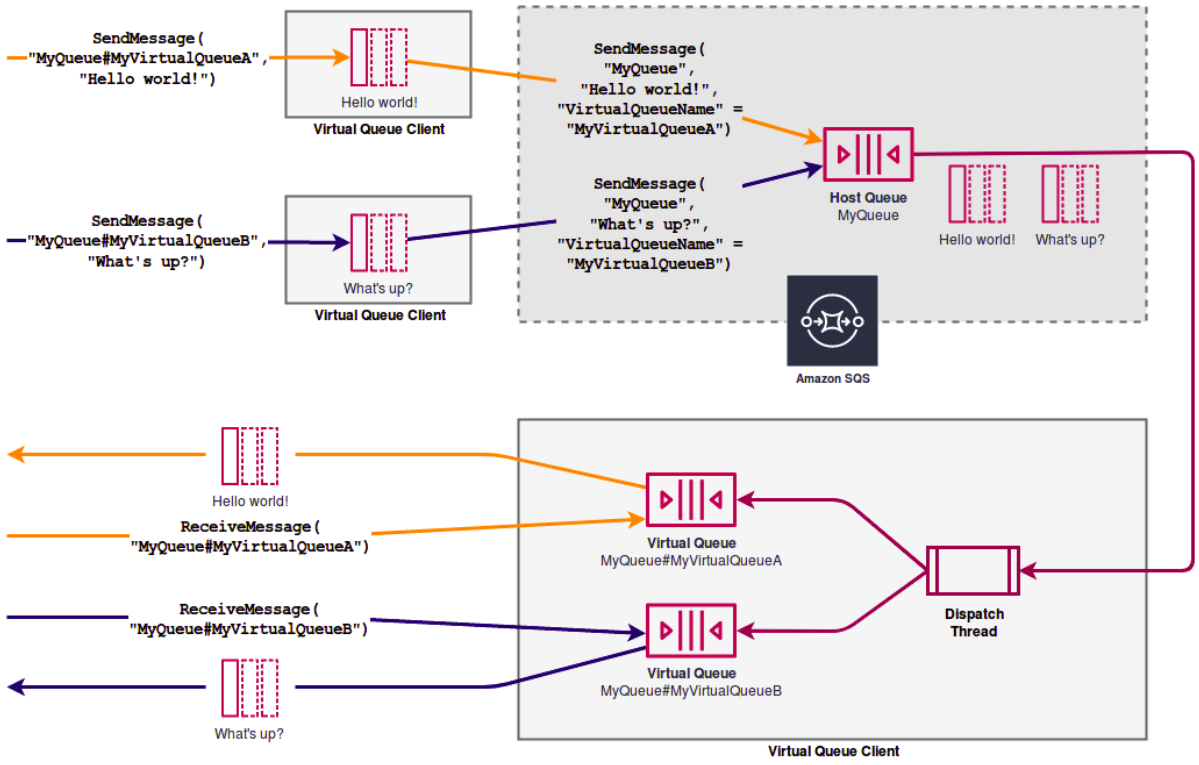
Temporary queues help you save development time and deployment costs when using common message patterns such as request-response. You can use the Temporary Queue Client to create high-throughput, cost-effective, application-managed temporary queues.

The client maps multiple temporary queues—application-managed queues created on demand for a particular process—onto a single Amazon SQS queue automatically. This allows your application to make fewer API calls and have a higher throughput when the traffic to each temporary queue is low. When a temporary queue is no longer in use, the client cleans up the temporary queue automatically, even if some processes that use the client aren't shut down cleanly.

The following are the benefits of temporary queues:

1. They serve as lightweight communication channels for specific threads or processes.
2. They can be created and deleted without incurring additional costs.

3. They are API-compatible with static (normal) Amazon SQS queues. This means that existing code that sends and receives messages can send messages to and receive messages from virtual queues.



AWS CloudHSM

AWS CloudHSM is a cryptographic service for creating and maintaining **hardware security modules (HSMs) in your AWS environment**. HSMs are computing devices that process cryptographic operations and provide secure storage for cryptographic keys. You can use AWS CloudHSM to offload SSL/TLS processing for web servers, protect private keys linked to an issuing certificate authority (CA), or enable Transparent Data Encryption (TDE) for Oracle databases.

You should consider using AWS CloudHSM over AWS KMS if you require your keys stored in dedicated, thirt-party validated hardware security modules under your exclusive control.

AWS Key Management Service (KMS)

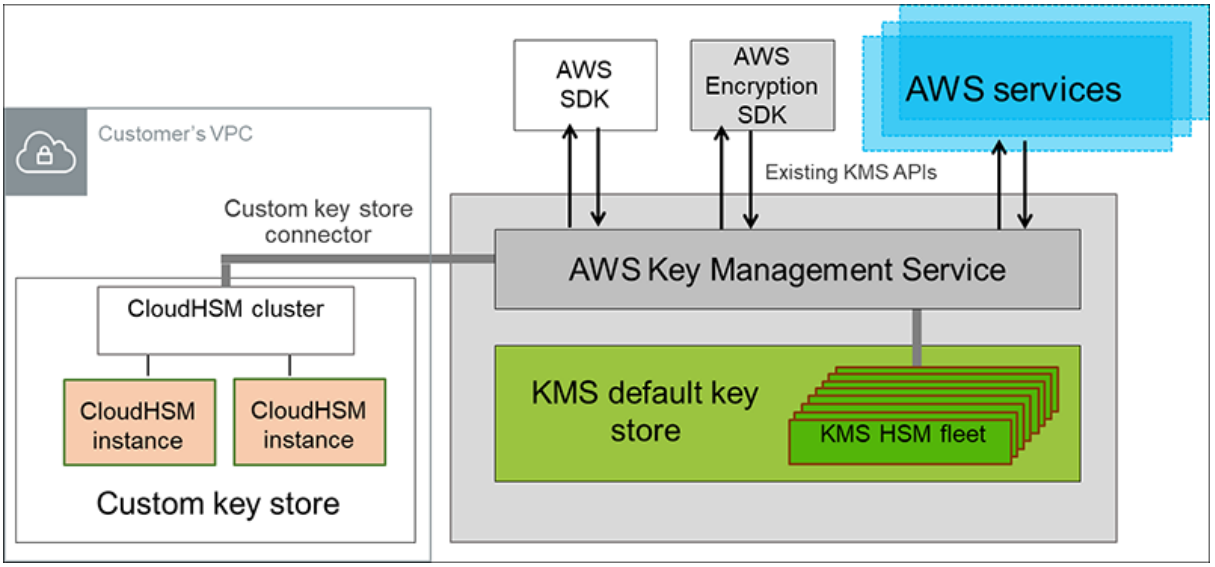
The AWS Key Management Service (KMS) custom key store feature combines the controls provided by **AWS CloudHSM** with the integration and ease of use of AWS KMS. You can configure your own CloudHSM cluster and authorize AWS KMS to use

it as a dedicated key store for your keys rather than the default AWS KMS key store. When you create keys in AWS KMS you can choose to generate the key material in your CloudHSM cluster. CMKs that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all AWS KMS operations that use those keys are only performed in your HSMs.

AWS KMS can help you integrate with other AWS services to encrypt the data that you store in these services and control access to the keys that decrypt it. To immediately remove the key material from AWS KMS, you can use a custom key store. Take note that each custom key store is associated with an AWS CloudHSM cluster in your AWS account. Therefore, when you create an AWS KMS CMK in a custom key store, AWS KMS generates and stores the non-extractable key material for the CMK in an AWS CloudHSM cluster that you own and manage. This is also suitable if you want to be able to audit the usage of all your keys independently of AWS KMS or AWS CloudTrail.

Since you control your AWS CloudHSM cluster, you have the option to manage the lifecycle of your CMKs independently of AWS KMS. There are four reasons why you might find a custom key store useful:

1. You might have keys that are explicitly required to be protected in a single-tenant HSM or in an HSM over which you have direct control.
2. You might have keys that are required to be stored in an HSM that has been validated to FIPS 140-2 level 3 overall (the HSMs used in the standard AWS KMS key store are either validated or in the process of being validated to level 2 with level 3 in multiple categories).
3. You might need the ability to immediately remove key material from AWS KMS and to prove you have done so by independent means.
4. You might have a requirement to be able to audit all use of your keys independently of AWS KMS or AWS CloudTrail.



Key Policy

A key policy is a resource policy for an AWS KMS key. Key policies are the primary way to control access to KMS keys. Every KMS key must have exactly one key policy. The statements in the key policy determine who has permission to use the KMS key and how they can use it. You can also use IAM policies and grants to control access to the KMS key, but every KMS key must have a key policy.

Unless the key policy explicitly allows it, you cannot use IAM policies to allow access to a KMS key. Without permission from the key policy, IAM policies that allow permissions have no effect. (You can use an IAM policy to deny permission to a KMS key without permission from a key policy.) The default key policy enables IAM policies.

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.

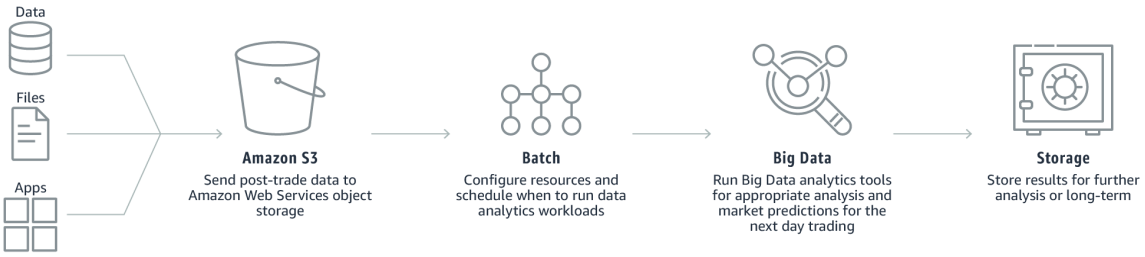
You cannot convert an existing single-Region key to a multi-Region key.

AWS Batch

AWS Batch is a powerful tool for developers, scientists, and engineers who need to **run a large number of batch and ML computing jobs**. By optimizing compute

resources, AWS Batch enables you to focus on analyzing outcomes and resolving issues, rather than worrying about the technical details of running jobs.

With AWS Batch, you can define and submit multiple simulation jobs to be executed concurrently. AWS Batch will take care of distributing the workload across multiple EC2 instances, scaling up or down based on the demand, and managing the execution environment. It provides an easy-to-use interface and automation for managing the simulations, allowing you to focus on the sofware itself rather than the underlying infrastructure.



Elastic Kubernetes Service

The Kubernetes Horizontal Pod Autoscaler automatically scales the number of Pods in a deployment, replication controller, or replica set based on that resource's CPU utilization. This can help your applications scale out to meet increased demand or scale in when resources are not needed, thus freeing up your nodes for other applications. When you set a target CPU utilization percentage, the Horizontal Pod Autoscaler scales your application in or out to try to meet that target.

Amazon EKS supports two autoscaling products:

- Karpenter
- Cluster Autoscaler

The **Kubernetes Cluster Autoscaler automatically adjusts the number of nodes in your cluster when pods fail or are rescheduled onto other nodes**. The Cluster Autoscaler uses Auto Scaling groups.

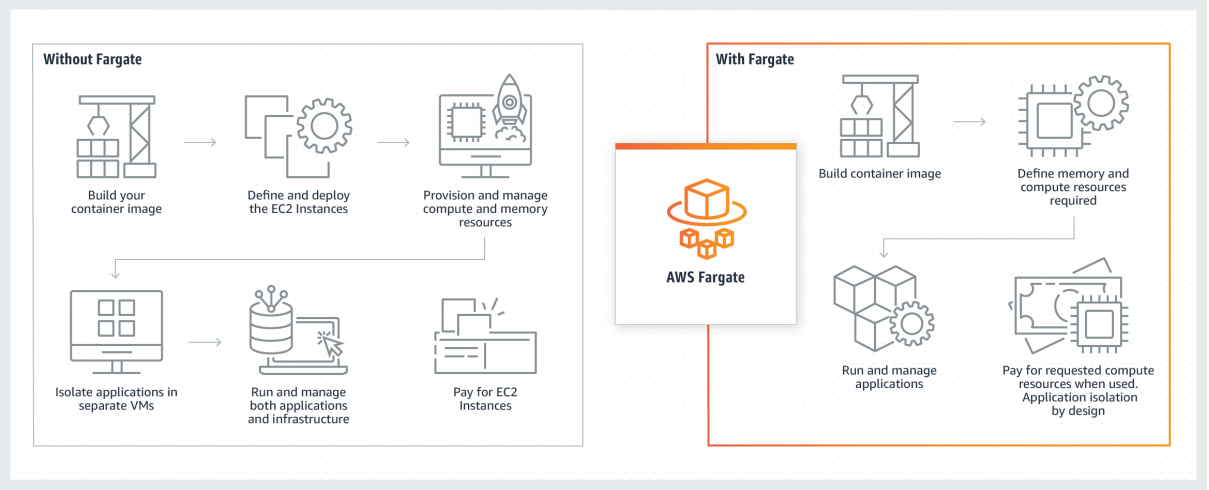
Karpenter is a flexible, high-performance Kubernetes cluster autoscaler that launches appropriately sized compute resources, like Amazon EC2 instances, in response to changing application load. It integrates with AWS to provision compute resources that precisely match workload requirements.

AWS Fargate

AWS Fargate is a serverless compute engine for containers that work 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, 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.

By default, Fargate tasks are given a minimum of **20 GiB of free ephemeral storage.**



AWS CloudFormation

AWS CloudFormation lets you model, provision, and manage AWS and third-party resources by treating **infrastructure as code**.

Use AWS CloudFormation StackSets to deploy the same template across AWS accounts and regions.

CreationPolicy

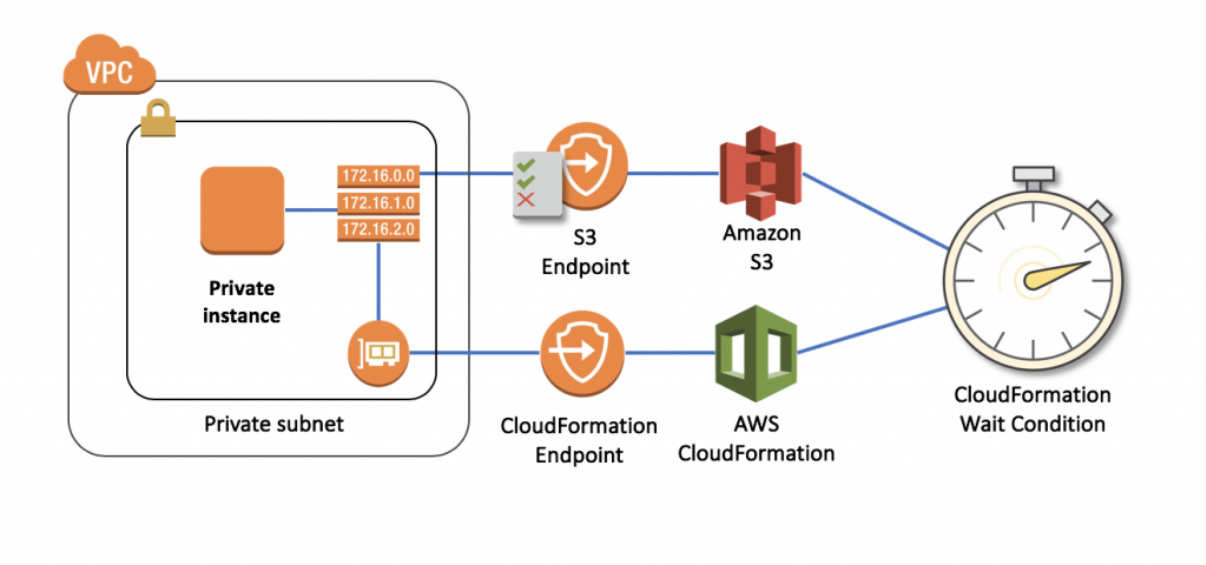
You can associate the `CreationPolicy` attribute with a resource to **prevent its status from reaching create complete until AWS CloudFormation receives a specified number of success signals or the timeout period is exceeded**. To signal a

resource, you can use the cfn-signal helper script or SignalResource API. AWS CloudFormation publishes valid signals to the stack events so that you track the number of signals sent.

The creation policy is invoked only when AWS CloudFormation creates the associated resource. Currently, the only AWS CloudFormation resources that support creation policies are `AWS::AutoScaling::AutoScalingGroup`, `AWS::EC2::Instance`, and `AWS::CloudFormation::WaitCondition`.

The creation policy is invoked only when AWS CloudFormation creates the associated resource. Currently, the only AWS CloudFormation resources that support creation policies are `AWS::AutoScaling::AutoScalingGroup`, `AWS::EC2::Instance`, and `AWS::CloudFormation::WaitCondition`.

Use the `CreationPolicy` attribute when you want to wait on resource configuration actions before stack creation proceeds. For example, if you install and configure software applications on an EC2 instance, you might want those applications to be running before proceeding. In such cases, you can add a `CreationPolicy` attribute to the instance and then send a success signal to the instance after the applications are installed and configured.



Security Group VS Network Access Control List

Security Group	Network Access Control List
Acts as a firewall for associated Amazon EC2 instances	Acts as a firewall for associated subnets
Controls both inbound and outbound traffic at the instance level	Controls both inbound and outbound traffic at the subnet level
You can secure your VPC instances using only security groups	Network ACLs are an additional layer of defense.
Supports allow rules only	Supports allow rules and deny rules
Stateful (Return traffic is automatically allowed, regardless of any rules)	Stateless (Return traffic must be explicitly allowed by rules)
Evaluates all rules before deciding whether to allow traffic	Evaluates rules in number order when deciding whether to allow traffic, starting with the lowest numbered rule.
Applies only to the instance that is associated to it	Applies to all instances in the subnet it is associated with
Has separate rules for inbound and outbound traffic	Has separate rules for inbound and outbound traffic
A newly created security group denies all inbound traffic by default	A newly created nACL denies all inbound traffic by default
A newly created security group has an outbound rule that allows all outbound traffic by default	A newly created nACL denies all outbound traffic default
Instances associated with a security group can't talk to each other unless you add rules allowing it	Each subnet in your VPC must be associated with a network ACL. If none is associated, the default nACL is selected.
Security groups are associated with network interfaces	You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.

AWS Systems Manager

AWS Systems Manager is a collection of capabilities to help you manage your applications and infrastructure running in the AWS Cloud. Systems Manager simplifies application and resource management, shortens the time to detect and resolve operational problems, and helps you manage your AWS resources securely at scale.

Parameter Store

Parameter Store provides secure, hierarchical storage for configuration data and secrets management. **You can store data such as passwords, database strings, Amazon Elastic Compute Cloud (Amazon EC2) instance IDs and Amazon Machine Image (AMI) IDs, and license codes as parameter values**. You can store values as plain text or encrypted data. You can then reference values by using the unique name you specified when you created the parameter. Parameter Store is also integrated with Secrets Manager. You can retrieve Secrets Manager secrets when using other AWS services that already support references to Parameter Store parameters.

AWS Secrets Manager

AWS Secrets Manager 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**.

Use AWS Secrets Manager with a new AWS KMS key to securely manage and store sensitive data within the EKS cluster's etcd key-value store.

Run Command

AWS Systems Manager Run Command lets you remotely and securely manage the configuration of your managed instances. A managed instance is any Amazon EC2 instance or on-premises machine in your hybrid environment that has been configured for Systems Manager. Run Command enables you to automate common administrative tasks and perform ad hoc configuration changes at scale. You can use Run Command from the AWS console, the AWS Command Line Interface, AWS Tools for Windows PowerShell, or the AWS SDKs. Run Command is offered at no additional cost.

Patch Manager

The primary focus of Patch Manager, a capability of AWS Systems Manager, is on installing operating systems security-related updates on managed nodes. By default, Patch Manager doesn't install all available patches, but rather a smaller set of patches focused on security. (Ref https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-manager-how-it-works-selection.html)

AWS Global Accelerator

AWS Global Accelerator is a networking service that helps you improve the availability and performance of the applications that you offer to your global users. AWS Global Accelerator is easy to set up, configure, and manage. It provides static IP addresses that provide a fixed entry point to your applications and eliminates the complexity of managing specific IP addresses for different AWS Regions and Availability Zones (AZs). **AWS Global Accelerator always routes user traffic to the optimal endpoint based on performance, reacting instantly to changes in application health, your user's location, and policies that you configure.**

Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP.

With AWS Global Accelerator, you are provided two global static customer-facing IPs to simplify traffic management. An easy and effective way to bring down the number of IP addresses allowed by the firewall and easily manage the entire network infrastructure. You On the back end, add or remove your AWS application origins, such as Network Load Balancers, Application Load Balancers, elastic IP address (EIP), and Amazon EC2 Instances, without making user-facing changes. To mitigate endpoint failure, AWS Global Accelerator automatically re-routes your traffic to your nearest healthy available endpoint.

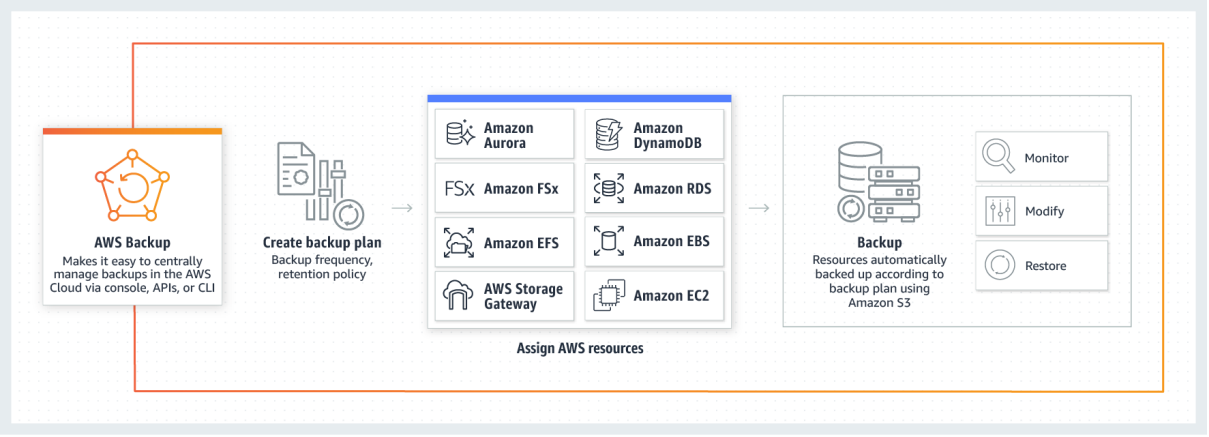
AWS VPN CloudHub

If you have multiple AWS Site-to-Site VPN connections, you can provide secure communication between sites using the AWS VPN CloudHub. This enables your remote sites to communicate with each other, and not just with the VPC. Sites that use AWS Direct Connect connections to the virtual private gateway can also be part

of the AWS VPN CloudHub. The VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC. This design is suitable if you have multiple branch offices and existing internet connections and would like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices.

AWS Backup

AWS Backup is a centralized backup service that makes it easy and cost-effective for you to backup your application data across AWS services in the AWS Cloud, helping you meet your business and regulatory backup compliance requirements. AWS Backup makes protecting your AWS storage volumes, databases, and file systems simple by providing a central place where you can configure and audit the AWS resources you want to backup, automate backup scheduling, set retention policies, and monitor all recent backup and restore activity.



Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking. After you create a hosted zone for your domain, such as example.com, you create records to tell the Domain Name System (DNS) how you want traffic to be routed for that domain.

For example, you might create records that cause DNS to do the following:

- Route Internet traffic for example.com to the IP address of a host in your data center.

- Route email for that domain (jose.rizal@tutorialsdojo.com) to a mail server (mail.tutorialsdojo.com).
- Route traffic for a subdomain called operations.manila.tutorialsdojo.com to the IP address of a different host.

Each record includes the name of a domain or a subdomain, a record type (for example, a record with a type of MX routes email), and other information applicable to the record type (for MX records, the hostname of one or more mail servers and a priority for each server).

You can create an alias record at the top node of a DNS namespace, also known as the zone apex, however, you cannot create a CNAME record for the top node of the DNS namespace. So, if you register the DNS name covid19survey.com, the zone apex is covid19survey.com. You can't create a CNAME record for covid19survey.com, but **you can create an alias record for covid19survey.com that routes traffic to www.covid19survey.com.**

For each VPC that you want to associate with the Route 53 hosted zone, change the following VPC settings to true:

enableDnsHostnames
enableDnsSupport

Routing Policies

Latency Routing lets Amazon Route 53 serve user requests from the AWS Region that provides the lowest latency. It does not, however, guarantee that users in the same geographic region will be served from the same location.

Geoproximity Routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a bias. A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource.

Geolocation Routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.

Weighted Routing lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (subdomain.tutorialsdojo.com) and choose how much traffic is routed to each resource. This can be useful for load balancing and testing new versions of software.

Failover

You can use **Route 53 health checking** to configure active-active and active-passive failover configurations. You configure active-active failover using any routing policy (or combination of routing policies) other than failover, and you configure active-passive failover using the failover routing policy.

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.

Amazon API Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on **AWS Lambda**, or any web application. Since it can use AWS Lambda, you can run your APIs without servers.

Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.

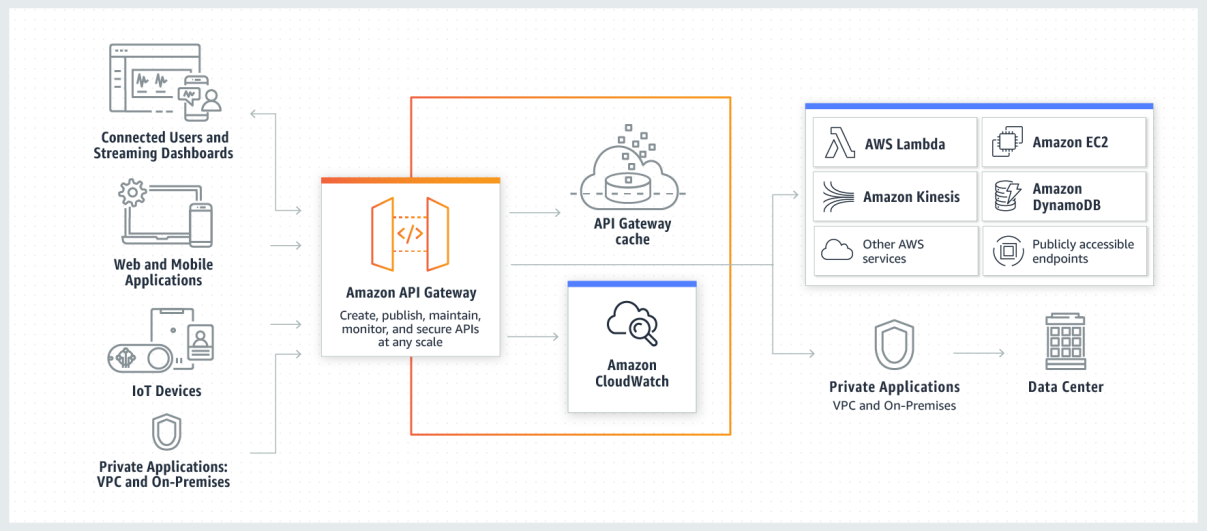
Enables you to build RESTful APIs and WebSocket APIs that are optimized for **serverless workloads.**

Implementing a **canary release deployment strategy** for the API Gateway is a great way to ensure your APIs remain stable and reliable. This strategy involves releasing a new version of your API to a small subset of users, allowing you to test the latest version in a controlled environment.

You can enable Amazon API caching in Amazon API Gateway to cache your endpoint's responses. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API. When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds.

Amazon API Gateway provides throttling at multiple levels including global and by service call. Throttling limits can be set for standard rates and bursts. For example, API owners can set a rate limit of 1,000 requests per second for a specific method in their REST APIs, and also configure Amazon API Gateway to handle a burst of 2,000 requests per second for a few seconds. Amazon API Gateway tracks the number of requests per second. Any request over the limit will receive a **429 HTTP response**. The client SDKs generated by Amazon API Gateway retry calls automatically when met with this response. Hence, enabling throttling limits and result caching in API Gateway is the correct answer.

You can add caching to API calls by provisioning an Amazon API Gateway cache and specifying its size in gigabytes. The cache is provisioned for a specific stage of your APIs. This improves performance and reduces the traffic sent to your back end. Cache settings allow you to control the way the cache key is built and the time-to-live (TTL) of the data stored for each method. Amazon API Gateway also exposes management APIs that help you invalidate the cache for each stage.



Amazon CloudFront

A web service that speeds up distribution of your static and dynamic web content to your users. **A Content Delivery Network (CDN) service**.

It delivers your content through a worldwide network of data centers called **edge locations**. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency, so that content is delivered with the best possible performance.

If you have objects that are smaller than 1GB or if the data set is less than 1GB in size, you should consider using Amazon CloudFront's PUT/POST commands for optimal performance.

You can set up Amazon CloudFront with origin failover for scenarios that require high availability.

Use field level encryption in Amazon CloudFront to protect sensitive data for specific content.

Many companies that distribute content over the Internet want to restrict access to documents, business data, media streams, or content that is intended for selected users, for example, users who have paid a fee. **To securely serve this private content by using CloudFront, you can do the following:**

- Require that your users access your private content by using special **CloudFront signed URLs or signed cookies**.
- Require that your users access your Amazon S3 content by using **CloudFront URLs, not Amazon S3 URLs**. Requiring CloudFront URLs isn't necessary, but it is recommended to prevent users from bypassing the restrictions that you

specify in signed URLs or signed cookies. You can do this by setting up an **origin access identity (OAI)** for your Amazon S3 bucket. You can also configure the custom headers for a private HTTP server or an Amazon S3 bucket configured as a website endpoint.

Configure an origin access identity (OAI) and associate it with the Amazon CloudFront distribution. Set up the permissions in the Amazon S3 bucket policy so that only the OAI can read the objects.

All objects and buckets by default are private. The pre-signed URLs are useful if you want your user/customer to be able to upload a specific object to your bucket, but you don't require them to have AWS security credentials or permissions.

Amazon Kinesis

Kinesis Data Stream

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources. **You can use an AWS Lambda function to process records in Amazon KDS**. By default, Lambda invokes your function as soon as records are available in the stream. Lambda can process up to 10 batches in each shard simultaneously. If you increase the number of concurrent batches per shard, Lambda still ensures in-order processing at the partition-key level.

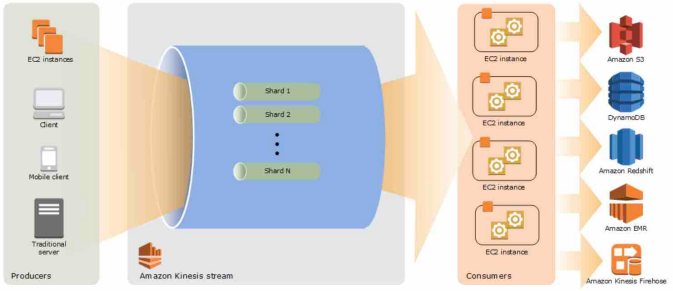
Amazon Kinesis Data Streams enables real-time processing of streaming big data.

The time period from when a record is added to when it is no longer accessible is called the **retention period**. A Kinesis data stream stores records from **24 hours by default** to a maximum of 8760 hours (365 days).

`ProvisionedThroughputExceededException` **means that you need to use batch messages. Increasing the number of shards is a short term fix and will substantially increase the cost.**

Kinesis Data Streams High-Level Architecture

The following diagram illustrates the high-level architecture of Kinesis Data Streams. The *producers* continually push data to Kinesis Data Streams, and the *consumers* process the data in real time. Consumers (such as a custom application running on Amazon EC2 or an Amazon Kinesis Data Firehose delivery stream) can store their results using an AWS service such as Amazon DynamoDB, Amazon Redshift, or Amazon S3.



Kinesis Data Streams Terminology

Kinesis Data Stream

A *Kinesis data stream* is a set of shards. Each shard has a sequence of data records. Each data record has a sequence number that is assigned by Kinesis Data Streams.

Data Record

A *data record* is the unit of data stored in a Kinesis data stream. Data records are composed of a sequence number, a partition key, and a data blob, which is an immutable sequence of bytes. Kinesis Data Streams does not inspect, interpret, or change the data in the blob in any way. A data blob can be up to 1 MB.

Capacity Mode

A data stream *capacity mode* determines how capacity is managed and how you are charged for the usage of your data stream. Currently, in Kinesis Data Streams, you can choose between an **on-demand** mode and a **provisioned** mode for your data streams. For more information, see Choosing the Data Stream Capacity Mode.

With the **on-demand** mode, Kinesis Data Streams automatically manages the shards in order to provide the necessary throughput. You are charged only for the actual throughput that you use and Kinesis Data Streams automatically accommodates your workloads' throughput needs as they ramp up or down. For more information, see On-demand Mode.

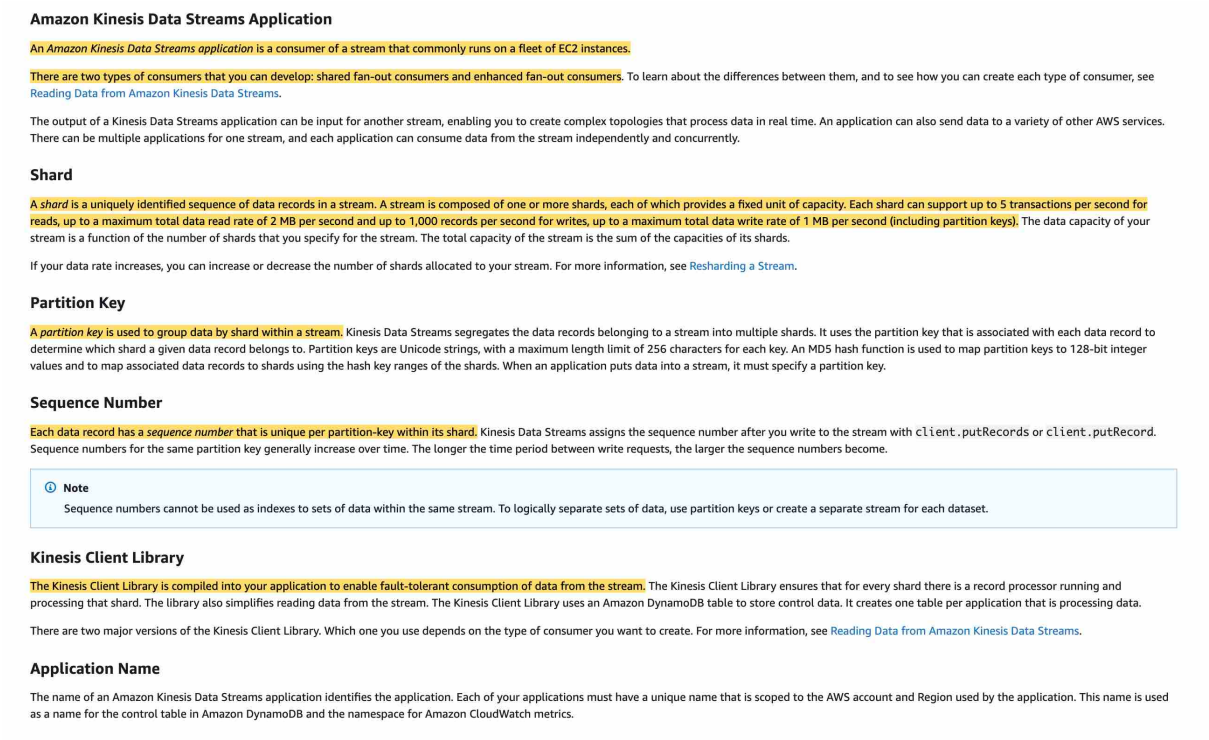
With the **provisioned** mode, you must specify the number of shards for the data stream. The total capacity of a data stream is the sum of the capacities of its shards. You can increase or decrease the number of shards in a data stream as needed and you are charged for the number of shards at an hourly rate. For more information, see Provisioned Mode.

Retention Period

The *retention period* is the length of time that data records are accessible after they are added to the stream. A stream's retention period is set to a default of 24 hours after creation. You can increase the retention period up to 8760 hours (365 days) using the IncreaseStreamRetentionPeriod operation, and decrease the retention period down to a minimum of 24 hours using the DecreaseStreamRetentionPeriod operation. Additional charges apply for streams with a retention period set to more than 24 hours. For more information, see Amazon Kinesis Data Streams Pricing.

Producer

Producers put records into Amazon Kinesis Data Streams. For example, a web server sending log data to a stream is a producer.



Amazon Kinesis Data Streams Application

An *Amazon Kinesis Data Streams application* is a consumer of a stream that commonly runs on a fleet of EC2 instances.

There are two types of consumers that you can develop: shared fan-out consumers and enhanced fan-out consumers. To learn about the differences between them, and to see how you can create each type of consumer, see Reading Data from Amazon Kinesis Data Streams.

The output of a Kinesis Data Streams application can be input for another stream, enabling you to create complex topologies that process data in real time. An application can also send data to a variety of other AWS services. There can be multiple applications for one stream, and each application can consume data from the stream independently and concurrently.

Shard

A *shard* is a uniquely identified sequence of data records in a stream. A stream is composed of one or more shards, each of which provides a fixed unit of capacity. Each shard can support up to 5 transactions per second for reads, up to a maximum total data read rate of 2 MB per second and up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second (including partition keys). The data capacity of your stream is a function of the number of shards that you specify for the stream. The total capacity of the stream is the sum of the capacities of its shards.

If your data rate increases, you can increase or decrease the number of shards allocated to your stream. For more information, see Resharding a Stream.

Partition Key

A *partition key* is used to group data by shard within a stream. Kinesis Data Streams segregates the data records belonging to a stream into multiple shards. It uses the partition key that is associated with each data record to determine which shard a given data record belongs to. Partition keys are Unicode strings, with a maximum length limit of 256 characters for each key. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards using the hash key ranges of the shards. When an application puts data into a stream, it must specify a partition key.

Sequence Number

Each data record has a *sequence number* that is unique per partition-key within its shard. Kinesis Data Streams assigns the sequence number after you write to the stream with `client.putRecords` or `client.putRecord`. Sequence numbers for the same partition key generally increase over time. The longer the time period between write requests, the larger the sequence numbers become.

> **Note**
> Sequence numbers cannot be used as indexes to sets of data within the same stream. To logically separate sets of data, use partition keys or create a separate stream for each dataset.

Kinesis Client Library

The Kinesis Client Library is compiled into your application to enable fault-tolerant consumption of data from the stream. The Kinesis Client Library ensures that for every shard there is a record processor running and processing that shard. The library also simplifies reading data from the stream. The Kinesis Client Library uses an Amazon DynamoDB table to store control data. It creates one table per application that is processing data.

There are two major versions of the Kinesis Client Library. Which one you use depends on the type of consumer you want to create. For more information, see Reading Data from Amazon Kinesis Data Streams.

Application Name

The name of an Amazon Kinesis Data Streams application identifies the application. Each of your applications must have a unique name that is scoped to the AWS account and Region used by the application. This name is used as a name for the control table in Amazon DynamoDB and the namespace for Amazon CloudWatch metrics.

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose is the easiest way to load streaming data into data stores and analytics tools. It can capture, transform, and load streaming data **into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk**, enabling near real-time analytics with existing business intelligence tools and dashboards you're already using today. It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.

Firehose cannot directly write into a DynamoDB table.

Kinesis Data Streams cannot directly write the output to Amazon S3. Unlike Amazon Kinesis Data Firehose, KDS does not offer a ready-made integration via an intermediary AWS Lambda function to reliably dump data into Amazon S3.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics is the easiest way to analyze streaming data in real-time. **You can quickly build SQL queries and sophisticated Java applications** using built-in templates and operators for common processing functions to organize, transform, aggregate, and analyze data at any scale. Kinesis Data Analytics enables

you to easily and quickly build queries and sophisticated streaming applications in three simple steps: setup your streaming data sources, write your queries or streaming applications and set up your destination for processed data. **Amazon Kinesis Data Analytics is used to build SQL queries and sophisticated Java applications.**

Amazon Athena

Amazon Athena is a serverless, interactive analytics service built on open-source frameworks, supporting open-table and file formats. **Athena provides a simplified, flexible way to analyze petabytes of data where it lives. Analyze data or build applications from an Amazon Simple Storage Service (S3) data lake and 30 data sources**, including on-premises data sources or other cloud systems using SQL or Python. Athena is built on open-source Trino and Presto engines and Apache Spark frameworks, with no provisioning or configuration effort required.

Submit a single SQL query to analyze data in relational, nonrelational, object, and custom data sources running on S3, on premises or in multicloud environments.

Use ML models in SQL queries or Python to simplify complex tasks, such as anomaly detection, customer cohort analysis, and sales predictions.

Deploy a reconciliation tool with an engine built for the cloud to validate vast amounts of data effectively at scale.

Apache Parquet

Apache Parquet is an open-source columnar storage format that is 2x faster to unload and takes up 6x less storage in Amazon S3 as compared to other text formats. One can COPY Apache Parquet and Apache ORC file formats from Amazon S3 to your Amazon Redshift cluster. Using AWS Glue, one can configure and run a job to transform CSV data to Parquet. Parquet is a columnar format that is well suited for AWS analytics services like Amazon Athena and Amazon Redshift Spectrum.

The comma-separated value (.csv) format is the default storage format for Amazon S3 target objects. For more compact storage and faster queries, you can instead use Apache Parquet (.parquet) as the storage format.

Amazon QuickSight

Amazon QuickSight powers data-driven organizations with unified business intelligence (BI) at hyperscale. With QuickSight, all users can meet varying analytic needs from the same source of truth through modern interactive dashboards, paginated reports, embedded analytics, and natural language queries. With Amazon Q in QuickSight, business analysts and business users can use natural language to build, discover, and share meaningful insights in seconds, turning insights into impact faster.

Amazon Q in QuickSight enhances business productivity using Generative BI capabilities to accelerate decision-making. With new **dashboard authoring capabilities** in Amazon Q, business analysts can use natural language prompts to build, discover, and share meaningful insights in seconds. Amazon Q makes it easier for business users to understand data with executive summaries, a new context-aware data Q&A experience, and data stories.

Create, schedule, and share reports and data exports from a single fully managed cloud-based BI service. Get business-critical information to users how and when they need it with critical operational reports and dashboards in the same solution.

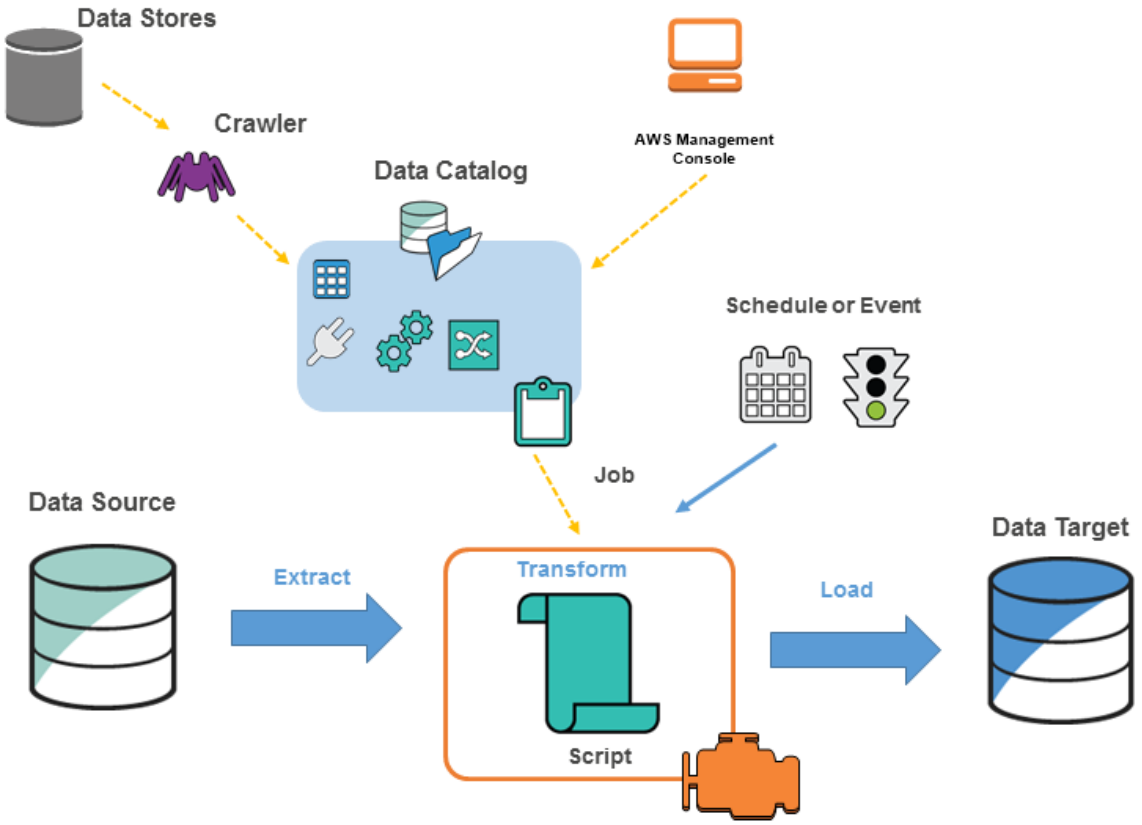
Amazon OpenSearch

OpenSearch is a fully open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and clickstream analysis.

AWS Glue

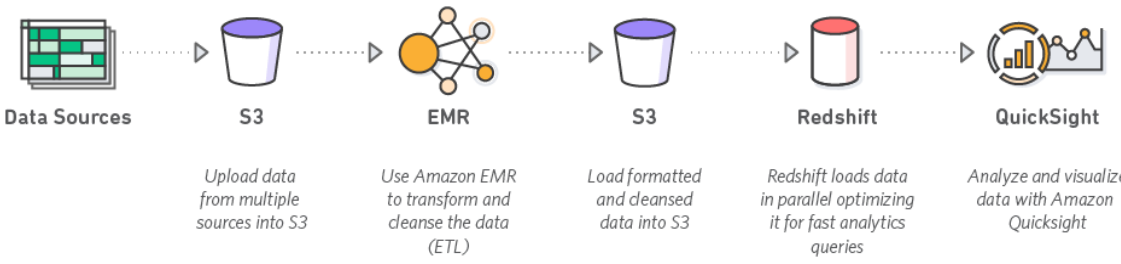
AWS Glue is a **fully managed extract, transform, and load (ETL) service**. AWS Glue makes it cost-effective to categorize your data, clean it, enrich it, and move it reliably between various data stores and data streams. This pattern provides different job types in AWS Glue and uses three different scripts to demonstrate authoring ETL jobs.

AWS Glue has a minimum billing duration of 1 minute (Glue version 2.0 and later), which is not cost-effective for processing small amounts of data quickly (in 10 seconds).



Amazon EMR

Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. By using these frameworks and related open-source projects such as Apache Hive and Apache Pig, you can **process data for analytics purposes and business intelligence workloads**. Additionally, you can use Amazon EMR to **transform and move large amounts of data into and out of other AWS data stores and databases such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB**.



Machine Learning Services

Amazon Comprehend Medical

Amazon Comprehend Medical is a natural language processing service that makes it easy to use machine learning to extract relevant medical information from unstructured text. Using Amazon Comprehend Medical, you can quickly and accurately gather information, such as medical conditions, medication, dosage, strength, and frequency from a variety of sources, like doctors' notes, clinical trial reports, and patient health records.

Amazon Textract

Amazon Textract is a machine learning (ML) service that automatically extracts text, handwriting, layout elements, and data from scanned documents. It goes beyond simple optical character recognition (OCR) to identify, understand, and extract specific data from documents. Today, many companies manually extract data from scanned documents such as PDFs, images, tables, and forms, or through simple OCR software that requires manual configuration (which often must be updated when the form changes). To overcome these manual and expensive processes, Textract uses ML to read and process any type of document, accurately extracting text, handwriting, tables, and other data with no manual effort. You can use one of our pretrained or custom features to quickly automate document processing, whether you're automating loans processing or extracting information from invoices and receipts. Textract provides you the ability to customize our pretrained features to meet the document processing needs specific to your business. Textract can extract the data in minutes instead of hours or days.

Amazon Lex

Amazon Lex enables you to build applications using a speech or text interface powered by the same technology that powers Amazon Alexa. Amazon Lex provides the deep functionality and flexibility of natural language understanding (NLU) and automatic speech recognition (ASR), so you can build highly engaging user experiences with lifelike conversational interactions and create new categories of products.

AWS Wavelength

AWS Wavelength combines the high bandwidth and ultralow latency of 5G networks with AWS compute and storage services so that developers can innovate and build a new class of applications.

Wavelength Zones are AWS infrastructure deployments that embed AWS compute and storage services within telecommunications providers' data centers at the edge of the 5G network, so application traffic can reach application servers running in Wavelength Zones without leaving the mobile providers' network. This prevents the latency that would result from multiple hops to the internet and enables customers to take full advantage of 5G networks. Wavelength Zones extend AWS to the 5G edge, delivering a consistent developer experience across multiple 5G networks around the world. Wavelength Zones also allow developers to build the next generation of ultra-low latency applications using the same familiar AWS services, APIs, tools, and functionality they already use today.

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.

You can deploy ACM certificates into AWS Elastic Load Balancing, Amazon CloudFront, Amazon API Gateway, and other integrated services. The most common application of this kind is a secure public website with significant traffic requirements.

ACM Private CA

This service is for enterprise customers building a public key infrastructure (PKI) inside the AWS cloud and is **intended for private use within an organization**. With ACM Private CA, you can **create your own CA hierarchy** and issue certificates with it for authenticating internal users, computers, applications, services, servers, and other devices and for signing computer code. Certificates issued by a private CA are trusted only within your organization, not on the internet.

CloudTrail

Amazon CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your Amazon Web Services account. **With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your Amazon Web Services infrastructure.**

By default, CloudTrail event log files are encrypted using Amazon S3 server-side encryption (SSE). **You can also choose to encrypt your log files with an AWS Key Management Service (AWS KMS) key**. You can store your log files in your bucket for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically. If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.

Amazon CloudWatch

Setup an Amazon CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, an EC2 Reboot CloudWatch Alarm Action can be used to reboot the instance.

CloudWatch has available Amazon EC2 Metrics for you to use for monitoring. CPU Utilization identifies the processing power required to run an application upon a selected instance. Network Utilization identifies the volume of incoming and outgoing network traffic to a single instance. Disk Reads metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application. However, there are certain metrics that are not readily available in CloudWatch such as memory utilization, disk space utilization, and many others which can be collected by setting up a custom metric.

Take note that there are certain differences between CloudWatch and Enhanced Monitoring Metrics. CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance, and **Enhanced Monitoring gathers its metrics from an agent on the instance**. As a result, you might find differences between the measurements, because the hypervisor layer performs a small amount of work. **Enhanced Monitoring is a feature of Amazon RDS.**

You need to prepare a **custom metric** using CloudWatch Monitoring Scripts which is written in Perl. You can also install **CloudWatch Agent** to collect more system-level metrics from Amazon EC2 instances. Here's the list of custom metrics that you can set up:

- Memory utilization
- Disk swap utilization

- Disk space utilization
- Page file utilization
- Log collection

CloudWatch Logs Insights

CloudWatch Logs Insights enables you to interactively search and analyze your log data in Amazon CloudWatch Logs. You can perform queries to help you quickly and effectively respond to operational issues. If an issue occurs, you can use CloudWatch Logs Insights to identify potential causes and validate deployed fixes.

CloudWatch Logs Insights includes a purpose-built query language with a few simple but powerful commands. CloudWatch Logs Insights provides sample queries, command descriptions, query autocompletion, and log field discovery to help you get started quickly. Sample queries are included for several types of AWS service logs.

CloudWatch Application Insights

Amazon CloudWatch Application Insights facilitates observability for your applications and underlying AWS resources. **It helps you set up the best monitors for your application resources to continuously analyze data for signs of problems with your applications**. Application Insights, which is powered by SageMaker and other AWS technologies, provides automated dashboards that show potential problems with monitored applications, which help you to quickly isolate ongoing issues with your applications and infrastructure. The enhanced visibility into the health of your applications that Application Insights provides helps reduce the "mean time to repair" (MTTR) to troubleshoot your application issues.

EventBridge

EventBridge is a serverless service that uses events to connect application components together, making it easier for you to build scalable event-driven applications. Event-driven architecture is a style of building loosely-coupled software systems that work together by emitting and responding to events. Event-driven architecture can help you boost agility and build reliable, scalable applications.

Amazon EventBridge is recommended when you want to build an application that reacts to events from SaaS applications and/or AWS services. Amazon EventBridge is the only event-based service that integrates directly with third-party SaaS partners.

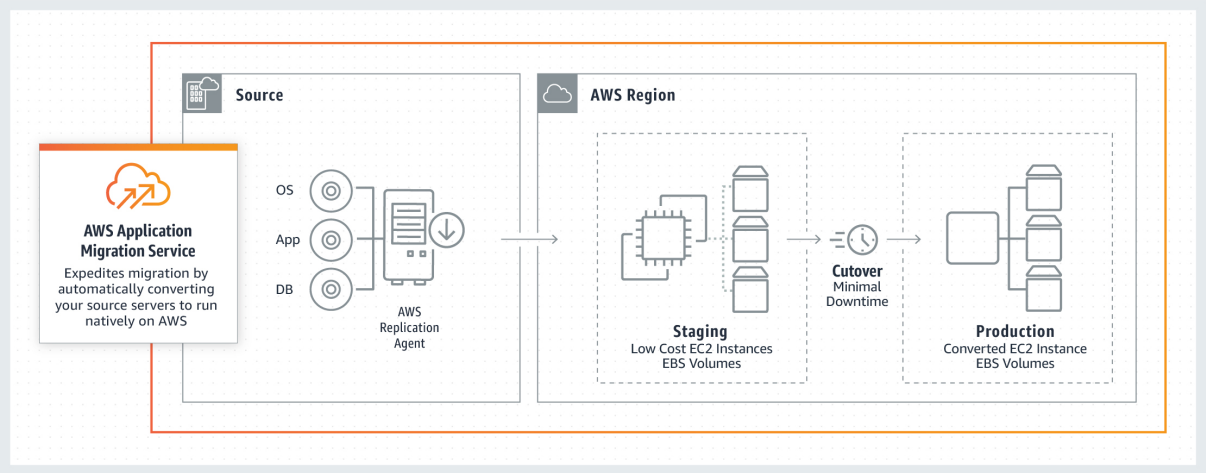
AWS Application Migration Service (AWS MGN)

AWS Application Migration Service (AWS MGN) is the primary migration service recommended for lift-and-shift migrations to AWS. AWS encourages customers who are currently using AWS Elastic Disaster Recovery to switch to AWS MGN for future migrations. AWS MGN enables organizations to move applications to AWS without having to make any changes to the applications, their architecture, or the migrated servers.

AWS Application Migration Service minimizes time-intensive, error-prone manual processes by automatically converting your source servers from physical, virtual machines, and cloud infrastructure to run natively on AWS.

The service simplifies your migration by enabling you to use the same automated process for a wide range of applications. By launching non-disruptive tests before migrating, you can be confident that your most critical applications such as SAP, Oracle, and SQL Server, will work seamlessly on AWS.

Implementation begins by installing the AWS Replication Agent on your source servers. When you launch Test or Cutover instances, AWS Application Migration Service automatically converts your source servers to boot and runs natively on AWS.



AWS Web Application Firewall (WAF)

AWS WAF is a web application firewall that lets you monitor the HTTP(S) requests that are forwarded to an Amazon CloudFront distribution, an Amazon

API Gateway REST API, an Application Load Balancer, or an AWS AppSync GraphQL API.

When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end-users. This means security doesn't come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on regional services, such as Application Load Balancer, Amazon API Gateway, and AWS AppSync, your rules run in the region and can be used to protect Internet-facing resources as well as internal resources.

Integrate WAF with AWS Firewall Manager to reuse the rules across all AWS accounts.

Each rule contains a statement that defines the inspection criteria and an action to take if a web request meets the criteria. When a web request meets the criteria, that's a match. You can configure rules to block matching requests, allow them through, count them, or run **CAPTCHA** controls against them.

A rate-based rule tracks the rate of requests for each originating IP address and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minute time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests.

AWS Firewall Manager

AWS Firewall Manager is a security management service which allows you to centrally configure and manage firewall rules across your accounts and applications in AWS Organizations. As new applications are created, Firewall Manager makes it easy to bring new applications and resources into compliance by enforcing a common set of security rules. Now you have a single service to build firewall rules, create security policies, and enforce them in a consistent, hierarchical manner across your entire infrastructure.

Supports:

- **AWS Web Application Firewall (AWS WAF)**
- **AWS Shield Advanced**
- **VPC Security Groups**
- **AWS Network Firewalls**

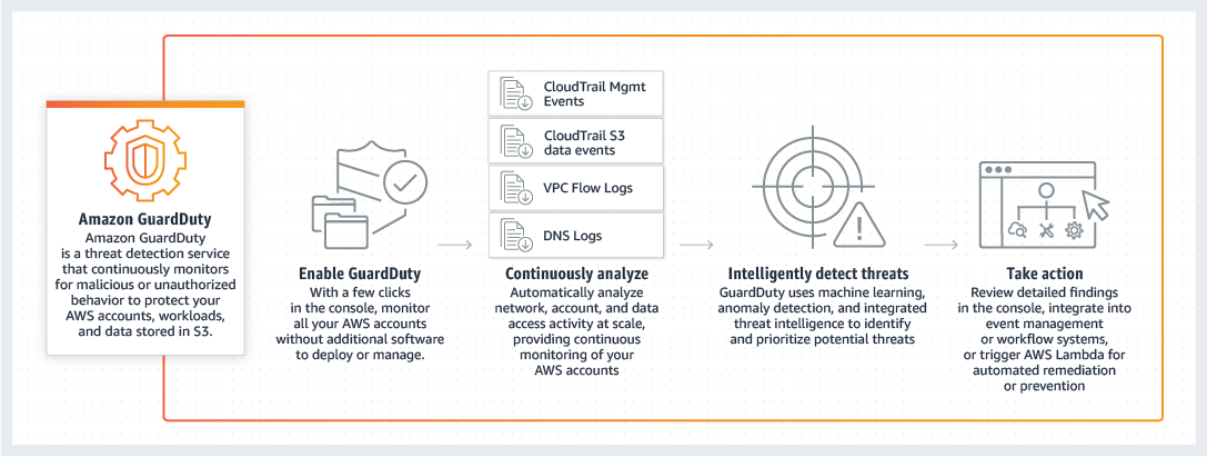
- **Amazon Route 53 Resolver DNS Firewall**

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 continuous streams of meta-data generated from your account and network activity found in AWS CloudTrail Events, Amazon VPC Flow Logs, and DNS Logs. It also uses integrated threat intelligence such as known malicious IP addresses, anomaly detection, and machine learning to identify threats more accurately.

Disabling the service will delete all remaining data, including your findings and configurations before relinquishing the service permissions and resetting the service.



Amazon Inspector

Amazon Inspector removes the operational overhead that is necessary to configure a vulnerability management solution. **Amazon Inspector works with both EC2 instances and container images in Amazon ECR to identify potential software vulnerabilities and to categorize the severity of the vulnerabilities.**

AWS Artifact

AWS Artifact is your go-to, central resource for compliance-related information that matters to you. **It provides on-demand access to AWS's 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).

AWS Health

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.

AWS Config

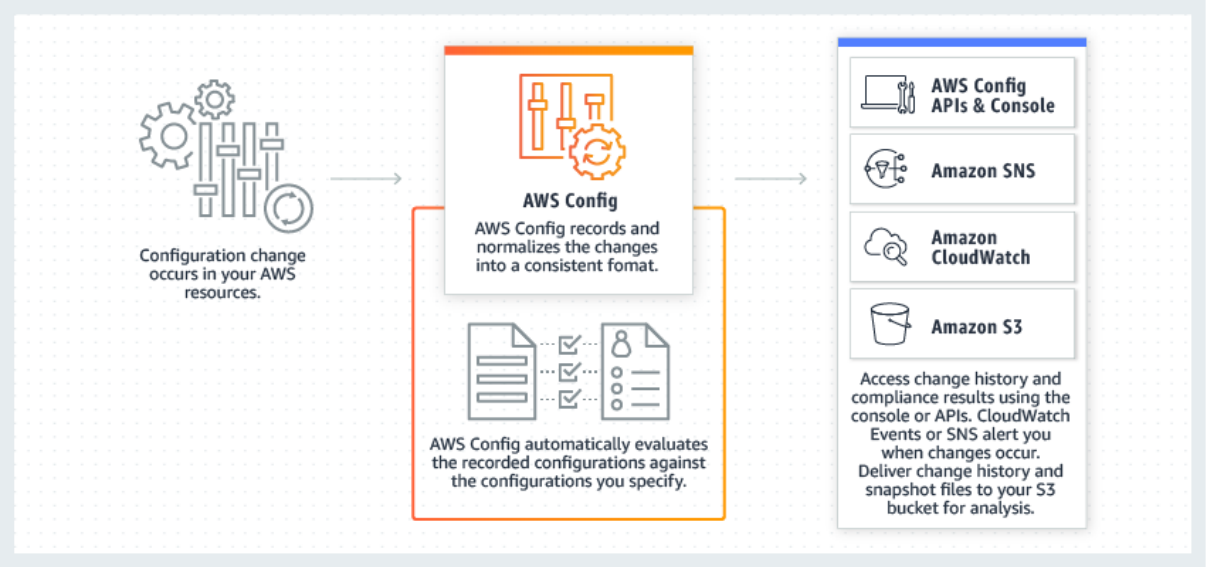
AWS Config is a service that enables you to **assess, audit, and evaluate the configurations of your AWS resources**. **Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations**. With 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.

You can assign metadata to your AWS resources in the form of tags. Each tag is a label consisting of a user-defined key and value. Tags can help you manage, identify, organize, search for, and filter resources. You can create tags to categorize resources by purpose, owner, environment, or other criteria.

You can use tags to control access by restricting IAM permissions based on specific tags or tag values. For example, IAM user or role permissions can include

conditions to limit EC2 API calls to specific environments (such as development, test, or production) based on their tags.

Manually creating a custom AWS Config rule to check for SSL expiry is unnecessary. **AWS Config already provides a built-in acm-certificate-expiration-check managed rule that you can use.**



AWS Control Tower

AWS Control Tower provides a single location to easily set up your new well-architected multi-account environment and govern your AWS workloads with rules for security, operations, and internal compliance. You can automate the setup of your AWS environment with **best-practices blueprints for multi-account structure, identity, access management, and account provisioning workflow**. For ongoing governance, you can select and apply pre-packaged policies enterprise-wide or to specific groups of accounts.

AWS Control Tower provides three methods for creating member accounts:

- Through the Account Factory console that is part of AWS Service Catalog.
- Through the Enroll account feature within AWS Control Tower.
- From your AWS Control Tower landing zone's management account, using Lambda code and appropriate IAM roles.

AWS Control Tower offers "guardrails" for ongoing governance of your AWS environment. Guardrails provide governance controls by preventing the deployment of resources that don’t conform to selected policies or detecting

non-conformance of provisioned resources. AWS Control Tower automatically impements guardrails using multiple building blocks such as AWS CloudFormation to establish a baseline, AWS Organizations service control policies (SCPs) to prevent configuration changes, and AWS Config rules to continuously detect non-conformance.

To save time and resources, you can use AWS Control Tower to **automate account creation**. With the appropriate user group permissions, you can specify standardized baselines and network configurations for all accounts in the organization.

AWS Trusted Advisor Service

Trusted Advisor inspects your AWS environment, and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. If you have a Basic or Developer Support plan, you can use the Trusted Advisor console to access all checks in the Service Limits category and six checks in the Security category.

AWS Cost Explorer

Cost Explorer is a tool within AWS that allows for detailed analysis of AWS costs and usage. It provides an interactive interface where you can visualize your data, detect cost trends, and dive deeper into cost drivers. **By using its granular filtering features, you can specifically look at EC2 costs, filter by instance type, and compare the costs over the last two months**. This direct approach requires no additional setup or external tools and is designed for in-depth cost analysis, making it the most efficient choice for the task at hand.

Use AWS Cost Explorer Resource Optimization to get a report of Amazon EC2 instances that are either idle or have low utilization and use AWS Compute Optimizer to look at instance type recommendations.

AWS Organizations

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage. AWS Organizations includes account management and **consolidated billing capabilities** that enable you to better meet the budgetary, security, and compliance needs of your business. As an administrator of an

organization, you can create accounts in your organization and invite existing accounts to join the organization.

In addition to this, **you can also configure a service control policy (SCP) to manage your AWS accounts**. SCPs help you enforce policies across your organization and control the services and features accessible to your other account. This way, you can ensure that your organization's resources are used only as intended and prevent unauthorized access. You can provide secure and centralized management of your AWS accounts by setting up AWS Organization, integrating AWS IAM Identity Center with your corporate directory service, and configuring SCPs. This simplifies your management process and helps you maintain better control over your resources.

Service control policy (SCP) does not affect service-linked role.

To migrate accounts from one organization to another, you must have root or IAM access to both the member and master accounts. Here are the steps to follow: 1. Remove the member account from the old organization 2. Send an invite to the member account from the new Organization 3. Accept the invite to the new organization from the member account

Consolidated Billing

You can use the consolidated billing feature in AWS Organizations to consolidate billing and payment for multiple AWS accounts. Every organization in AWS Organizations has a management account that pays the charges of all the member accounts.

Consolidated billing has the following benefits:

- **One bill** – You get one bill for multiple accounts.
- **Easy tracking** – You can track the charges across multiple accounts and download the combined cost and usage data.
- **Combined usage** – You can combine the usage across all accounts in the organization to share the volume pricing discounts, Reserved Instance discounts, and Savings Plans. This can result in a lower charge for your project, department, or company than with individual standalone accounts.
- **No extra fee** – Consolidated billing is offered at no additional cost.

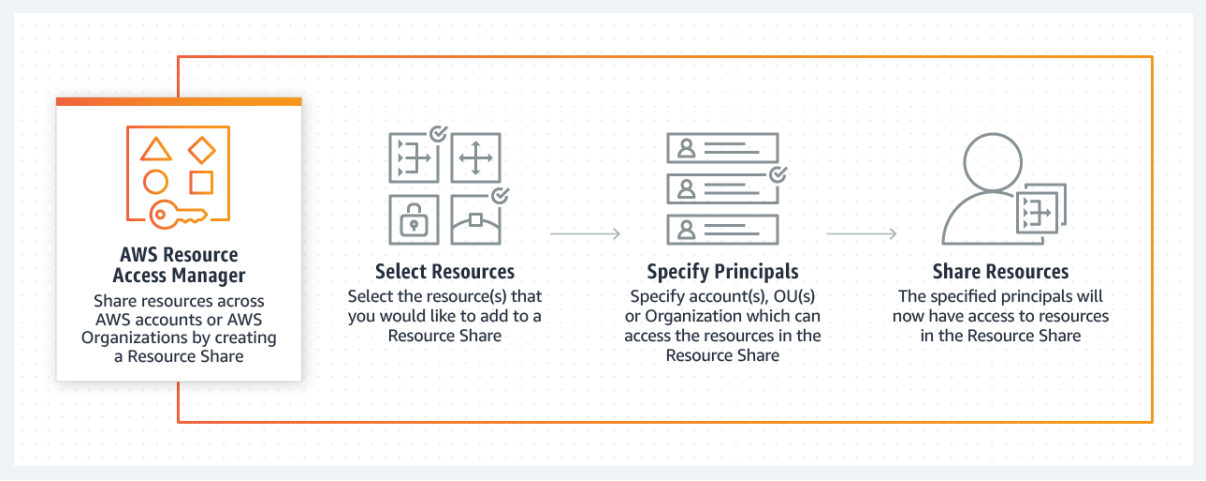
AWS Resource Manager (RAM)

AWS Resource Access Manager (RAM) is a service that enables you to easily and securely share AWS resources with any AWS account or within your AWS Organization. You can share AWS Transit Gateways, Subnets, AWS License Manager configurations, and Amazon Route 53 Resolver rules resources with RAM.

Many organizations use multiple accounts to create administrative or billing isolation, and limit the impact of errors. **RAM eliminates the need to create duplicate resources in multiple accounts, reducing the operational overhead of managing those resources in every single account you own**. You can create resources centrally in a multi-account environment, and use RAM to share those resources across accounts in three simple steps: create a Resource Share, specify resources, and specify accounts. RAM is available to you at no additional charge.

You can procure AWS resources centrally, and use RAM to share resources such as subnets or License Manager configurations with other accounts. This eliminates the need to provision duplicate resources in every account in a multi-account environment, reducing the operational overhead of managing those resources in every account.

Use case: **You have multiple AWS accounts within a single AWS Region managed by AWS Organizations and you would like to ensure all Amazon EC2 instances in all these accounts can communicate privately. Create a virtual private cloud (VPC) in an account and share one or more of its subnets with the other accounts using Resource Access Manager is the cheapest option.**



Amazon Workspaces

Amazon WorkSpaces is a managed, secure cloud desktop service. You can use Amazon WorkSpaces to provision either Windows or Linux desktops in just a few

minutes and quickly scale to provide thousands of desktops to workers across the globe.

Architecture and Design

Segregate users in different environments

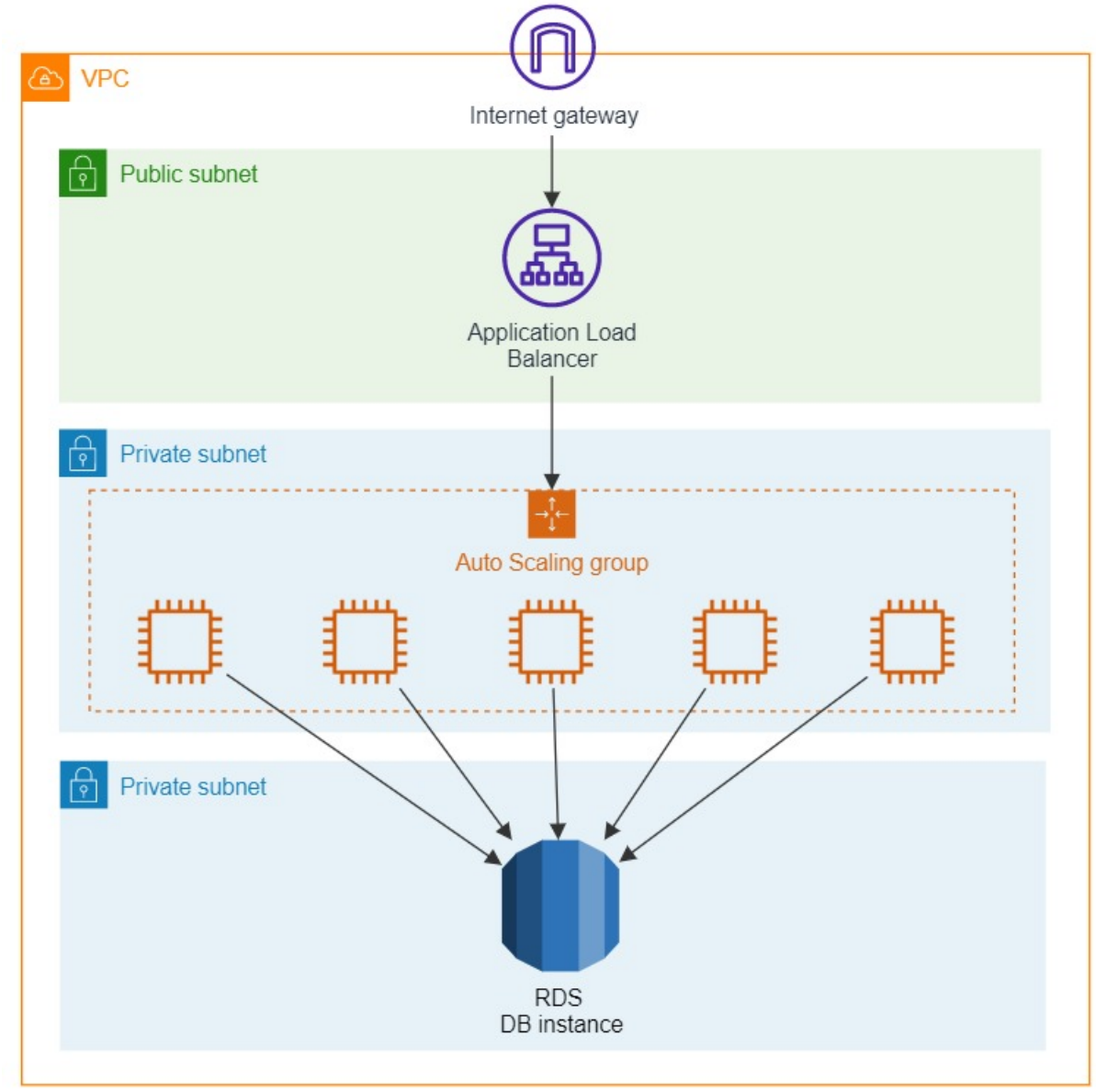
For this scenario, the best way to achieve the required solution is to use a combination of Tags and IAM policies. You can define the tags on the UAT and production EC2 instances and add a condition to the IAM policy which allows access to specific tags.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you've assigned to it.

By default, IAM users don't have permission to create or modify Amazon EC2 resources or perform tasks using the Amazon EC2 API. (This means that they also can't do so using the Amazon EC2 console or CLI.) To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need and then attach those policies to the IAM users or groups that require those permissions.

Defense-in-depth

A defense-in-depth strategy is one of the design principles for security in the AWS cloud. This strategy entails implementing security controls at multiple layers (for example, edge of network, VPC, load balancing, every instance and compute service, operating system, application, and code).



Prevent Distributed Denial of Service (DDoS)

A Denial of Service (DoS) attack is an attack that can make your website or application unavailable to end users. To achieve this, attackers use a variety of techniques that consume network or other resources, disrupting access for legitimate end users.

To protect your system from DDoS attack, you can do the following:

- **Use an Amazon CloudFront service for distributing both static and dynamic content.**
- **Use an Application Load Balancer with Auto Scaling groups for your EC2 instances. Prevent direct Internet traffic to your Amazon RDS database by**

deploying it to a new private subnet.

- **Set up alerts in Amazon CloudWatch to look for high `Network In` and CPU utilization metrics.**

Services that are available within AWS Regions, like Elastic Load Balancing and Amazon Elastic Compute Cloud (EC2), allow you to build Distributed Denial of Service resiliency and scale to handle unexpected volumes of traffic within a given region. Services that are available in AWS edge locations, like Amazon CloudFront, AWS WAF, Amazon Route53, and Amazon API Gateway, allow you to take advantage of a global network of edge locations that can provide your application with greater fault tolerance and increased scale for managing larger volumes of traffic.

In addition, you can also use **AWS Shield** and **AWS WAF** to fortify your cloud network. **AWS Shield is a managed DDoS protection service** that is available in two tiers: Standard and Advanced. AWS Shield Standard applies always-on detection and inline mitigation techniques, such as deterministic packet filtering and priority-based traffic shaping, to minimize application downtime and latency.

AWS WAF is a web application firewall that helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. You can use AWS WAF to define customizable web security rules that control which traffic accesses your web applications. If you use AWS Shield Advanced, you can use AWS WAF at no extra cost for those protected resources and can engage the DRT to create WAF rules.

Expiring certificates notifications

The first option uses an AWS ACM built-in Certificate Expiration event, which is raised through Amazon EventBridge, to invoke a Lambda function. In this option, the function is configured to publish the result as a finding in Security Hub, and also as an SNS topic used for email subscriptions. As a result, an administrator can be notified of a specific expiring certificate, or an IT service management (ITSM) system can automatically open a case or incident through email or SNS.

The second option uses the recently launched **DaysToExpiry** metric to schedule a batch search of expiring certificates and to log all the findings. The metric also provides a single SNS notification for all expiring certificates.

Third option: **leverage AWS Config managed rule to check if any third-party SSL/TLS certificates imported into ACM are marked for expiration within 30**

days. Configure the rule to trigger an Amazon SNS notification to the security team if any certificate expires within 30 days

Build a shared services Amazon Virtual Private Cloud (Amazon VPC)

Consider an organization that has built a hub-and-spoke network with AWS Transit Gateway. VPCs have been provisioned into multiple AWS accounts, perhaps to facilitate network isolation or to enable delegated network administration. When deploying distributed architectures such as this, **a popular approach is to build a "shared services VPC”**, which provides access to services required by workloads in each of the VPCs. This might include directory services or VPC endpoints. Sharing resources from a central location instead of building them in each VPC may reduce administrative overhead and cost.

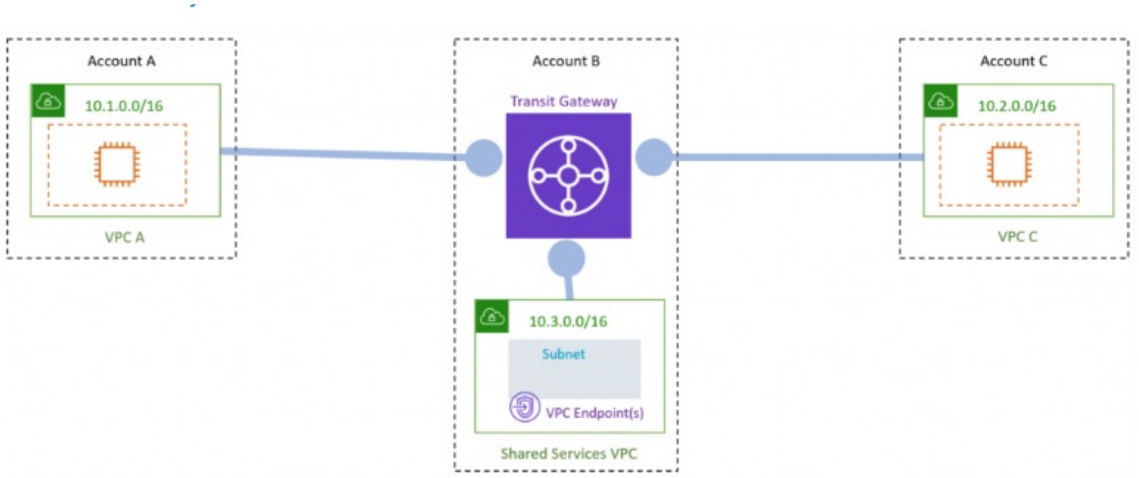


Figure 3: Centralized VPC Endpoints (multiple VPCs)

Resilient Direct Connect faster then 1 Gbps

Maximum resilience is achieved by separate connections terminating on separate devices in more than one location. This configuration offers customers maximum resilience to failure. As shown in the figure above, such a topology provides resilience to device failure, connectivity failure, and complete location failure. You can use Direct Connect Gateway to access any AWS Region (except AWS Regions in China) from any AWS Direct Connect locations.

Maximum Resiliency for Critical Workloads