

# AWS Certified Solutions Architect Associate (SAA-C02) – Exam Guide

Mike Gibbs MS, MBA, CCIE #7417, GPC-PCA, AWS-CSA

Nathan Vontz

[www.gocloudarchitects.com](http://www.gocloudarchitects.com)

AWS Certified Solutions Architect Associate (SAA-C02) – Exam Guide by Mike Gibbs MS, MBA, CCIE #7417, GCP- PCA, AWS-CSA & Nathan Vontz

Port Saint Lucie, FL 34953

[www.gocloudarchitects.com](http://www.gocloudarchitects.com)

© 2021 Mike Gibbs & Nathan Vontz

All rights reserved. No portion of this book may be reproduced in any form without permission from the publisher, except as permitted by U.S. copyright law.

For permissions contact:

[elitetechcareers@gmail.com](mailto:elitetechcareers@gmail.com)

## Disclaimer

LIABILITY: Mike Gibbs, Nathan Vontz, GoCloudArchitects and its organizations and agents specifically DISCLAIM LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES and assumes no responsibility or liability for any loss or damage suffered by any person as a result of the use or misuse of our these materials or any of the information or content on our website, blogs and social media posts. Mike Gibbs, Nathan Vontz, GoCloudArchitects and its organizations and agents assume or undertakes NO LIABILITY for any loss or damage suffered as a result of the use or misuse of any information or content or any reliance thereon.

## Table of Contents

<b><u>Chapter 1 Introduction to Cloud Computing</u></b>	<b>1</b>
<a href="#">The History of the Enterprise Network and Data Center</a>	1
<a href="#">Connections to the Cloud</a>	5
<a href="#">How to Access and Manage Resources in the Cloud</a>	9
<b><u>Chapter 2 Storage Options on the AWS Cloud Platform</u></b>	<b>11</b>
<a href="#">AWS Primary Storage Options</a>	11
<a href="#">Amazon Simple Storage Service (S3)</a>	12
<a href="#">Elastic Block Storage (EBS)</a>	21
<a href="#">Elastic File System (EFS)</a>	24
<a href="#">AWS Storage Gateway</a>	25
<a href="#">Migrating Data to AWS</a>	27
<a href="#">Storage for Collaboration</a>	28
<a href="#">Amazon FSx for Windows</a>	28
<b><u>Chapter 3 Computing on the AWS Platform (EC2)</u></b>	<b>31</b>
<a href="#">Amazon Machine Images (AMI)</a>	32
<a href="#">Autoscaling</a>	33
<a href="#">Instance Purchasing Options</a>	33
<a href="#">Tenancy Options</a>	35
<a href="#">Securing EC2 Access</a>	35
<a href="#">IP Addresses for EC2 Instances</a>	37
<a href="#">Accessing EC2 Instances</a>	37
<b><u>Chapter 4 Introduction to Databases</u></b>	<b>39</b>
<a href="#">Relational Databases</a>	46
<a href="#">DynamoDB</a>	46
<a href="#">Data Warehousing on AWS</a>	48
<a href="#">Database Storage Options</a>	49
<a href="#">Backing Up the Database</a>	50
<a href="#">Scaling the Database</a>	51
<a href="#">Designing a High-Availability Database Architecture</a>	55
<b><u>Chapter 5 The AWS Virtual Private Cloud</u></b>	<b>58</b>
<a href="#">The OSI Model</a>	58
<a href="#">IP Addressing</a>	59
<a href="#">Routing Tables and Routing</a>	65
<a href="#">Internet Gateways</a>	68
<a href="#">NAT Instances</a>	69
<a href="#">NAT Gateway</a>	70
<a href="#">Elastic IP Addresses (EIPs)</a>	71

<a href="#">Endpoints</a>	72
<a href="#">VPC Peering</a>	74
<a href="#">AWS CloudHub</a>	77
<a href="#">Network Access Control Lists (NACL)</a>	79
<a href="#">Security Groups</a>	80
<b><a href="#">Chapter 6 AWS Network Performance Optimizations</a></b>	83
<a href="#">Placement Groups</a>	83
<a href="#">Amazon Route 53</a>	86
<a href="#">Load Balancers</a>	89
<b><a href="#">Chapter 7 Security</a></b>	94
<a href="#">AWS Shared Security Model</a>	94
<a href="#">Principle of Least Privilege</a>	96
<a href="#">Industry Compliance</a>	97
<a href="#">Identity and Access Management</a>	97
<a href="#">Identity Federations</a>	102
<a href="#">Creating IAM Policies</a>	107
<a href="#">Further Securing IAM with Multifactor Authentication</a>	111
<a href="#">Multi-Account Strategies</a>	112
<a href="#">Preventing Distributed Denial of Service Attacks</a>	114
<a href="#">Amazon Web Application Firewall (WAF)</a>	115
<a href="#">AWS Shield</a>	116
<a href="#">AWS Service Catalog</a>	117
<a href="#">AWS Systems Manager Parameter Store</a>	118
<b><a href="#">Chapter 8 AWS Applications and Services</a></b>	120
<a href="#">AWS Simple Queueing Service (SQS)</a>	120
<a href="#">AWS Simple Notification Service (SNS)</a>	124
<a href="#">AWS Simple Workflow Service (SWF)</a>	126
<a href="#">AWS Kinesis</a>	127
<a href="#">AWS Elastic Container Service (ECS)</a>	132
<a href="#">AWS Elastic Kubernetes Service (EKS)</a>	135
<a href="#">AWS Elastic Beanstalk</a>	136
<a href="#">AWS CloudWatch</a>	137
<a href="#">AWS Config</a>	138
<a href="#">AWS CloudTrail</a>	140
<a href="#">AWS CloudFront</a>	141
<a href="#">AWS Lambda</a>	144
<a href="#">AWS Lambda@Edge</a>	146
<a href="#">AWS CloudFormation</a>	147
<a href="#">AWS Certificate Manager (ACM)</a>	148

<b><u>Chapter 9 Cost Optimization</u></b>	150
<a href="#">Financial Differences Between Traditional Data Centers and Cloud Computing</a>	150
<a href="#">Optimizing Technology Costs on the AWS Cloud</a>	150
<a href="#">AWS Budgets</a>	153
<a href="#">AWS Trusted Advisor</a>	154
<b><u>Chapter 10 Building High Availability Architectures</u></b>	156
<a href="#">What is Availability?</a>	156
<a href="#">Building a High Availability Network</a>	156
<b><u>Chapter 11 Passing the AWS Exam</u></b>	160
<b><u>Chapter 12 Practice Exam</u></b>	162

## **Chapter 1**

### **Introduction to Cloud Computing**

#### **The History of the Enterprise Network and Data Center**

Ever since computing resources provided a competitive advantage, organizations have been investing in their computing resources. Over time technology became not only a competitive advantage but a necessary resource to be competitive in today's business environment. Since technology can bring extreme advances in productivity, sales, marketing, communication, and collaboration, organizations invested more and more into technology. Organizations, therefore, built large and complex data centers, and connected those data centers to an organization's users with specialized networking, security, and computing hardware resources. The enterprise data center became huge networks, often requiring thousands of square feet of space, incredible amounts of power, cooling, hundreds, if not thousands, of servers, switches, routers, and many other technologies. Effectively, the enterprise and especially the global enterprise environments became massive networks—just like a cloud computing environment. The net result was a powerful private cloud environment.

Global enterprise data centers and high-speed networks work well. However, these networks come with a high cost and high level of expertise to manage these environments. Network and data center technology is not simple, and it requires a significant staff of expensive employees to design, operate, maintain, and fix these environments. Some large enterprise technology environments take billions of dollars to create and operate.

Global enterprise networks and data centers still have merit for high security, ultrahigh-performance environments requiring millisecond-level latency, and ultrahigh network and computing performance. An example environment that benefits from this traditional model are global financial environments, where shaving milliseconds off network, server, and application performance can equate to a significant competitive advantage. However, for many customers, the costs of procuring and operating the equipment are just too costly.

Recent advances in network, virtualization, and processing power make the transition to the cloud computing feasible.

#### **Why Now Is the Optimal Time for Cloud Computing**

Let's first look at virtualization, as it is the key enabling technology of cloud computing. Server performance has increased dramatically in recent years. It is no longer necessary to have a server or multiple servers for every application. Since today's servers are so powerful, they can

be partitioned into multiple logical servers in a physical server, reducing the need for so many servers. This reduces space, power, and cooling requirements. Additionally, virtualization makes it simple to move, add, or change server environments. Previously, any time you wanted to make a server change, you had to buy a new server, which could take weeks or months, install the operating system and all the applications dependencies, and then find a time to upgrade the server when it wasn't being used. This process was lengthy, as changes to any part of an IT environment can affect many other systems and users. With virtualization, to upgrade a server you have to copy only the server file to another machine, and it's up and running.

Server virtualization became so critical in improving hardware resource utilization for computing, that soon organizations explored moving the network to virtualized servers. Now routing, firewalling, and many other functions can be shifted to virtualized services with software-defined networking. For several years organizations have migrated their traditional datacenters to virtualized enterprise data centers, and it has worked well. However, network speed (bandwidth) has made such significant gains, while the cost of this high-performance networking has decreased substantially. Therefore, it is now possible to move the data center to a cloud computing environment and still achieve high performance with lower total costs. With the ability to purchase multiple 10-gigabit-per-second links to AWS, it's now feasible to connect an organization to a cloud provider at almost the same speed as if the application is in the local data center, but with the benefits of a cloud computing environment.

## **Hybrid Cloud**

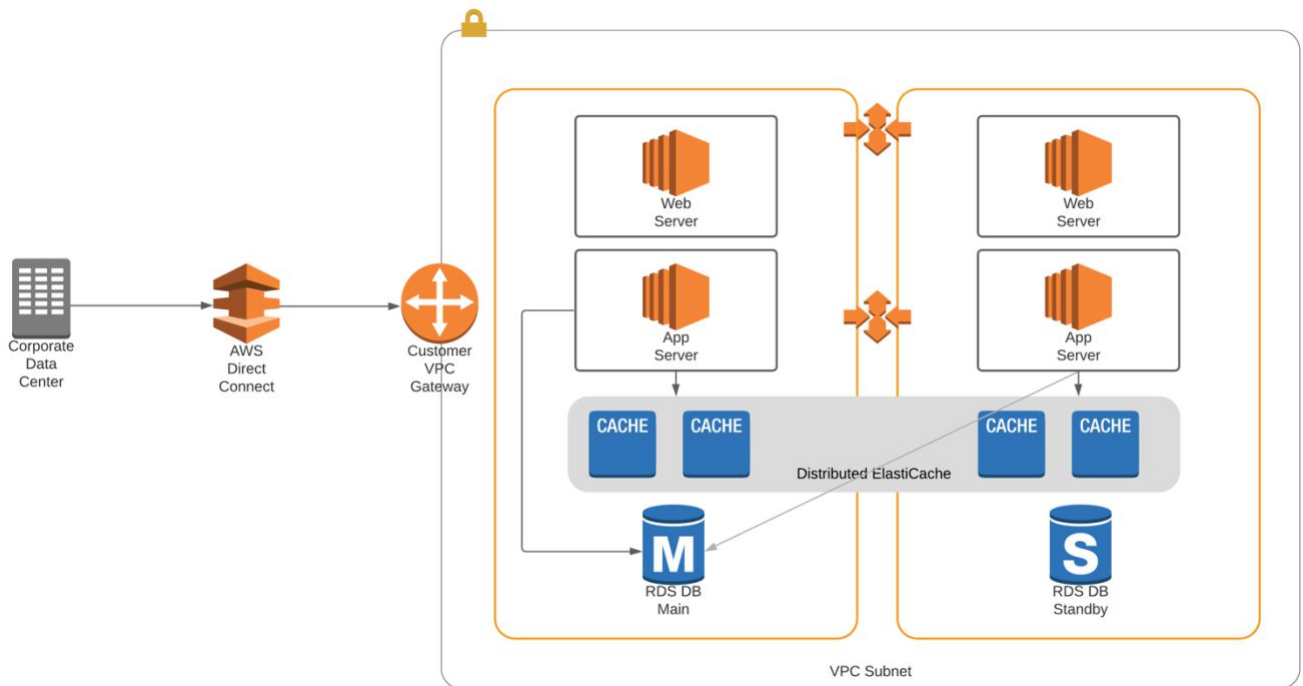
A hybrid cloud combines a standard data center with outsourced cloud computing. For many organizations, the hybrid cloud is the perfect migration to the cloud. In a hybrid architecture, the organization can run its applications and systems in its local data center and offload part of the computing to the cloud. This provides an opportunity for the organization to leverage its investment in its current technology while moving to the cloud. Hybrid clouds provide an opportunity to learn and develop the optimal cloud or hybrid architecture.

Applications for hybrid cloud include:

- Disaster recovery – Run the organization computing locally, with a backup data center in the cloud.
- On-demand capacity – Prepare for spikes in application traffic by routing extra traffic to the cloud.
- High performance – Some applications benefit from the reduced latency and higher network capacity available on-premises, and all other applications can be migrated to the cloud to reduce costs and increase flexibility.
- Specialized workloads – Move certain workflows to the cloud that require substantial development time, i.e., machine learning, rendering, transcoding.
- Backup – The cloud provides an excellent means to back up to a remote and secure location.



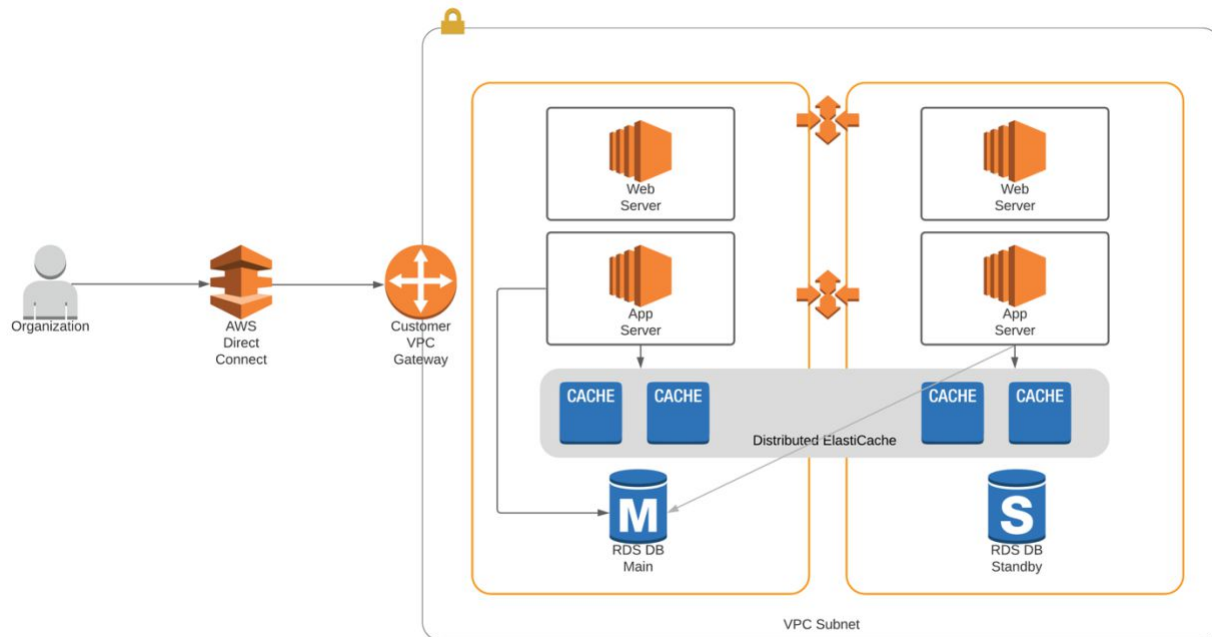
The diagram below shows an example of a hybrid cloud computing environment.



### Pure Cloud Computing Environment

In a pure cloud computing environment, all computing resources are in the cloud. This means servers, storage, applications, databases, and load balancers are all in the cloud. The organization is connected to the cloud with a direct connection or a VPN connection. The speed and reliability of the connection to the cloud provider will be the key determinant of the performance of this environment.

The diagram below shows an example of a pure cloud computing environment on the AWS platform.



A pure cloud computing environment has several advantages:

Scalability – The cloud provides incredible scalability.

Agility – Adding computing resources can occur in minutes versus weeks in a traditional environment.

Pay-as-you-go pricing – Instead of purchasing equipment for maximum capacity, which may be idle 90 percent of the time, exactly what is needed is purchased when needed. This can provide tremendous savings.

Professional management – Managing data centers is very complicated. Space, power, cooling, server management, database design, and many other components can easily overwhelm most IT organizations. With the cloud, most of these are managed for you by highly skilled individuals, which reduces the risk of configuration mistakes, security problems, and outages.

Self-healing – Cloud computing can be set up with health checks that can remediate problems before they have a significant effect on users.

Enhanced security – Most cloud organizations provide a highly secure environment. Most enterprises would not have access to this level of security due to the costs of the technology and the individuals to manage it.

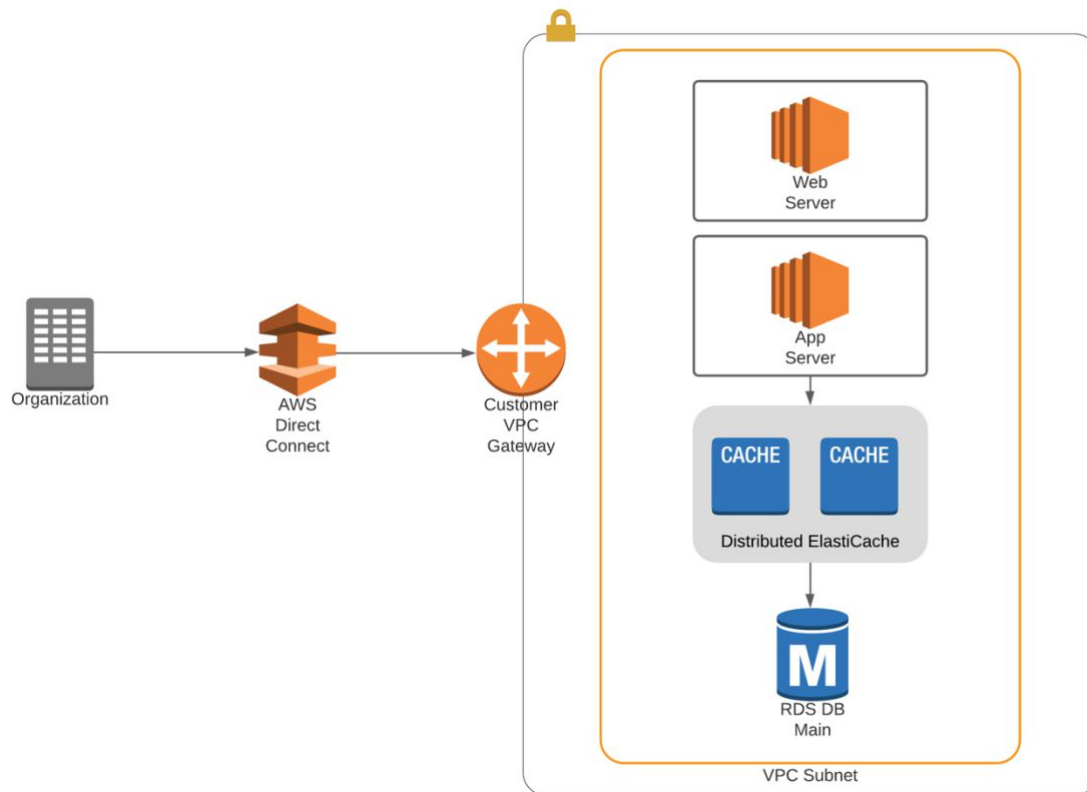
### **Connections to the Cloud**

If an organization moves its computing environment to the cloud, then the connection to the cloud becomes critical. If the connection to the cloud fails, then the organization can no longer access cloud resources. The performance needs and an organization's dependency on IT will determine the connection requirements to the cloud.

For most organizations, getting a "direct" connection to the cloud will be the preferred method. A direct connection is analogous to a private line in the networking world because it is effectively a wire that connects the organization to the cloud. This means guaranteed performance, bandwidth, and latency. As long as the connection is available, performance is excellent. This is unlike a VPN connection over the internet, where congestion anywhere on the internet can negatively affect performance.

Since network connections can fail, a direct connection is generally combined with a VPN backup over the internet. A VPN can send the data securely over the internet to AWS. A VPN provides data security via encryption and permits the transfer of routing information and the use of private address space. VPNs work by creating an IP security (IPsec) tunnel over the internet.

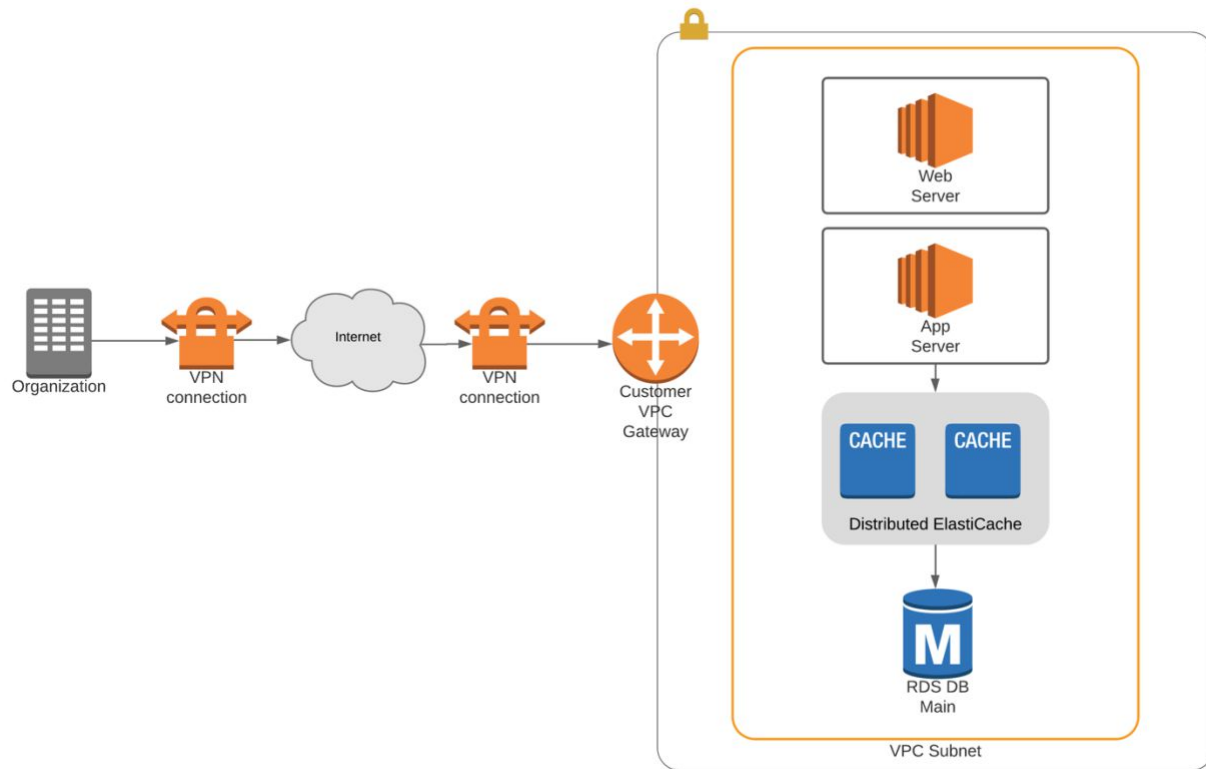
The diagram below shows an example of a direct connection to the AWS platform.



### VPN Connection to AWS

The simplest and cheapest means to connect to AWS is a VPN. A VPN provides a means to “tunnel” traffic over the internet in a secure manner. Encryption is provided by IPsec, which provides a means to provide encryption (privacy), authentication (identifying of the user), data authenticity (meaning the data has not been changed), and non-repudiation (meaning, the user can’t say they didn’t send the message after the fact). However, the problem with VPN connections is that while the connection speed to the internet is guaranteed, there is no control of what happens on the internet. So, there can be substantial performance degradation based upon the availability, routing, and congestion on the internet. VPN-only connections are ideal for remote workers and small branches of a few workers, where if they lose connectivity, there will not be significant costs to the organization.

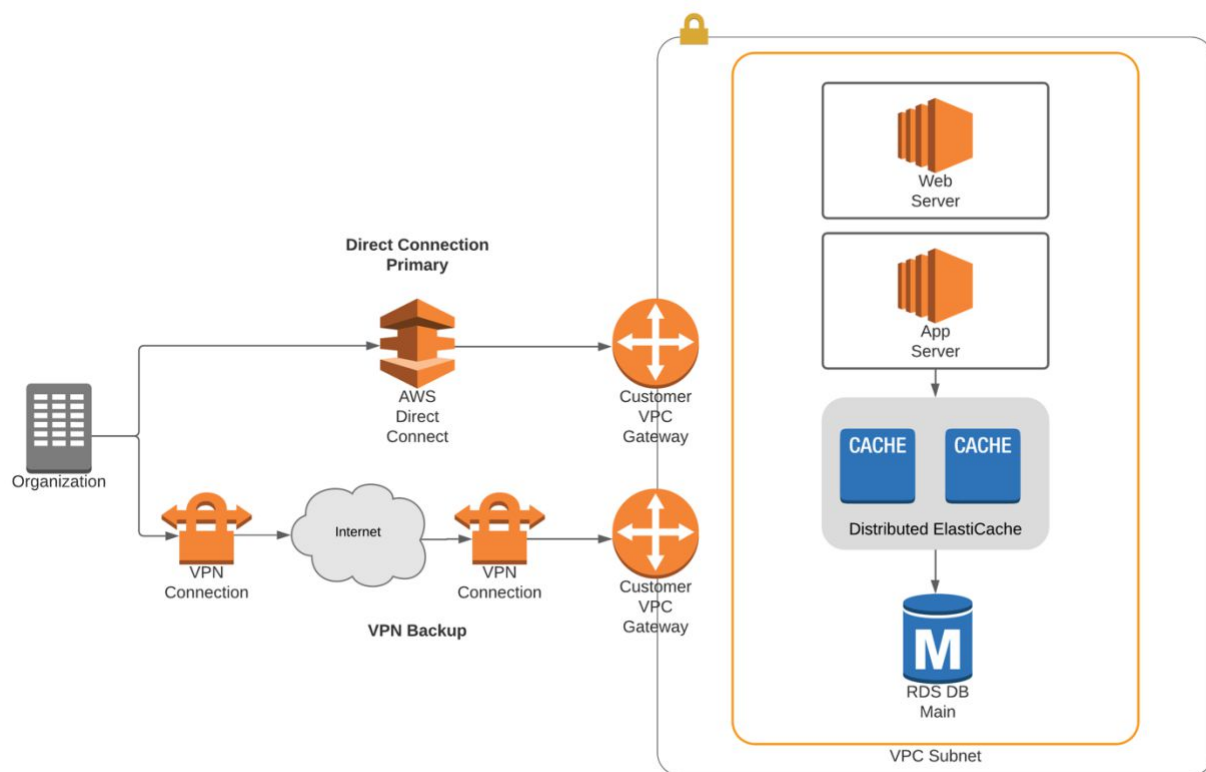
The diagram below shows an example of a VPN connecting to the AWS platform.



## High-Availability Connections

Connecting to the cloud with high availability is essential when an organization depends upon technology.

The highest availability architectures will include at least two direct connections to the cloud. Ideally, each connection is with a separate service provider, a dedicated router, and each router connected to different power sources. This configuration provides redundancy to the network connection, power failures, and the routers connecting to the cloud. For organizations that need 99.999 percent availability, this type of configuration is essential. For even higher availability, there can be a VPN connection as a backup to the other direct connections.



## High Availability at Lower Costs

A lower cost means to achieve high availability is to have a dedicated connection and a VPN backup to the cloud. This will work well for most organizations, assuming they can tolerate reduced performance when using the backup environment.

## **Basic Architecture of the AWS Cloud**

The AWS cloud comprises multiple regions connected with the Amazon high-speed network backbone. Many regions exist, but to understand the topology, here is a simplified explanation. Think of a region as a substantial geographic space (a significant percentage of a continent). Now, each region has numerous data centers. Each data center within a region is classified as an availability zone. For high-availability purposes and to avoid single points of failure, applications and servers can be in multiple availability zones. Large global organizations can optimize performance and availability by connecting to multiple regions with multiple availability zones per region.

## **How to Access and Manage Resources in the Cloud**

There are three ways to manage cloud computing resources on AWS. The methods to configure AWS cloud computing resources are the AWS Management Console, the Command Line Interface, and connecting via an API through the software development kit.

### **AWS Management Console**

The AWS Management Console is a simple-to-use, browser-based method to manage configurations. There are numerous options, with guidance and help functions. Most users will use the management console to set things up and perform basic system administration, while performing other system administration functions with the command line interface. You can access the management console at this URL. <https://aws.amazon.com/console/>

### **AWS CLI**

The AWS Command Line Interface enables you to manage computing resources via Linux commands and JavaScript Object Notation (JSON) scripting. This is efficient but requires more knowledge, training, and organizations need to know exactly what is needed and how to configure properly. With the CLI, there is no guidance as in the AWS Management Console. You will access the CLI by using secure shell. If you're using a Linux, Unix, or MacOS computer you can access the secure shell from a terminal window. If your using windows, you will need to install an SSH application, one popular application is putty. You can download Putty at no cost from this URL. <https://www.putty.org>

### **AWS SDK**

The AWS Software Development Kit (SDK) provides a highly effective method to modify and provision AWS resources on demand. This can be automated and can provide the ultimate

scalability. This will require sophisticated knowledge to build the programming resources and is recommended for experts.



## **Chapter 2**

### **Storage Options on the AWS Cloud Platform**

There are several storage options available to AWS cloud computing customers. They are: AWS Simple Storage Service (S3), Elastic Block Storage, Elastic File System, Storage Gateways, and WorkDocs.

In traditional data centers, there are two kinds of storage—block storage and file storage. Block storage is used to store data files on storage area networks (SANs) or cloud-based storage environments. It is excellent for computing situations where there is a need for fast, efficient, and reliable data transportation. File storage is stored on local systems, servers, or network file systems.

#### **AWS Primary Storage Options**

In the AWS cloud environment, there are three types of storage: block storage, object storage, and file storage.

##### **Block Storage**

Block storage places data into blocks and then stores those blocks as separate pieces. Each block has a unique identifier. This type of storage places those blocks of data wherever it is most efficient. This enables incredible scalability and works well with numerous operating systems.

##### **Object Storage**

Object-based storage differs from block storage. Object storage breaks data files into pieces called objects. It then stores those objects in a single place that can be used by multiple systems that have network access to the storage. Since each object will have a unique ID, it's easy for computing resources to access data on file-based storage. Additionally, each object has metadata or information about the data to make it easier to find when needed.

##### **File Storage**

File storage is traditional storage. It can be used for a systems operating system and network file systems. Examples of this are NTFS-based volumes for Windows systems and NFS volumes for Linux/UNIX systems. These volumes can be mounted and directly accessed by numerous computing resources simultaneously.

## **AWS Object Storage**

The AWS platform provides an efficient platform for block storage with Amazon Simple Storage Service, otherwise known as S3.

### **Amazon Simple Storage Service (S3)**

Amazon S3 provides high-security, high-availability, durable, and scalable object-based storage. S3 has 99.999999999 percent durability and 99.99 percent availability. Durability refers to a file getting lost or deleted. Availability is the ability to access the system when you need access to your data. So, this means data stored on S3 is highly likely to be available when you need it.

Since S3 is object-based storage, it provides a perfect opportunity to store files, backups, and even static website hosting. Computing systems cannot boot from object-based storage or block-based storage. Therefore, S3 cannot be used for the computing platforms operating system. Since block-based storage is effectively decoupled from the server's operating system, it has near limitless storage capabilities.<sup>1</sup>

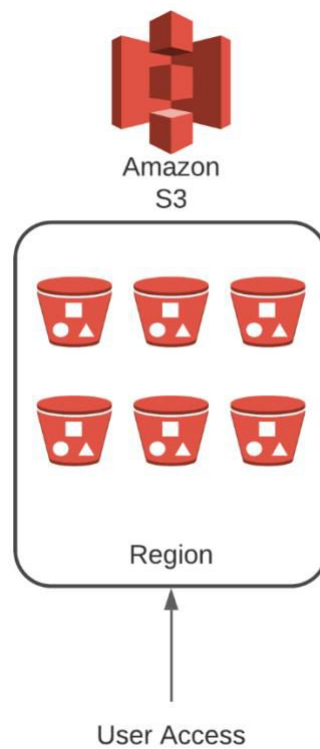
S3 is typically used for these applications:

- Backup and archival for an organization's data.
- Static website hosting.
- Distribution of content, media, or software.
- Disaster recovery planning.
- Big data analytics.
- Internet application hosting.

Amazon S3 is organized into buckets. Each bucket is a container for files stored on S3. Buckets create a top-level name space, meaning they must be globally unique and can be accessed by their DNS name. An example of this address can be seen below:

<http://mybucketname.s3.amazonaws.com/file.html>

The diagram below shows the organizational structure of S3.



Since the buckets use DNS-type addressing, it is best to use names that follow standard DNS naming conventions. Bucket names can have up to sixty-three characters including letters, numbers, hyphens, and periods. It's noteworthy that the path you use to access the files on S3 is not necessarily the location to where the file is stored. The URL used to access your file is really a pointer to the database where your files are stored. S3 functions a lot like a database behind the scenes, which enables you to do incredible things with data stored on S3 like SQL queries. An organization can have up to 100 buckets per account without requesting a bucket limit increase from AWS.

Buckets are placed in different geographic regions. When creating the bucket, to achieve the highest performance, place the bucket in a region that is closest. Additionally, this will decrease data transfer charges across the AWS network. For global enterprises, it may be necessary to place buckets in multiple geographic regions. AWS S3 provides a means for buckets to be replicated automatically between regions. This is called cross-region replication.

### **S3 Is Object-Based Storage**

S3 is used with many services within the AWS cloud platform. Files stored in S3 are called objects. AWS allows most objects to be stored in S3. Every object stored in an S3 bucket has a

unique identifier, also known as a key. A key can be up to 1,024 bytes, which are comprised of Unicode characters and can include slashes, backslashes, dots, and dashes.

Single files can be as small as zero bytes, all the way to 5 TB per file. This provides ultimate flexibility. Objects in S3 can have metadata (information about the data), which can make S3 extremely flexible and can assist with searching for data. The metadata is stored in a name-value pair environment, much like a database. Since metadata is placed in a name-value pair, S3 provides a SQL-based method to perform SQL-based queries of your data stored in S3. To promote scalability, AWS S3 uses an eventually consistent system. While this promotes scalability, it is possible that after you delete an object, it may be available for a short period.

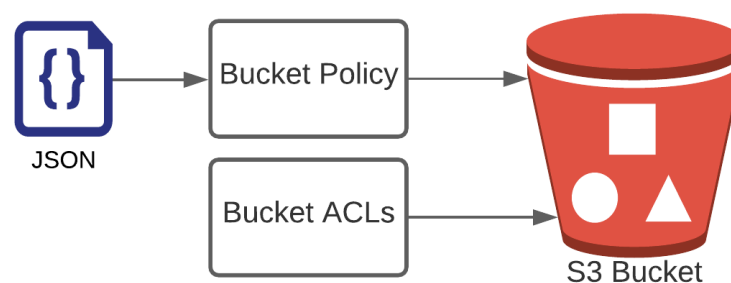
### Securing Data in S3

The security of your data is paramount, and S3 storage is no exception. S3 is secured via the buckets policy and user policies. Both methods are written in JSON access-based policy language.

Bucket policies are generally the preferred method to secure your data in S3. Bucket policies allow for granular control of the security of your digital assets stored in S3. Bucket policies are based on IAM users and roles. S3 bucket policies are similar to the way Microsoft authenticates users and grants access to certain resources based upon the user's role, permissions, and groups in active directory. <sup>2</sup>

S3 also allows for using ACLs to secure your data. However, the control afforded to your data via an ACL is much less sophisticated than IAM policies. ACL-based permissions are essentially read, write, or full control.

The diagram below shows how ACL and bucket policies are used with AWS S3.



## **S3 Storage Tiers**

AWS S3 offers numerous storage classes to best suit an organization's availability and financial requirements. The main storage classes can be seen below:

- Amazon S3 Standard
- Amazon S3 Infrequent Access (Standard-IA)
- Amazon S3 Infrequent Access (Standard-IA) – One Zone
- Amazon S3 Intelligent-Tiering
- Amazon S3 Glacier

### **Amazon S3 Standard**

Amazon S3 Standard provides high-availability, high-durability, high-performance, and low-latency object storage. Given the performance of S3, it is well suited for storage of frequently accessed data. For most general-purpose storage requirements, S3 is an excellent choice.

### **Amazon S3 Infrequent Access (Standard-IA)**

Amazon S3 Infrequent Access provides the same high-availability, high-durability, high-performance, and low-latency object storage as standard S3. The major difference is the cost, which is substantially lower. However, with S3 Infrequent Access, you pay a retrieval fee every time data is accessed. This makes S3 extremely cost-effective for long-term storage of data not frequently accessed. However, access fees might make it cost-prohibitive for frequently accessed data.

### **Amazon S3 Infrequent Access (Standard-IA) – One Zone**

Amazon S3 Infrequent Access provides the same service as Amazon S3-IA but with reduced durability. This is great for backups of storage due to its low cost.

### **Amazon S3 Intelligent Tiering**

Amazon S3 Intelligent Tiering provides an excellent blend of S3 and S3-IA. Amazon keeps frequently accessed data on S3 standard and effectively moves your infrequently accessed data to S3-IA. So, you have the best and most cost-effective access to your data.

### **Amazon S3 Glacier**

Amazon Glacier offers secure, reliable, and very low-cost storage for data that does not require instant access. This storage class is perfect for data archival or backups. Access to data must be requested; after three to five hours the data becomes available. Data can be accessed sooner by paying for expedited retrievals. Glacier also has a feature called vault lock. Vault lock can be used for archives that require compliance controls, i.e., medical records. With vault lock, data

cannot be modified but can be read when needed. Glacier, therefore, provides immutable data access, meaning it can't be changed while in Glacier.

## Managing Data in S3

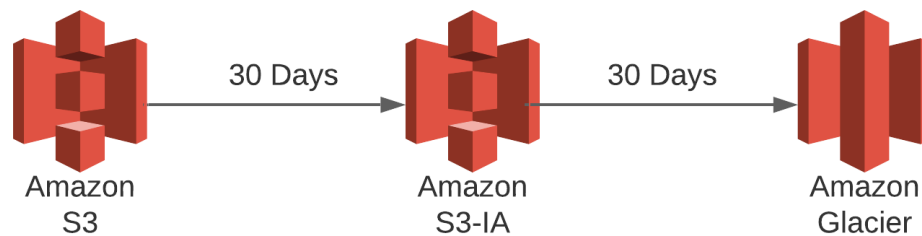
This next section describes how to manage and protect your data on S3.

### S3 Lifecycle Management

The storage tiers provided by S3 provide an excellent means to have robust access to data with a variety of pricing models. S3 lifecycle management provides an effective means to automatically transition data to the best S3 tier for an organization's storage needs.

For example, let's say you need access to your data every day for thirty days, then infrequently access your data for the next thirty days, and you may never access your data again but want to maintain for archival purposes. You can set up a lifecycle policy to automatically move your data to the optimal location.

You can store your data in S3, then after thirty days, have your data automatically moved to S3-IA, and after thirty days, the data can be moved to Glacier for archival purposes. This can be seen in the diagram below.



That's the power of the lifecycle policies.

Lifecycle policies can be configured to be attached to the bucket or specific objects as specified by an S3 prefix.

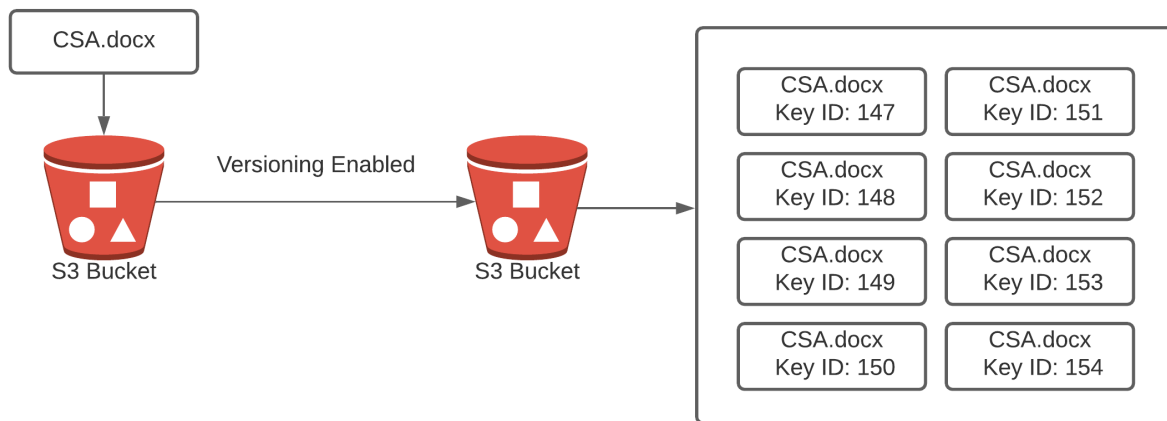
### S3 Versioning

To help secure S3 data from accidental deletion, S3 versioning is the AWS solution.

Amazon S3 versioning protects data against accidental or malicious deletion by keeping multiple versions of each object in the bucket, identified by a unique version ID. Therefore, multiple copies of objects are stored in S3 when versioning is enabled. Every time there is a change to the object, S3 will store another version of that object. Versioning allows users to

preserve, retrieve, and restore every version of every object stored in their Amazon S3 bucket. If a user makes an accidental change or even maliciously deletes an object in your S3 bucket, you can restore the object to its original state simply by referencing the version ID besides the bucket and object key. Versioning is turned on at the bucket level. Once enabled, versioning cannot be removed from a bucket; it can be suspended only.

The diagram below shows how S3 versioning maintains a copy of all previous versions by using a Key ID.



### Multifactor Authentication Delete

To provide additional protection from data deletion, S3 supports multifactor authentication to delete an object from S3. When an attempt to delete an object from S3 is made, S3 will request an authentication code. The authentication code will be a one-time password that changes every few seconds. This one-time authentication code can be provided from a hardware key generator or a software-based authentication solution, i.e., Google Authenticator.

### Organizing Data in S3

As previously discussed, S3 storage is very similar to a database. Essentially the data is stored as a flat arrangement in a bucket. While this scales well, it is not necessarily the most organized structure for end users. S3 allows the user to specify a prefix and delimiter parameter so the user can organize their data in what feels like a folder. Essentially the user could use a slash / or backslash \ as a delimiter. This will make S3 storage look and feel like a traditional Windows or Linux file system organized by folders.<sup>3</sup> For example:

- mike/2020/awsvideos/storage/S3.mp4
- mike/2020/awsvideos/compute/ec2.mp4
- mike/2020/awsvideos/database/dynamo.mp4

## **Encrypting Your Data**

S3 supports a variety of encryption methods to enhance the security of your data. Generally, all data containing sensitive or organization proprietary data should be encrypted. Ideally, you encrypt your data on the way to S3, as well as when the data is stored (or resting) on S3. A simple way to encrypt data on the way to S3 is to use https, which uses SSL to encrypt your data on its way to the S3 bucket. Encrypting data on S3 can be performed using client-side encryption or server-side encryption.<sup>4,5</sup>

### **Client-Side Encryption**

Client-side encryption means encrypting the data files prior to sending to AWS. This means the files are already encrypted when transferred to S3 and will stay encrypted when stored on S3. To encrypt files using client-side encryption, there are two options available to use. Files can be encrypted with a client-side master key or a client master key using the AWS key management system (KMS). When using client-side encryption, you maintain total control of the encryption process, including the encryption keys.

### **Server-Side Encryption**

Alternatively, S3 supports server-side encryption. Server-side encryption is performed using S3 and KMS. Amazon S3 automatically will encrypt when storing and decrypt when accessing your data on S3. There are several methods to perform server-side encryption and key management. These options are discussed below.

#### **SSE-KMS (Customer-Managed Keys with AWS KMS)**

The SSE-KMS is a complete key management solution. The user manages their own master key, but the key management system manages the data key. This solution provides extra security by having separate permissions for using a customer-managed key, which provides added protection against unauthorized access of your data in Amazon S3. Additionally, SSE-KMS helps with auditing by providing a trail of how, who and when your data was accessed.

#### **SSE-S3 (AWS-Managed Keys)**

SSE-S3 is a fully integrated encryption solution for data in your S3 bucket. AWS performs all key management and secure storage of your encryption keys. Using SSE-S3, every object is encrypted with a unique encryption key. All object keys are then encrypted by a separate master key. AWS automatically generates new encryption keys and automatically rotates them monthly.



## SSE-C (Customer-Provided Keys)

SSE-C is used in environments when you want total autonomy over key management. When using SSE-C, AWS S3 will perform all encryption and decryption of your data, but you have total control of your encryption keys.

## Tuning S3 for Your Needs

There are a few tuning options to make S3 perform even better. They are covered below.

### Presigned URLs

All objects stored in S3 are private by default. Meaning, only the owner has access to the objects in their bucket. Therefore, to share a file with another user or organization, you need to provide access to the object. You can generate a presigned URL to share an S3-based object with others. When you generate the presigned URL, the owner's encryption key is used to sign the URL, which allows the receiver temporary access to the object.

You can use the following authentication options to generate a presigned URL. Each method has a different expiration to the authorization provided by the presigned URL. These options can be seen in the table below:

Presigned URL Expiration	
Method	Expiration Time
IAM Instance Profile	Up to 6 Hours
AWS Security Token Service	Up to 36 Hours
IAM User	Up to 7 Days
Temporary Token	When the Token Expires

### Multipart Uploads

AWS S3 allows for objects of up to 5 GB in size to be stored. However, when trying to upload large files, many things can go wrong and interrupt the file transmission. It is a best practice to send files larger than 100 MB as a multipart upload. In a multipart upload, the file is broken into pieces, and each piece is sent to S3. When all the pieces are received, S3 puts the pieces back together into a single file. This helps dramatically if anything goes wrong with the transmission; only a part of the file needs to be re-sent instead of the whole file. This improves both the speed and reliability of transferring large files to S3.

## Range Gets

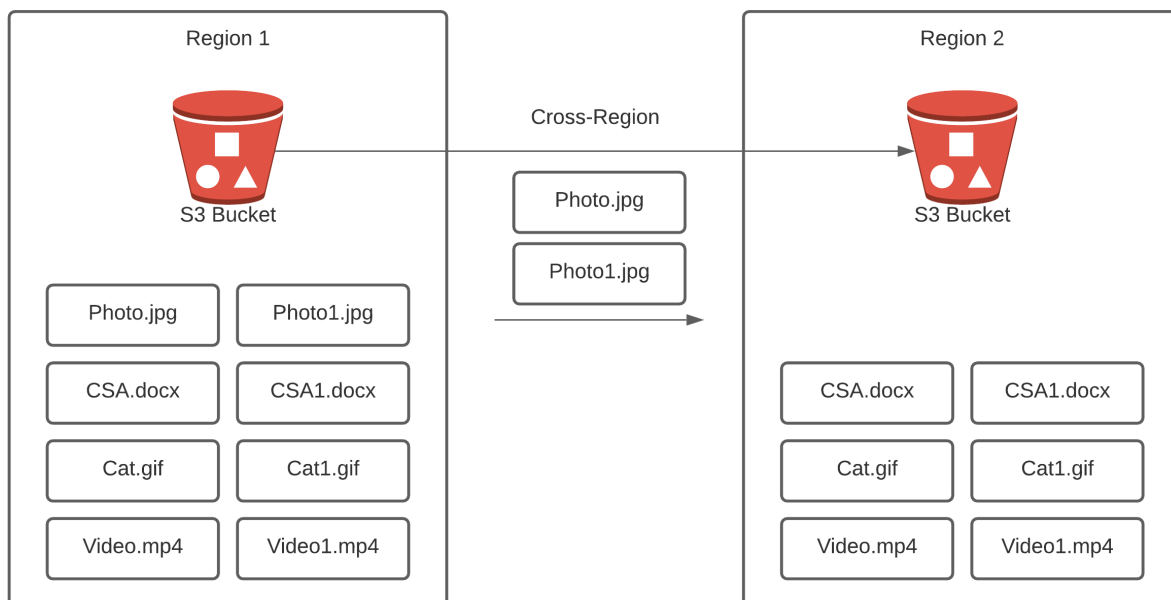
S3 supports large files. Sometimes you need access to some of the data in the file, but not the entire object. A range get allows you to request a portion of the file. This is highly useful when dealing with large files or when on a slow connection, i.e., mobile network.

## Cross-Region Replication

S3 cross-region replication automatically copies the objects in an S3 bucket to another region. Therefore, all S3 buckets will be synchronized. After cross-region replication is turned on, all new files will be copied to the region for which cross-region replication has been enabled. Objects in the bucket before turning on cross-region replication will need to be manually copied to the new bucket.

Cross-region replication is especially useful when hosting a global website's backend on S3. If website users experience latency due to the distance of the users to the website, the website can be hosted locally off a local S3 bucket. An example would be a US company with the website hosted in New York that suddenly gets a large customer base in Japan. The S3 bucket can be manually copied to a bucket in Asia. With cross-region replication enabled, any future changes to the website's main bucket in New York will be automatically copied to the S3 bucket in Asia. This can dramatically reduce latency and provide a disaster recovery backup of the main web site.

The diagram below shows cross-region replication copying files from one region to another region.



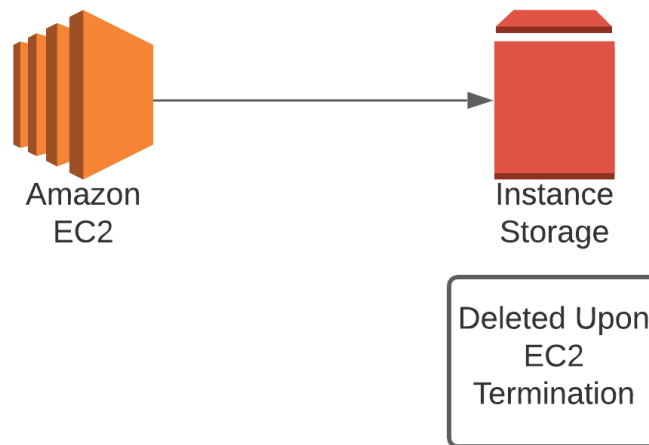
## Storage for Computing Resources

The next class of storage is storage directly available to computing platforms. These include instance storage, elastic block storage, and elastic file systems.

### Instance Storage

An instance storage is essentially a transient storage platform for an elastic computing instance. It is temporary in that when the instance is stopped, the volume is automatically deleted. This will be covered in much more depth when we discuss the EC2 platform.

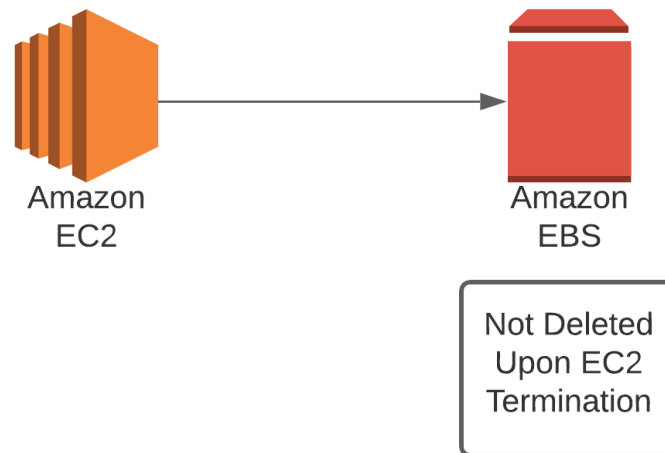
The diagram below shows how EC2 instance storage is used within the AWS platform. Note the instance storage is deleted upon EC2 termination.



### Elastic Block Storage (EBS)

Elastic block storage is a high-performance block storage platform for using EC2 instances and AWS databases. An EBS volume can be mounted and used by EC2 instances, and the data on an EBS volume is not deleted on an EC2 instance termination. EBS volumes are designed for high throughput and high-transaction workloads. This storage is ideal for databases, enterprise applications, containers and big data applications. EBS is scalable and can scale to multiple petabytes of data. EBS is highly available with the availability of 99.999 percent, so it's perfect for mission-critical use. EBS functions like a virtual hard drive. EBS is automatically backed up to another availability zone to protect against any device failure. <sup>6</sup>

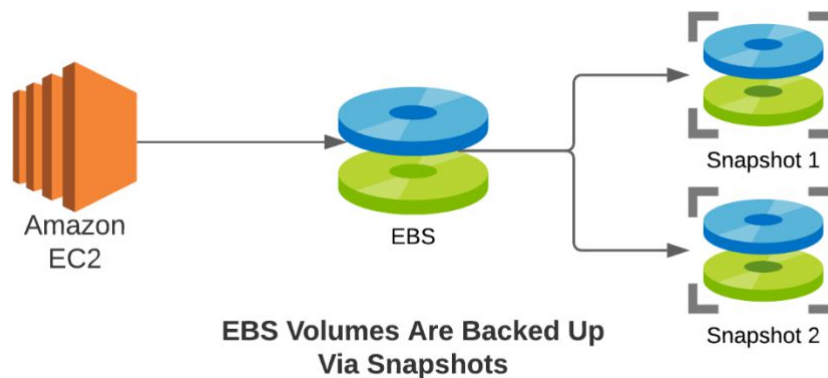
The diagram below shows how EBS instances are attached to an EC2 instance.



### Backing Up EBS Volumes

EBS volumes can be backed up to other regions in the form of EBS snapshots. An EBS snapshot is a point-in-time copy of your data. A snapshot can be shared with other users and can be copied to other regions. Snapshots can be versioned, and multiple versions can be maintained. Therefore, you can restore your data to any previous snapshots. EBS snapshots are incrementally backed up to optimize speed and storage space. Unlike traditional incremental backups, you can delete some backups and still make a complete restore of current data stored on EBS volumes.<sup>7</sup>

The diagram below shows how EBS volumes are backed up using snapshots.



## **EBS Volume Types**

There are several types of EBS volumes available. Choosing the right volume type can make a substantial difference in the performance of your system. This next section describes the EBS volumes available and how to choose the right EBS volume type to meet the systems requirements. The four types of EBS volumes are EBS Provisioned IOPS (io1), EBS General Purpose SSD (gp2), EBS Throughput Optimized HDD (st1), and EBS Cold HDD (sc1).

### **EBS Provisioned IOPS (io1)**

EBS-provisioned IOPS is the highest performance SSD storage available on the EBS platform. This platform enables the user to purchase guaranteed speed of read and write performance, in terms of inputs and outputs per second (IOPS). EBS-provisioned IOPS is optimal for applications that require high disk input and output performance (IO). It's designed for databases or other applications requiring low latency. The throughput speeds provided by EBS PIOPS volumes are up to 1,000 MB/second.

### **EBS General Purpose SSD (gp2)**

EBS general purpose SSD (gp2) is an SSD-based storage solution. It provides a good balance of price and performance. GP2 is an excellent platform for a boot volume, as it has good performance and does not get erased upon EC2 instance shutdown. GP2 is great for transactional workloads. It is great for dev and test environments that require low latency, at a much lower cost than PIOPS based volumes used in production environments.

### **EBS Throughput Optimized HDD (st1)**

EBS Throughput Optimized HDD (st1) is low-cost magnetic storage. St1 is designed for frequently accessed workloads. It supports a fairly significant throughput of 500 MB/second but with higher latency than with SSD-based volumes. St1 is designed for situations that require a low-cost option to store substantial data. It works well with applications that have sequential read and writes to the disk.

### **EBS Cold HDD (sc1)**

EBS Cold HDD (sc1) is the lowest cost option for EBS volumes. Sc1 is for workloads that do not require frequent disk access but still need a reliable storage platform.

## **Choosing the Correct EBS Volume Type**

Choosing the correct EBS volume type can make the difference between optimal performance and performance problems within the system. The primary determination of volume selection is determining whether the application requires high throughput, low latency, or both.

For applications requiring high throughput and low latency, choose EBS-provisioned IOPS. Applications that require moderate throughput, but low latency will perform well on EBS general purpose SSD volumes. For applications that require high throughput but are less sensitive to latency, EBS Throughput Optimized HDD are a great option, especially if the application performs sequential read and writes of data. Lastly, EBS Cold HDD is perfect for storing a large amount of infrequently accessed data that is not sensitive to higher latencies than SSD volumes.

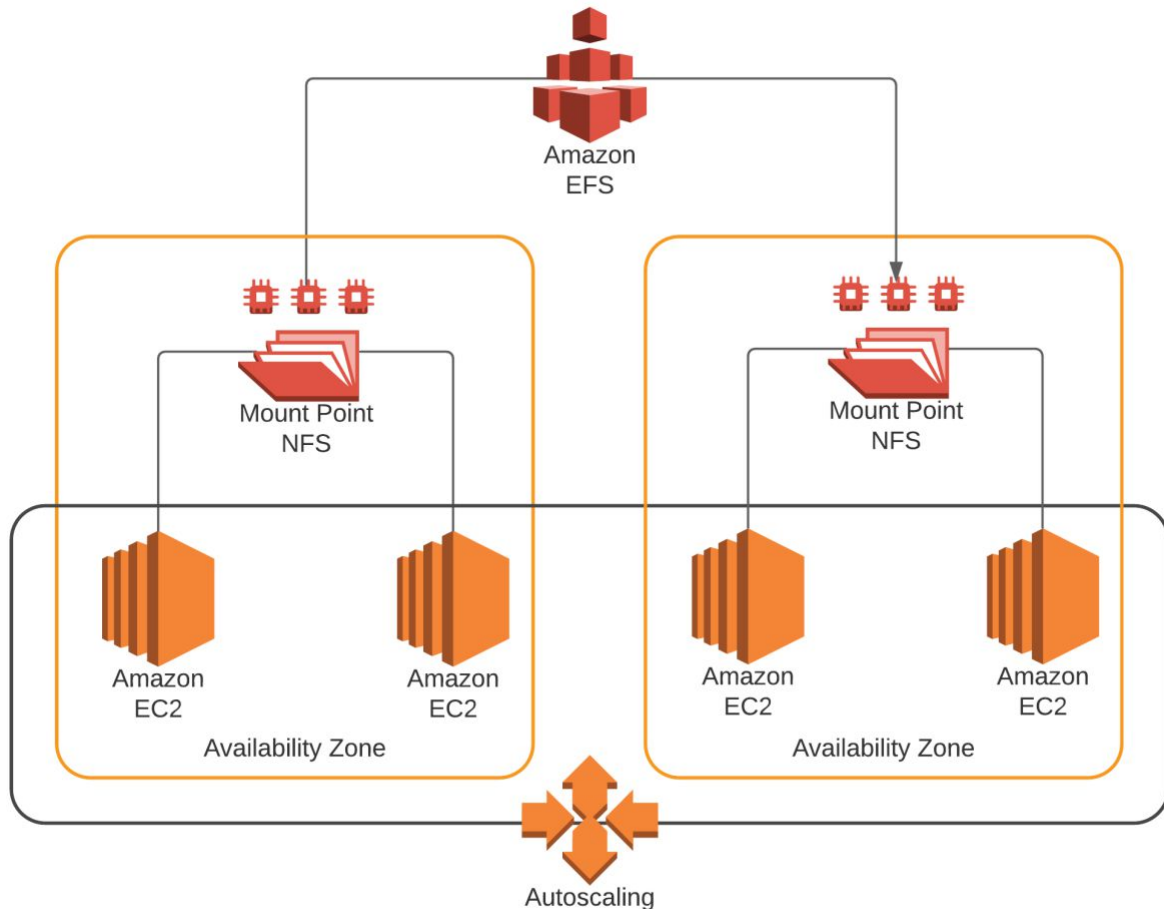
## **Elastic File System (EFS)**

The next storage for AWS is the Elastic File System (EFS). EFS is a high-performance, highly scalable file system for networked computers. It is essentially the AWS version of the network file system (NFS) originally invented by Sun Microsystems. Since EFS is a network file system, it can be accessed simultaneously by many computing instances. EFS is best used when a high-performance network file system is required. Think of corporate file shares or multiple servers needing access to the same files simultaneously.

There are two versions of EFS available. The two versions of EFS are standard and infrequent access. Standard EFS is the normal version and the highest performance option. EFS-Infrequent access is a lower-cost option for files not accessed frequently. EFS has numerous benefits:

- Scalable – High throughput, high IOPS, high capacity, and low latency.
- Elastic – EFS will automatically adjust sizing to meet the required storage capacity.
- Pricing – Pay for what is used.
- POSIX compatible – This enables access from standard file servers both on premises and on the cloud. Works with traditional NFS-based file permissions and directories.

The diagram below shows how EFS is mounted and used by multiple EC2 instances on the AWS platform.



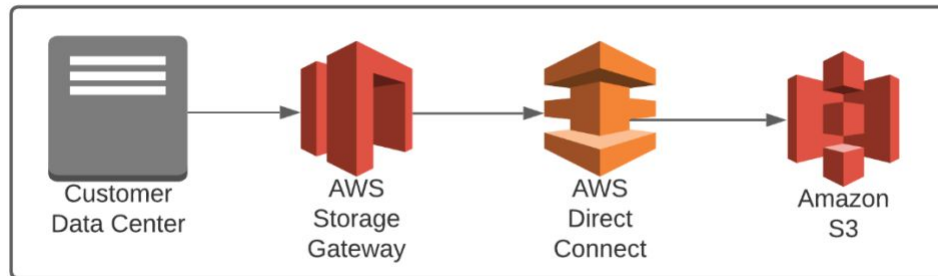
## AWS Storage Gateway

Connecting on-premises environments to the cloud and achieving optimal performance often requires some tuning. This is especially true when connecting servers to object-based storage on S3. A great option to connect the on-premises data center to S3-based storage is with a storage gateway. This makes the storage look and feel like a network file system to the on-premises computing systems.<sup>9</sup>

A storage gateway is a virtual machine that runs in the data center. It is a prebuilt virtual machine from AWS available in VMware, Hyper V, or KVM. This virtual machine acts as a virtual file server. The on-premises computing resources read and write to storage gateway, which then synchronizes to S3. You can access the storage gateway with SMB- or NFS-based shares. SMB is optimal for Windows devices, while NFS is optimal for Linux machines.

Three types of storage gateways are available for different applications. They are volume gateways, both cached volumes and stored volumes, and tape gateways.

The diagram below shows an example of a storage gateway on the AWS platform.



### **Storage Gateway Cached Volume**

In a volume gateway cached volume, the organization's data is stored on S3. The cache volume maintains frequently accessed data locally on the volume gateway. This provides low-latency file access for frequently accessed files to the on-premises systems.

### **Storage Gateway Stored Volume**

In a storage gateway stored volume, the on-premises systems store your files to the storage gateway, just like any other file server. The storage gateway then asynchronously backs up your data to S3. The data is backed up to S3 via point in time snapshots. Since the data is on S3 via a snapshot, these snapshots can be used by EC2 instances should something happen to the on-premises data center. This provides an excellent and inexpensive offsite disaster recovery option.

### **Tape Gateway**

The tape gateway provides a cloud backup solution. It essentially replaces backup tapes used by some enterprise environments for deep archival purposes. Since it is virtual, there is no need to maintain physical tapes and the infrastructure to support a tape-based backup solution. With the tape gateway, data is copied to Glacier or Deep Archive.



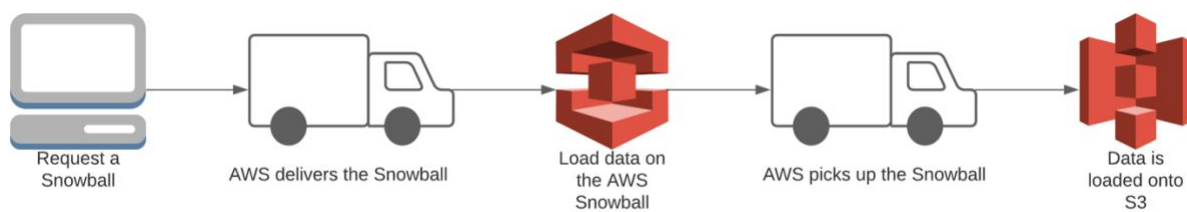
## Migrating Data to AWS

Moving an organization's data center to the cloud involves setting up the cloud infrastructure as well as moving an organization's data to AWS. Data can be copied to the cloud over a VPN or a direct connection. But, if there is a tremendous amount of data, or rapid timelines to move to the cloud, it might be necessary to move files to the cloud in a more efficient manner. Additionally, it might be cheaper to send data directly to S3, as opposed to adding additional high-performance, high-capacity direct connections to AWS. To that end, there are two methods to send data directly to AWS instead of using a network connection. These methods are the AWS Snowball and the AWS Import/Export service.

### AWS Snowball

The AWS Snowball is a rugged computer with substantial storage that can be rented from AWS. AWS ships the Snowball to the customer. The customer places the Snowball on their network and copies the files they want to move to AWS. The files on the Snowball are encrypted. The organization then ships the Snowball to AWS. Once the Snowball is received by AWS, AWS employees move the data from the Snowball to the organization's cloud computing environment.

The diagram below shows an example of an AWS Snowball.



### AWS Import/Export Service

The import-export service is essentially a means to ship data to AWS. Essentially, you copy your data to a hard drive or hard drives. The organization then ships the hard drives to AWS. Once the hard drives are received by AWS, they move the data from the hard drives to your cloud computing environment.<sup>10</sup>

The diagram below shows how the AWS Import/Export service is used to quickly move large amounts of data to AWS.



## Storage for Collaboration

In recent years there has been a significant move to cloud storage for collaboration. This is especially useful for working on creative projects, documents, or presentations. AWS has created Amazon WorkDocs for this purpose.

### Amazon WorkDocs

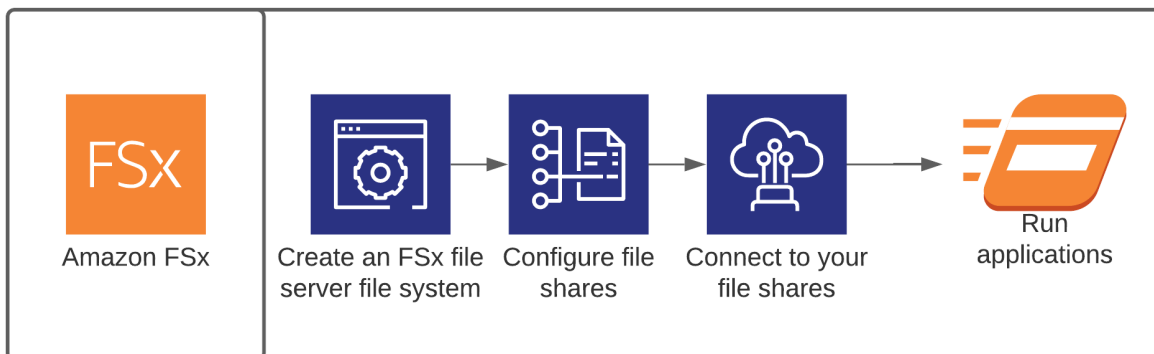
Amazon WorkDocs is a fully managed, secure content creation, storage, and collaboration service. It is similar in functionality to Google Drive or Dropbox. WorkDocs enables collaboration across projects and facilitates things like shared document editing. It is a simple solution with pay-as-you-go pricing. WorkDocs can be accessed with a web interface or with client software. Client software is available for Windows and macOS clients. This is secure storage and meets all main forms of compliance standards, i.e., HIPAA, PCI, and ISO.<sup>11</sup>

## Amazon FSx for Windows

Some organizations heavily utilize the Windows platform for many critical services. Organizations that need native windows file servers can use Amazon FSx. Amazon FSx are hosted Microsoft Windows file servers. FSx is a Windows server, therefore it supports Microsoft features such as storage quotas and active directory integration. As a Windows file server, it uses the Server Message Block (SMB) protocol. Windows-based SMB file shares can be accessed by Windows, macOS, and Linux hosts.

FSx is a high-availability service with high-availability single and multiple availability zone options. FSx provides data protection through encryption, both in transit and at rest. Additionally, to protect against data loss, FSx provides fully managed backups.

The diagram below shows an example of AWS FSx being used for Windows hosts.



## Labs

- 1) Create a budget to protect yourself from costly overages during your lab practice. Link on how to create a budget below.  
<https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/budgets-create.html>
- 2) Create an S3 bucket and upload several small files. Link on how to create an S3 bucket below.  
<https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-bucket.html>
- 3) Generate a presigned URL on one of the files you uploaded to the s3 bucket. Link on how to pre-sign a URL below.  
<https://docs.aws.amazon.com/cli/latest/reference/s3/presign.html>
- 4) Set up static website hosting. You will need a basic html page to upload to S3. You can easily save a word document as a webpage and upload to S3. Link on how to create static website hosting below.  
<https://docs.aws.amazon.com/AmazonS3/latest/user-guide/static-website-hosting.html>
- 5) Create an EC2 instance. Link on how to create EC2 instance below.  
[https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2\\_GetStarted.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)
- 6) Secure Shell (SSH) into EC2 instance. Link on how to SSH into EC2 instance below.  
<https://docs.aws.amazon.com/quickstarts/latest/vmlaunch/step-2-connect-to-instance.html>
- 7) Create an EBS Volume. Link on how to create an EBS volume below.  
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-volume.html>

- 8) Mount EBS volume to EC2 Instance. Link on how to mount EBS volume is below.  
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html>
- 9) Create an EFS Volume. Link on how to create EFS volume below.  
<https://docs.aws.amazon.com/efs/latest/ug/gs-step-two-create-efs-resources.html>
- 10) Shut down all services and instances when you are finished this section to avoid being charged for running AWS instances.

## **Summary**

AWS has a comprehensive storage platform for computing instances, backup, disaster recovery, and collaboration purposes. Now that you know the options available, you can choose the best option for your application's requirements.

## Chapter 3

### Computing on the AWS Platform (EC2)

This chapter describes the computing options available within the AWS Platform. AWS provides numerous computing options that can be sized based upon an organization's needs. Sizing compute resources are similar to sizing virtual machines in a traditional data center. Servers should be configured based upon the CPU, memory, storage (size and performance), graphics processing unit (GPU), and network performance requirements.<sup>12</sup>

AWS primary computing platform is called Elastic Compute Cloud (EC2). EC2 servers are virtual machines launched on the AWS platform. EC2 servers are called instances. EC2 instances are sized like any virtualized server environment. AWS has numerous options to meet an organization's needs. Instance types should be chosen based upon the need for the following computing options:

- CPU cores (virtual CPUs)
- Memory (DRAM)
- Storage (capacity and performance)
- Network performance

Instances are available in a wide range of sizes and configurations to provide a means to perfectly size a system based upon an organization's needs. AWS has a multitude of possible server configurations to assist the organization with properly sizing their computing instances.<sup>13</sup>

A summary of the EC2 instance types is below:

EC2 Instance Types	Specialty	Use Case
A1	Arm-based Workloads	Web Servers
C5	Compute Optimized	Batch Processing, Media Transcoding
G3	GPU Based Workloads	Machine Learning
I3	High Speed Storage	Data Warehousing, High Performance Databases
M5	General Purpose	Databases
M6	General Purpose	Application Servers, Gaming Servers
R5	Memory Optimized	Caches, High Performance Databases
T3	Burstable Computing Platform	Web apps, Test Environments
X1	Lowest Pricing Per GB DRAM	Bid Data Processing Engines, In-Memory Databases

Every instance type can be ordered in various sizes. Check with AWS for sizing, as sizes are subject to change.

EC2 instances support Windows and Linux operating systems. An organization can build its own virtual machines, or can start with a prebuilt virtual machine. A prebuilt virtual machine is called an Amazon Machine Image (AMI). It is important to note that AMIs will need a storage volume for the instance. The storage volume may be an instance volume or an EBS volume. It is

essential to choose the correct storage volume type. Standard instance volumes will be deleted upon instance reboot or termination. EBS volumes are persistent, and data will be available after reboot or instance termination.

### **Amazon Machine Images (AMI)**

Since the AMI is the basis for the virtual machine, it's important to choose the correct AMI. AMIs can be obtained from the following sources:

- Published by AWS – These images are prebuilt by Amazon for a variety of needs. They are available in a variety of operating systems.
- AWS Marketplace – These are prebuilt machines by AWS partners. These machines are generally created for specific uses (i.e., licensed software from a third party).
- AMIs from existing instances – This is generally a customer-created AMI. It's created from an existing server. This enables a full machine copy, with all applications and package dependencies installed.
- AMIs from uploaded servers – An AMI can be created from a physical server to virtual machine conversion. Additionally, an AMI can be imported from a virtual machine conversion (i.e., VMware or VirtualBox virtual machine).

A complete AMI has several components:

- Operating system from one of the four AMI sources.
- Launch permissions for the instance.
- A block device mapping of storage volumes in the system.

### **Automating the First Boot for EC2 Instances**

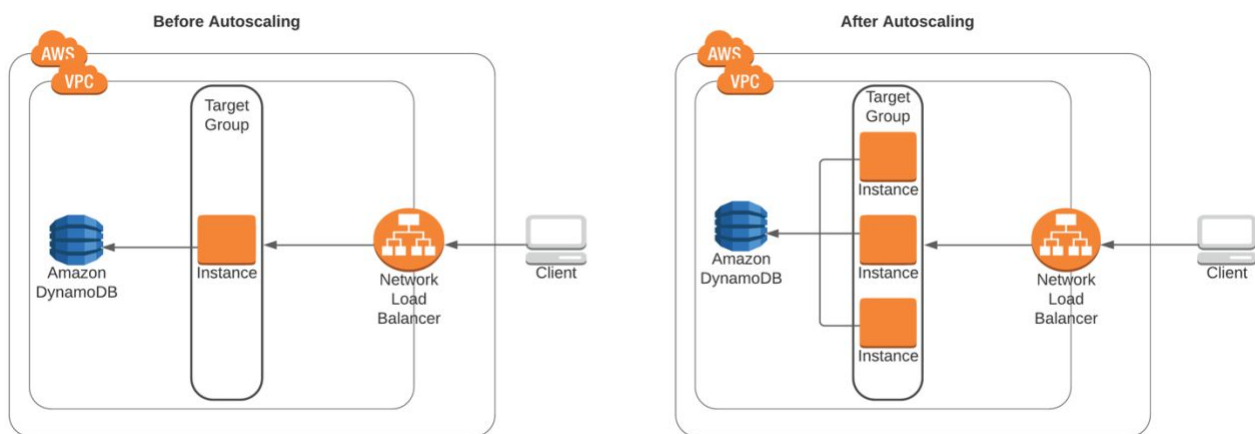
Starting with a full AMI is a great way to launch an EC2 instance. It is fast and efficient. However, the base AMI may not have all the necessary services installed and may need the latest software patches. There are two possible ways to take an AMI and configure it for an organization's use. The first option is to launch the EC2 instance with the premade AMI. After the instance boots, manually update the operating system and then install necessary services. Alternatively, the instance can be automatically updated upon boot with a bootstrap script. Bootstrap scripts for Linux can be as simple as a basic shell script. Windows systems can be bootstrapped with a power shell script.

## Autoscaling

One of the key drivers to the move to the cloud is autoscaling. In the traditional environment, servers are configured for peak demand plus a growth factor to meet increased demand over time. With traditional servers, capacity must always exceed demand, because if demand were to increase beyond the server's capacity, either the site would become unavailable or business could be lost. Oversizing can become very expensive, causing the organization to pay for resources that may never use.

Autoscaling completely changes the computing paradigm. Autoscaling enables an organization to size its computing resources based upon average demand. If demand for computing resources increases, autoscaling can spin up another instance as needed. Therefore, an organization will have almost unlimited computing capacity, while paying for only what is used. Many solutions architects consider autoscaling one of the best features of the cloud.

The diagram below shows how autoscaling increases compute instances to handle increased load.



## Instance Purchasing Options

AWS has numerous options for purchasing computing instances. Each type of instance is optimized for different computing requirements. The type of computing instances available can be seen below:

- On-demand instances
- Reserved instances
- Scheduled reserved instances
- Spot instances
- Dedicated hosts

## **On-Demand Instances**

On-demand computing instances are computing instances that are available when needed. On-demand instances are charged by either the second or hour of use, and facilitate autoscaling. On-demand instances are optimal when reliable computing capacity is needed without complete knowledge of how long the application will be used or its capacity requirements.

## **Reserved Instances**

A reserved instance is an instance where an organization commits to purchase a specified compute capacity for a specified period of time. It is essentially a contract to purchase a computing instance and can be for one to three years. By purchasing instances based upon long-term use, the organization can receive substantial savings over on-demand pricing. Reserved instances are optimal when an organization knows how long the application will be used and its capacity requirements.

## **Scheduled Reserved Instances**

A scheduled reserved instance is a special type of reserved instance. This type of instance is optimal when you have a need for a specific amount of computing power on a scheduled basis. For example, if an organization has a mission-critical batch job that runs every Saturday and Sunday.

## **Spot Instances**

A spot instance is an instance that is pulled from unused AWS capacity. Spot instances are sold in an auction-like manner in which an organization places a bid. If the bid price is equal to or greater than the spot price, a spot instance is purchased. Spot instances are deeply discounted and can be a great option. The drawback of spot instances is that they can be terminated if the spot price goes above the price that was bid on the instances. Spot instances are ideal when an organization needs extra computing capacity at a great price for non-mission-critical use.

## **Dedicated Hosts**

A dedicated host is a dedicated server. Dedicated hosts are bare metal servers. A bare metal server is a physical server without an operating system or applications installed. With a dedicated host, the organization can install any operating system or application required. Dedicated hosts are optimal when an organization needs access to system-level information, such as actual CPU usage. Dedicated hosts are an excellent option when an application that has a license that is dedicated to a physical machine.



## **Tenancy Options**

After the computing platform is chosen, it is necessary to determine the tenancy of the computing platform. The tenancy, or where the instances are located, can make a substantial impact on performance and availability. The tenancy options are:

- Shared tenancy
- Dedicated instances
- Dedicated hosts
- Placement groups

### **Shared Tenancy**

This is the standard tenancy for an EC2 instances. With shared tenancy, the physical server hosted at AWS will contain virtual machines for several customers.

### **Dedicated Instance**

This is a server that is completely dedicated to a single customer. This server can house multiple virtual machines. All virtual machines on the dedicated instance belong to the single customer who owns the instance.

### **Dedicated Host**

As previously described, this is a bare metal server and is dedicated to a single customer.

### **Placement Groups**

AWS uses the concept of placement groups. Placement groups are where the computing instances are located. There are three options for placement groups. The options below are covered in the VPC section:

- Clustered
- Spread
- Partitioned

## **Securing EC2 Access**

Keeping an organization's technology secure requires an end-to-end security posture. EC2 instances are no exception, and they should be kept as secure as possible. Keeping the instance patched with the latest security updates and turning off unnecessary services is a major part of securing EC2 instances. AWS provides a security feature called security groups. Security groups can dramatically help in locking down EC2 services. Note that security groups are an essential

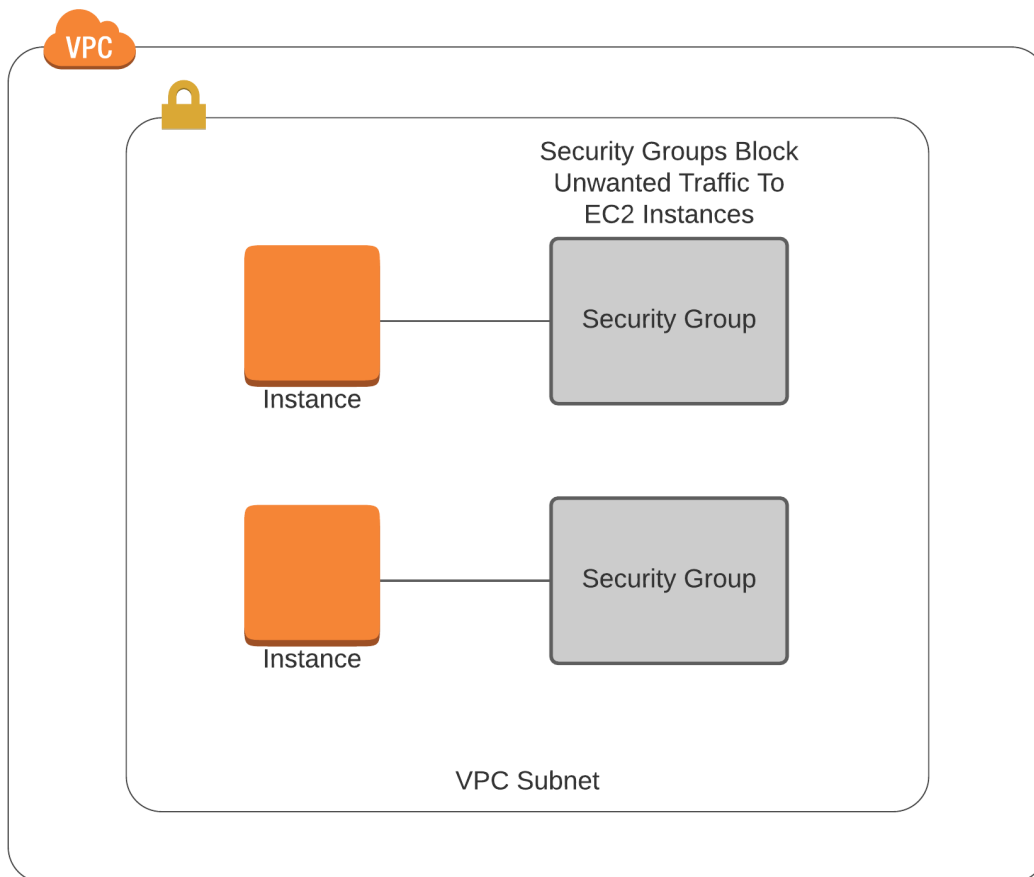
part of many AWS services. Security groups will be covered in depth in the security section of this book.

## Security Groups

A security group is a virtual firewall that lets an organization configure what network traffic is allowed into the server or AWS service. There are several key concepts with security groups:

- The default security policy is to deny traffic. An organization must explicitly permit desired traffic for the security group, or all traffic will be denied.
- Security groups include the source and destination IP address, protocol TCP/UDP, and port numbers.
- The security group is only as effective as its configuration. Be as specific as possible and allow only the traffic necessary into the system. Block all other traffic.

The diagram below shows how security groups protect compute instances by keeping unwanted traffic out of the server.



## IP Addresses for EC2 Instances

In order for an EC2 instance to function on a network, the instance must have an IP address. EC2 instances can have a private IP address, public IP address, or both. Instances also come with a fully qualified domain name assigned by AWS that can be used to connect to an EC2 instance. There are several key components to addressing EC2 instances:

- IP addresses are assigned to network interfaces and not computing systems.
- Depending upon the instance type, an instance can have multiple network interfaces.
- Each network instance must be on a different subnet, then other interfaces on the EC2 instance.
- Each IP address assigned to an interface must be unique within the VPC.
- IP addresses can be IPv4 or IPv6.
- All interfaces are automatically assigned an IPv6 globally unique address, which can be manually disabled.
- EC2 instances with public IP addresses with an internet gateway are reachable from the internet.
- EC2 instances with private IP addresses are not reachable from the internet unless a NAT instance and an internet gateway is used.
- EC2 instances with private IP addresses and a NAT gateway without an internet gateway will not be reachable from the internet, but will be able to connect to the internet for operating system patches and other needed connectivity.

Note the section on VPCs discusses networking and IP addressing in much more depth.

## Accessing EC2 Instances

Accessing and managing EC2 instances are critically important to maintaining a high-availability architecture. EC2 instances can be accessed via the following methods:

- Directly from the EC2 console.
- Via Secure Shell (SSH) for Linux machines.
- Via Remote Desktop Protocol (RDP) for Windows systems.
- Some management can be configured over the API using the AWS SDK.

## Labs

- 1) Launch an EC2 instance in the same manner as in the previous section. Install an application of your choice. Create an AMI from this instance. Link on how to create an AMI from an EC2 instance below.  
<https://docs.aws.amazon.com/toolkit-for-visual-studio/latest/user-guide/tkv-create-ami-from-instance.html>

- 2) Create a new instance from the AMI you created. Link on how to create an EC2 from an AMI below.  
<https://aws.amazon.com/premiumsupport/knowledge-center/launch-instance-custom-ami/>
- 3) Delete all running instances when you are completed with this lab.

## **Chapter 4 Introduction to Databases**

### **What Is a Database?**

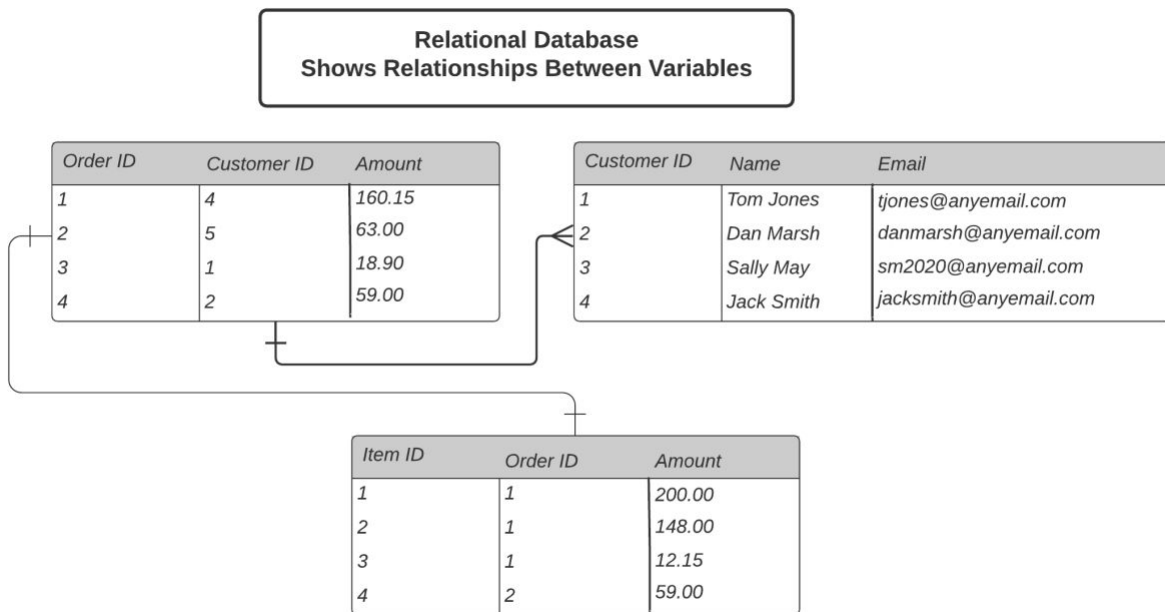
Databases have become such a critical component of modern information systems. But what exactly is a database and why are databases used? Databases are applications that allow for storage of large amounts of information. Databases can help an organization find key performance metrics from their data in order to make strategic business decisions. In the modern application environment, most web applications and many enterprise applications integrate with databases for information storage, management, and access to information. AWS has four forms of databases:

- Relational databases
- NoSQL databases
- Data warehousing databases
- Data lakes

### **Relational Databases**

Relational databases are the most common form of databases. Relational databases help organizations find the relationships between different aspects of their business by showing how data is related to each other. Relational databases store data in a manner that is very similar to a spreadsheet, with columns and rows.<sup>13</sup>

The diagram below shows how relational databases help show the relationship between different variables.

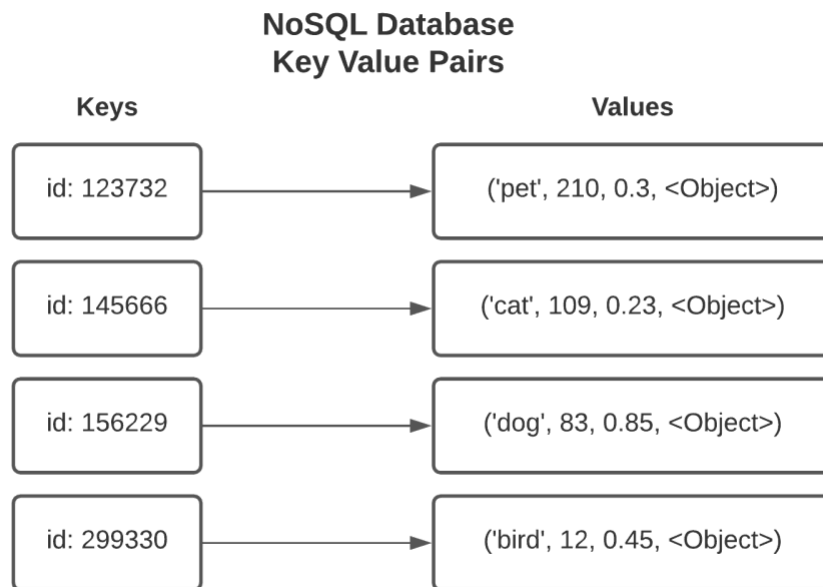


## NoSQL Databases

A NoSQL database stands for “not only SQL”. NoSQL databases facilitate enhanced flexibility and scalability by allowing a more flexible database schema. NoSQL databases can handle structured and unstructured data. Being able to work with structured and unstructured data can allow NoSQL databases to scale far beyond relational databases.<sup>14</sup> NoSQL databases are optimal under the following circumstances:

- When you need to store large amounts of unstructured data.
- When the database schema may change.
- When you need flexibility.
- When an organization needs rapid deployment of the database.

The diagram below shows how NoSQL databases work with key value pairs.

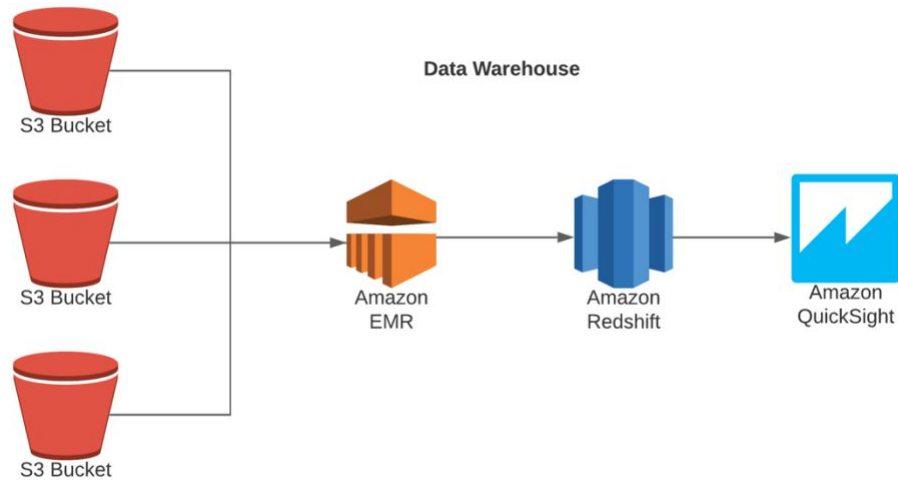


### Data Warehousing Databases

Data warehouses are designed to assist with business intelligence and analytics. Data warehouses are designed to perform analysis on large amounts of historical business data. A data warehouse is comprised of the following components:

- Database to store data.
- Tools for visualizing the data.
- Tool for prepping and loading data.

The diagram below shows an example of a data warehouse on the AWS platform.



## Data Lakes

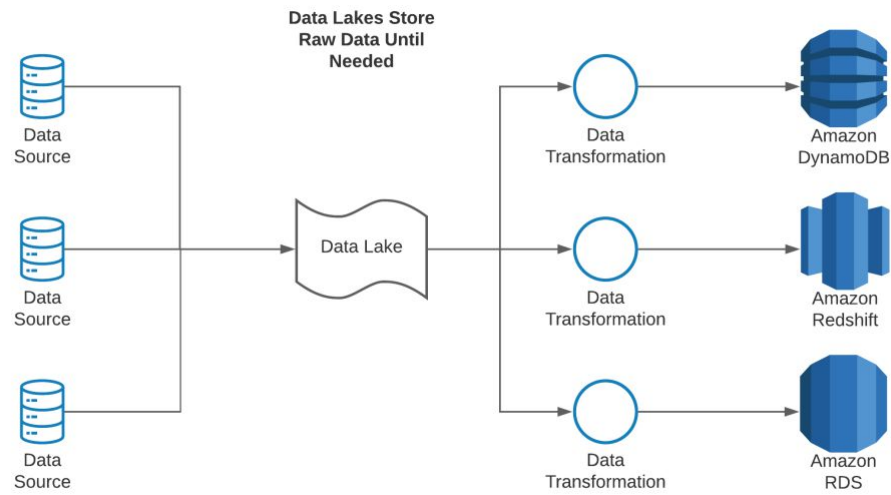
A data lake is not exactly a database, but it includes database elements, so it's included in the database section of this book.<sup>15, 16</sup>

### What Is a Data Lake?

A data lake is a repository that allows an organization to store structured and unstructured data in the same place. Data lakes allow an organization to store and analyze extremely large amounts of data.



The diagram below shows an example of a data lake on the AWS platform.



## Benefits of a Data Lake

Data lakes allow an organization to store virtually any type of data at an almost unlimited scale. Unlike a database, a data lake can store data in its native format until it is needed by another application. In a data lake, the data can be queried and searched for relevant data and solutions to current problems. Data lakes are highly adaptable and can be changed at any time to meet an organization's requirements.

## Relational Databases

Relational databases are the most common type of database and are best for structured data. Relational databases show the relationships between different variables stored in the database. With relational databases, as soon as data is written, it will be immediately available for query. The instant consistency is based upon relational databases following the ACID model.<sup>17</sup> More information on the ACID model can be seen below:

- Atomic – Transactions are all or nothing.
- Consistent – Data is consistent immediately after writing to the database.
- Isolated – Transactions do not affect each other.
- Durable – Data in the database will not be lost.

In the AWS environment multiple relational databases are supported. Supported databases include:

- Amazon Aurora
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle Database
- PostgreSQL

### **Amazon Aurora**

Amazon Aurora is the Amazon-branded relational database service. It is a fully managed database service. Aurora is MySQL- and PostgreSQL-compatible database. Aurora is a high-performance database with speeds up to five times faster than MySQL and three times faster than PostgreSQL databases.

Aurora is typically used in enterprise applications and software as a service (SAAS) applications.

### **MariaDB**

MariaDB is an open-source relational database. It was created by the developers of the MySQL database. MariaDB has additional features and advanced functionality when compared to MySQL. MariaDB supports a larger connection pool and is comparatively faster than MySQL but doesn't support data masking and dynamic columns.

### **Microsoft SQL Server**

Microsoft SQL Server is the Microsoft-branded relational database solution. AWS supports multiple versions of Microsoft SQL:

- SQL Server 2008
- SQL Server 2012
- SQL Server 2014

Microsoft SQL Server enables organizations to bring their Windows-based workflows to the cloud. Microsoft SQL Server offers tools including the SQL Server Management Studio to help manage the infrastructure. Microsoft SQL Server supports high-availability clustering and failover options, but in a different manner than other relational databases. Please check Microsoft documentation for the most up-to-date information when configuring Microsoft SQL databases.

AWS supports four versions of the Microsoft SQL databases:

- Enterprise
- Express
- Standard
- Web

## MySQL

MySQL is one of the original open-source relational databases and has been around since 1995. MySQL is extremely popular and is used in a wide variety of web applications.

## Oracle Databases

Oracle is one of the most popular relational databases in the world. It has an extensive feature set and functionality. Unlike open-source databases, oracle databases are developed, licensed, and managed by Oracle. AWS RDS for Oracle supports multiple versions of the Oracle database:

- Standard
- Standard One
- Enterprise

Each of these supported versions of AWS RDS for Oracle have different performance, flexibility, and scalability options. AWS offers two licensing options with the AWS:

- License included – In this version the database is licensed to AWS. Two license options are available for this option:
  - Standard Edition One
  - Standard Edition Two
- Bring your own license – You have a license for Oracle, and you host your database on AWS. This provides much more license flexibility and is available with these license options:
  - Standard
  - Enterprise
  - Standard Edition One
  - Standard Edition Two

## PostgreSQL Database

PostgreSQL is an open-source relational database. It has a very advanced feature set, and enhanced functionality when compared to MySQL.

## DynamoDB

Many enterprises require a database with near unlimited scalability and flexibility beyond what can be achieved with a relational database. AWS offers a fully managed NoSQL database called DynamoDB for organizations that need the scalability and flexibility of a NoSQL database.<sup>18, 19,20,21</sup>

DynamoDB is a fully managed, high-availability NoSQL database service. DynamoDB has multiple advantages:

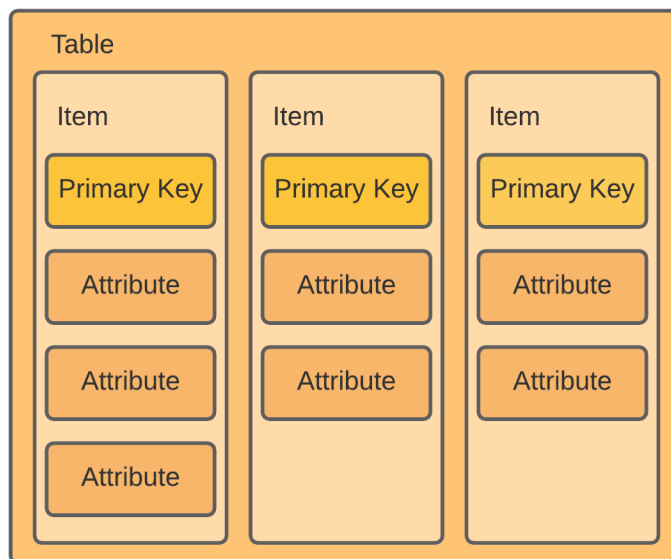
- Fully managed by AWS, so there is less management for the organization.
- Because it is serverless, there is near unlimited scalability, as the database is not bound to the capacity of a physical server or servers. Additionally, AWS manages servers, operating systems, and security.
- High availability – By default, DynamoDB is placed in multiple availability zones.
- High-performance storage – DynamoDB uses high performance SSD storage.
- Data protection – All data is encrypted by default in DynamoDB.
- Low latency – DynamoDB can be configured for sub millisecond latency when used with DynamoDB Accelerator (which is an in-memory cache for DynamoDB).
- Backups – DynamoDB can be backed up with minimal or no effect on database performance.

## Key Things to Know about DynamoDB

DynamoDB has a very flexible schema, as it is not bound to the same table and column scheme used by relational databases. This flexibility allows for significant customization and scalability. DynamoDB works best with name/value pairs in the primary index. DynamoDB can also have secondary indexes, which allows applications to use additional query patterns over traditional SQL databases.

DynamoDB secondary indexes can be global or local. Global indexes can span across all database partitions. Secondary partitions have virtually unlimited capacity. The only real limitation is a key value cannot exceed 10 gigabytes (GB). Local secondary indexes have the same partition key as the base table.

The diagram below shows an example of the DynamoDB key value pair architecture.



DynamoDB by default does not follow the ACID model used by SQL databases, which helps increase its scalability. DynamoDB by default uses the BASE model:

- Basically Available – The system should be available for queries.
- Soft State – Data in the database may change over time.
- Eventually Consistent – Writes to the database are eventually consistent. This means that a database read immediately after a write may not be available, but will be over time.

DynamoDB with the default configuration follows the eventually consistent model. However, DynamoDB can be configured to have strongly consistent reads if needed.

### **DynamoDB Pricing**

DynamoDB is priced based upon throughput. To achieve the best performance and pricing, it is necessary to provision read and write capacity. Read and write capacity are provisioned prior to use and should be set to accommodate for the organization's needs.

Autoscaling can also be used with DynamoDB. Autoscaling will watch the databases and scale up as needed, but it will not scale down. Since DynamoDB autoscaling does not scale back down, this method can lead to higher long-term costs.

DynamoDB can also be set up for on-demand capacity. However, on-demand capacity is more expensive than provisioned capacity. Therefore, for optimal pricing with DynamoDB, it's best to know read and write capacity when the database is provisioned.

### **When to Use DynamoDB**

DynamoDB is the optimal choice when near unlimited database scalability and low latency are required. Additionally, DynamoDB is an excellent choice when storing data from a large number of devices, such as internet of things (IoT). Some common DynamoDB use cases are:

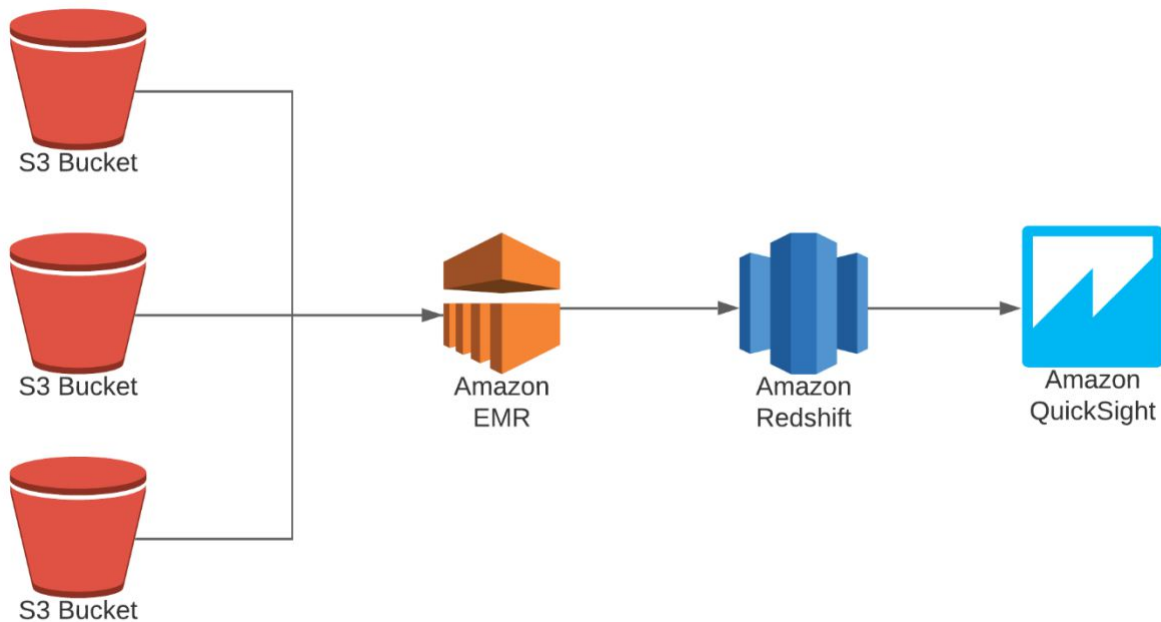
- Gaming applications
  - Storing game state
  - Player data stores
  - Leaderboards
- Financial applications
  - Storing a large number of user transactions
- E-Commerce applications
  - Shopping carts
  - Inventory tracking
  - Customer profiles and accounts

### **Data Warehousing on AWS**

Data warehousing is becoming a critical business tool. Data warehouses enable businesses to get actionable insights from their data. The AWS data warehousing solution is called Amazon Redshift. Amazon Redshift is a fast, powerful, and fully managed data warehouse solution. Amazon Redshift supports petabyte-scale data warehousing. Amazon Redshift is based upon PostgreSQL and supports SQL queries. Additionally, Redshift will work with many applications that perform SQL queries.<sup>22</sup>

Redshift is a highly scalable platform. The Redshift architecture is built around clusters of computing nodes. The primary node is considered a leader node, with supporting nodes called compute nodes. Compute nodes support the leader node. Queries are directed to the leader node.

The diagram below shows how data warehousing is performed with Amazon Redshift and other AWS services.



### Scaling Amazon Redshift Performance

Scaling Amazon Redshift is achieved by adding additional nodes. Amazon offers two types of nodes:

- Dense compute nodes – Dense compute nodes are based upon high speed SSD RAID arrays.
- Dense storage nodes – Dense storage nodes are based upon magnetic disk RAID arrays.

Generally speaking, SSD-based arrays have higher throughput than magnetic arrays. SSD-based arrays have much higher IOPS than magnetic-based RAID arrays and perform much better in applications that require high input/output (IO) performance.

### Database Storage Options

After determining the optimal database for an organization's needs, it is necessary to determine the proper storage options for the database. AWS databases are stored on EBS volumes, and the database storage options are:

- Provisioned IOPS (PIOPS)
  - Highest performance

- Lowest latency
  - Highest throughput
- General purpose SSD
  - High performance
  - Lower latency
  - Moderate throughput
- Magnetic storage
  - Moderate performance
  - Moderate throughput
  - Lowest cost
  - Highest latency
  - Designed for light IO requirements

## **Database Management and Optimizations**

There are many components to successfully architecting and scaling an enterprise-wide database. This section will cover the following topics:

- Backing up the database.
- Scaling the database.
- Designing for high availability.
- Protection of data with encryption.
- Extraction, transforming, and loading tools (ETL).

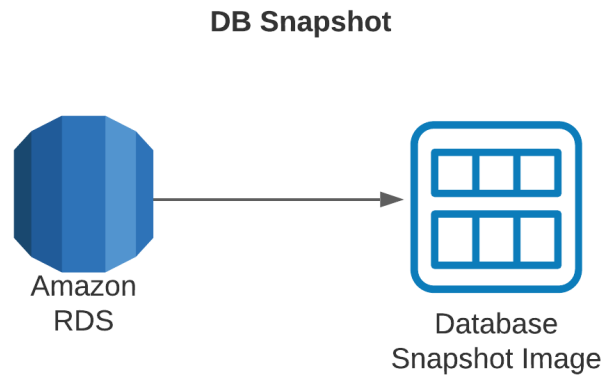
### **Backing Up the Database**

Databases are automatically backed up by AWS. Database backups copy the entire server, not just the data stored on the database. Backups can be retained for up to thirty-five days, which is configurable from one to thirty-five days. Automated backups happen at a defined window each day. While the database is being backed up, it may be unavailable or have significantly degraded performance.<sup>23</sup>

Databases can also be backed up manually. Manual backups are in the form of a DB snapshot. DB snapshots are a point-in-time copy of the databases EBS volume. DB snapshots are maintained until manually deleted.



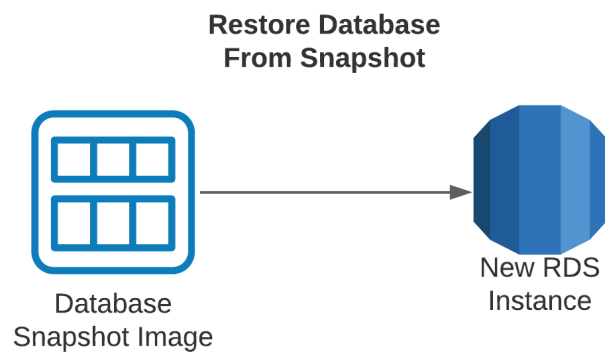
The diagram below shows how a database is manually backed up on the AWS platform.



### Restoring a Database Backup

Databases can easily be restored from backups. When an organization needs to restore a database, a new database instance is created. Since a new database is created, it will have a new IP address and DNS name. Therefore, if a database is restored, it may be necessary to update other applications with the new IP address or DNS name.

The diagram below shows how a database is restored from a snapshot image.



### Scaling the Database

Databases are a mission-critical application for many enterprises. As a mission-critical application, the database must scale to meet an organization's needs. There are several

methods to increase the scalability of the database. Often it will take a combination of scaling methods to meet an organization's needs.

The first scaling method refers to scaling up versus scaling out. Scaling up refers to simply increasing the capacity of the server housing the database. At some point scaling up is not feasible, as a database can exceed the capacity of even the most powerful servers. Scaling out, by comparison, involves adding additional compute instances. Scaling out has two options: partitioning for NoSQL databases and read replicas for relational databases.

### **Scaling Out for NoSQL Databases**

Partitioning the database involves chopping the database into multiple logical pieces called shards. The database has the intelligence to know how to route the data and the requests to the correct shard. Effectively, sharding breaks down the database into smaller, more manageable pieces. Partitioning the database is effective for NoSQL databases like DynamoDB and Cassandra.

### **Scaling Out for Relational Databases**

Relational databases are scaled out by adding additional servers. With relational databases, the additional servers are called read replicas. A read replica is a read-only copy of the main database instance. Read replicas are synchronized in near real time. Read replicas are used to decrease the load on the main database server by sending read requests to read replicas as opposed to the primary server. This reduces CPU, memory, and disk IO on the main server. AWS supports up to five read replicas. Read replicas are helpful in the following scenarios:

- When there is a lot of read activity.
- To increase performance, but read replicas are for performance and not disaster recovery.
- When query traffic is slowing things down.
- For more capacity, as offloading read requests from the main database can save valuable resources.

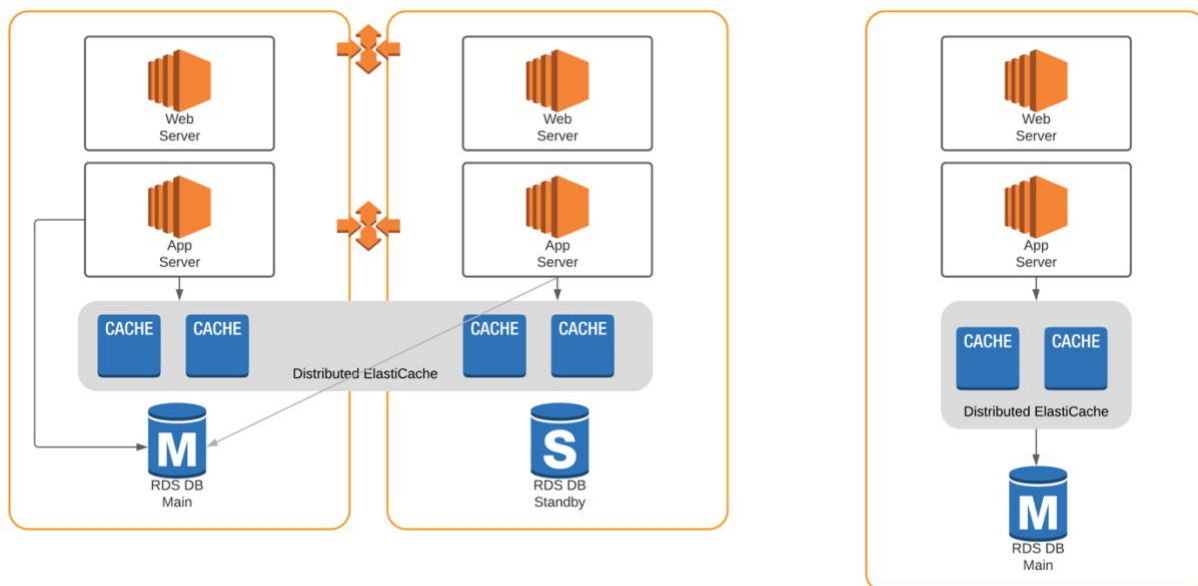
### **Database Caching**

Database caching is another means to increase the scalability of a database. Caching is a method that takes frequently accessed information and places it in memory, so the request does not need to be forwarded to the database server. Caching works by taking the first request for information to the database server. The server then responds to the request, and the cache temporarily stores the results of the request in memory. Future requests for the same information will be responded to by the cache and will not be sent to the database server. Future requests for information not stored in the cache will be sent to the server. Caching data reduces requests to the server, freeing up server resources. To prevent stale data, the cache

will not keep information in memory forever, instead the cache has a timeout to expire old data and make room for fresh data. This timeout, referred to as the time to live (TTL) can be configured based upon an organization's needs.

AWS supports two caching types. These are ElastiCache for Memcached and ElastiCache for Redis. Memcached is designed for simplicity. ElastiCache for Redis has a substantial feature set and functionality. Caching is an excellent method to increase scalability. Caching is beneficial only when there are frequent requests for the same information, or queries, as if all requests are for new information, they will all be sent to the main server, mitigating any benefit to the cache.

The diagram below shows an example of a database caching on the AWS platform.



## Database Queueing

Database queueing can make a significant improvement in increasing the write performance of a database. Additionally, queueing can significantly help reduce CPU usage and other resources in the database. AWS has a queueing services called Simple Queue Service (SQS). Using SQS effectively decouples the database writes from the actual database.<sup>24</sup> The database works as in the following manner:

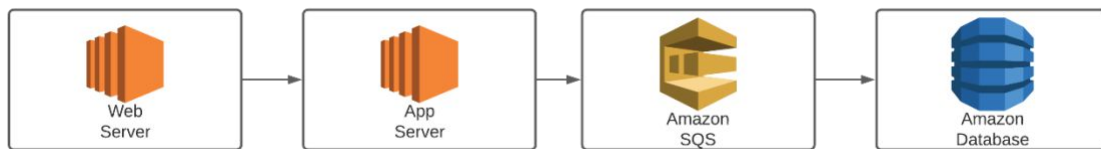
1. Data is sent to the SQS queue.
2. The queue looks at the database.
3. If the database is free, the message is sent from the queue to the database and removed from the queue.

4. If the database is busy or unavailable, the message waits in the queue until the database is available.
5. When the database is available, the message is sent to the database and removed from the SQS queue.

AWS offers two versions of the SQS queue:

- AWS Standard SQS Queue – This is a simple queue to temporarily store messages prior to being written to the database. With Standard SQS queues, there is no guarantee to the order of messages leaving the queue. This is the default option.
- First in, first out (FIFO) – This option guarantees that the messages will exit the queue in the order that they were received

The diagram below shows an example of an SQS queue on the AWS platform.



### How Does SQS Help?

SQS helps by reducing write contention to the database. SQS helps with availability as well; if the database is temporarily down, messages can be retained in the queue. Since messages are retained in the queue until the database is ready, it can dramatically smooth CPU, memory, and disk IO performance. Additionally, the number of messages in the SQS queue can be used to scale out the database or computing service using the SQS queue.

### When to Use SQS

SQS is optimal to use in the following circumstances:

- To increase scalability when there are a lot of write requests to the system.
- To decrease the load on the database behind the SQS queue.
- When it's not known exactly how much performance is needed, but the organization wants to be able to account for large spikes in traffic.
- When extra insurance is desired that critical messages won't be lost.
- When you want to decouple your application to increase availability, modularity, and scalability.

## Designing a High-Availability Database Architecture

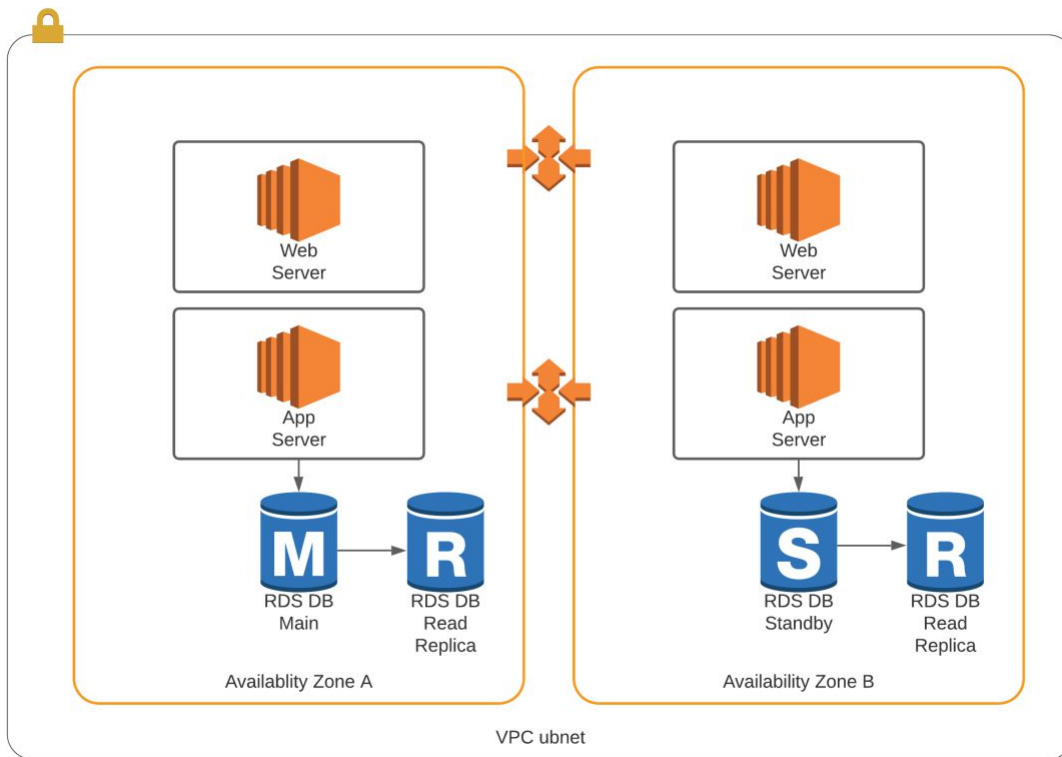
Given the crucial nature of databases in the enterprise computing environment, making sure the database is available when needed is of utmost importance. The key to all high-availability designs is to avoid any single point of failure.

As a reminder, the AWS network is divided into regions and zones. Regions are large geographic areas, while an availability zone (AZ) is really a data center inside of a region. Regions may have many availability zones.

A high-availability database architecture will have database instances placed in multiple availability zones (multi-AZ). In a multi-AZ environment, there are multiple copies of the database, one in every availability zone. It is important to note that multi-AZ environments do not increase database performance. Multi-AZ environments are for redundancy to enhance availability purposes. In a multi-AZ environment, data from the primary (master) database is synchronously copied to the backup database in the other AZ. If the primary database were to fail, the database instance in the other AZ will take over. A failover to the backup database will be triggered in the following circumstances:

- The primary database instance fails.
- There is an outage in an availability zone.
- The database instance type is changed.
- The primary database is under maintenance (i.e., patching an operating system).
- A manual failover has been initiated (i.e., reboot with failover).

The diagram below shows a high-availability application and database architecture.



## Labs

- 1) Create a MySQL database. Link on how to create a MySQL database below.  
[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_GettingStarted.CreatingConnecting.MySQL.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.MySQL.html)
- 2) Create a read replica of the database. Link on how to create a read replica below.  
[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_ReadRepl.html#USER\\_ReadRepl.Create](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html#USER_ReadRepl.Create)
- 3) Promote the read replica to be the master database. Link on how to promote a read replica below.  
[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_ReadRepl.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html)
- 4) Set up DynamoDB. Link on how to set up DynamoDB below.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SettingUp.DynamoWebService.html>
- 5) Create a table in DynamoDB. Link on how to create a table in DynamoDB below.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/getting-started-step-1.html>

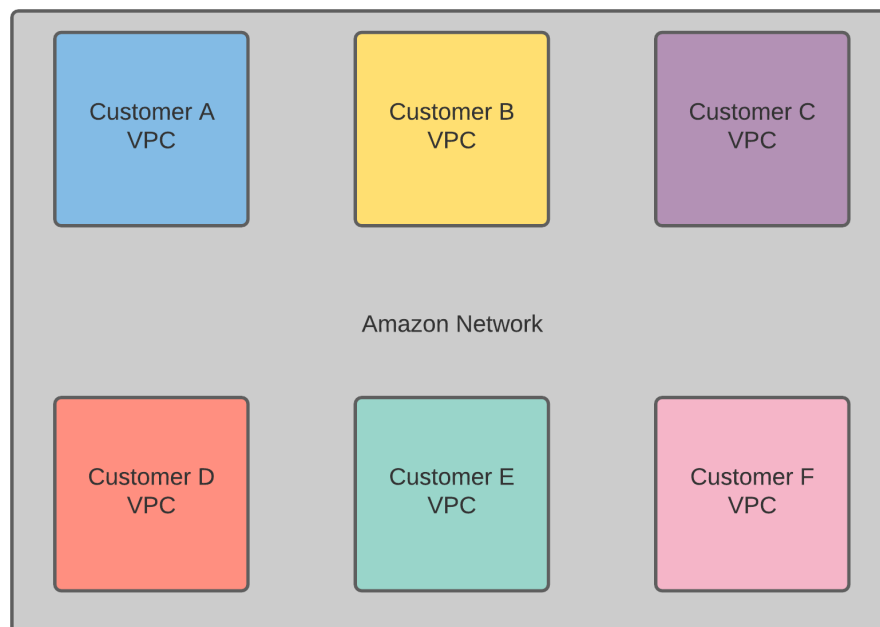
- 6) Delete all databases and tables when you are completed this lab to avoid being billed for services that you are not using.

## Chapter 5 The AWS Virtual Private Cloud

### What Is the AWS VPC?

The AWS VPC is essentially a private virtual network inside the AWS network. While AWS is physically a shared network, each VPC is logically isolated from other AWS customers. The AWS network supports private and public addresses for each of their customers. Since AWS customers are logically separated, there is no contention for IP address space between VPCs. Each VPC will have its own routing table that is responsible for directing traffic. As with any network, a proper IP addressing scheme is essential for scalability.<sup>26</sup>

The diagram below shows an example logically isolated customer VPCs on the AWS platform.



### The OSI Model

Throughout this book, especially the VPC section, we often reference the open systems interconnect (OSI) model. The OSI model divides network communications into seven layers. Each layer of this model has specific functions for network communication. Knowledge of the OSI model can be helpful in troubleshooting and understanding the components of network communication. The table below shows the seven layers of the OSI model and the associated functionality at each level.<sup>27</sup>



Operating Systems Interconnection Model (OSI)				
Layer	Number	Function	Examples	Name
Application	7	User Interface	HTTP, DNS, SSH	Data
Presentation	6	Presentation and Data Encryption	TLS	Data
Session	5	Controls Connection	Sockets	Data
Transport	4	Protocol Selection	TCP, UDP	Segments
Network	3	Logical Address	IP Address	Packets
Datalink	2	Hardware Connection	MAC Address	Frames
Physical layer	1	Physical Connection	Wire, Fiber	Bits

## IP Addressing

An IP address is a logical address assigned to a computing device that identifies that device on the network. An IP address is similar to an address on a home. In order for mail to be delivered, the house must have a unique address so that the postal service can identify the correct home and deliver the mail. Every address must be unique, even if the only thing that separates similar looking addresses is the postal code. IP addresses are no different. For any device to talk to another device on an IP network, their addresses must be unique. There are two versions of IP addresses:

- IPv4 – Original IP address used in networking and has been in use since 1970s and is still the dominant address space.
- IPv6 – Newer IP addressing model is designed to overcome the limitations of IPv4.

IP addresses operate at layer 3 of the OSI model. An IP address is a logical address, in that they are assigned to a network interface. IP addresses are not hard coded, like a MAC address (layer 2), that is permanently assigned to an ethernet interface. IP addresses are a 32-bit address, which was perfect when the internet was formed. However, the internet grew far beyond what anyone expected, and there were not enough public IP addresses. The Internet Engineering Task Force (IETF) came up with two solutions to the IP address space shortage.<sup>28,29</sup> The first solution is to provide private IP address space as specified by the Request for Comments (RFC) 1918 and IPv6 addresses. Private IP addresses are to be used on internal networks and are not globally routable. Private IP addresses are available in the following address space:

- 10.0.0.0/8
- 172.16.0.0/16 through 172.31.0.0/16
- 192.168.0.0/16

## IP Address Classes

IPv4 has five classes of IP address space. Address classes are legacy, and not really used in today's modern environment. IP address classes are covered for historical purposes and so the reader can better understand classless interdomain routing (CIDR).

- Class A addresses
  - 1.0.0.0- 126.255.255.255/8
- Class B addresses
  - 128.0.0.0- 191.255.255.255/16
- Class C addresses
  - 192.0.0.0- 223.255.255.255/24
- Class D addresses (Multicast)
  - 224.0.0.0- 239.255.255.255
- Class E addresses (Experimental)
  - 240.0.0.0- 255.255.255.255

Classful addresses are not used anymore since there is a shortage of IP addresses. Instead, subnetting is used to optimize IP address usage and availability.

## Subnetting and Supernetting

Since there are a limited number of IP addresses, it's essential to use IP addresses carefully. One way to make use of an organization's IP address space with subnetting. Subnetting is effectively taking an IP network and chopping it into smaller networks. Please see the graphic below:

Network	Subnet Mask	Effective Addresses	Effective AWS Addresses
192.168.1.0	255.255.255.0	253	
Submitted To /28 Subnets			
192.168.1.0	255.255.255.240	14	11
192.168.1.16	255.255.255.240	14	11
192.168.1.32	255.255.255.240	14	11
192.168.1.48	255.255.255.240	14	11
192.168.1.64	255.255.255.240	14	11
192.168.1.80	255.255.255.240	14	11
192.168.1.96	255.255.255.240	14	11
192.168.1.112	255.255.255.240	14	11
192.168.1.128	255.255.255.240	14	11
192.168.1.144	255.255.255.240	14	11
192.168.1.160	255.255.255.240	14	11
192.168.1.176	255.255.255.240	14	11
192.168.1.192	255.255.255.240	14	11
192.168.1.208	255.255.255.240	14	11
192.168.1.224	255.255.255.240	14	11
192.168.1.240	255.255.255.240	14	11

In this table, the 192.168.1.0/24 network has been submitted into sixteen /28 subnets.

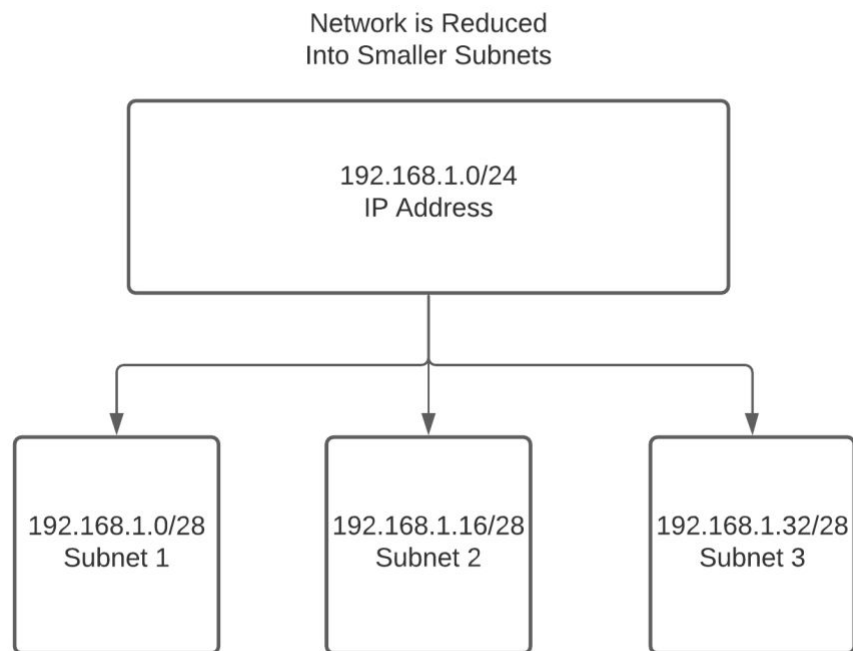
Subnetting is critical for two reasons. The first reason is that every interface on a system needs to be on a different network or subnet. The second reason is there is a practical limitation of how many hosts can be on a subnet due to system broadcasts.

All interfaces need to be on a different subnet. Imagine a router with thirty interfaces. Each interface needs to communicate only with the router on the far end of the connection. In practicality, only two IP addresses are needed—one on each side of the connection. If the network 192.168.1.0/24 were attached to both sides of a WAN link, instead of only using two addresses, this link would use up all 253 addresses available on that subnet. By comparison, subnetting from a /24 to a /28 would allow for sixteen subnets available that would each support eleven hosts. Note that this is a reduction from the fourteen available addresses that would be present with a /28 subnet mask. This is because AWS reserves the first three IP addresses and the broadcast address space. An additional point to remember is that the smallest subnet supported by the AWS platform is a /28.

The second key reason that subnetting is essential is related to constraining broadcast traffic. Host systems often identify each other by sending broadcast traffic to the local subnet. Broadcast traffic is different than traditional traffic. With traditional traffic, a system sends a message to another system. With broadcast traffic, when a host sends a broadcast, every host on the subnet sees and must process the broadcast. Additionally, network switches forward

broadcast traffic out every port except the port where the broadcast was sent. Broadcast traffic can easily overwhelm computing and network hardware. So, limiting the size of broadcast domains is essential. Limiting the reach of broadcasts is achieved by using smaller subnets and routing traffic between subnets.

The diagram below shows an example of a network being partitioned into smaller subnets.



## Supernetting

Supernets are the exact opposite of subnets. Supernets combine multiple smaller subnets into a larger network called a supernet. As we have previously stated, optimizing performance often involves reducing the size of broadcast domains with effective subnetting. Therefore, supernetting is generally used more to optimize routing than for addressing computer systems.

### Supernetting to Optimize Routing

Routers distribute traffic across the network by building a map of all available subnets. These subnets are subsets of the full class A, B, or C networks. Therefore, modern routing is called classless interdomain routing (CIDR). Routers often have memory and CPU limitations that limit the number of routes the router can support. As an example, AWS permits only 100 routes in a VPC. To minimize the number of routes in the routing table while maintaining full reachability, it is often necessary to summarize routes. Route summarization is effectively taking several subnets and supernetting them into a single network. A full discussion of routing, switching, and IP addressing is beyond the scope of this book. For those interested in detailed routing and

switching information, we recommend the book *Routing TCP/IP* by Jeff Doyle and Jennifer Dehaven Carroll.

## How It Works

Let's look at this section of a VPC routing table:

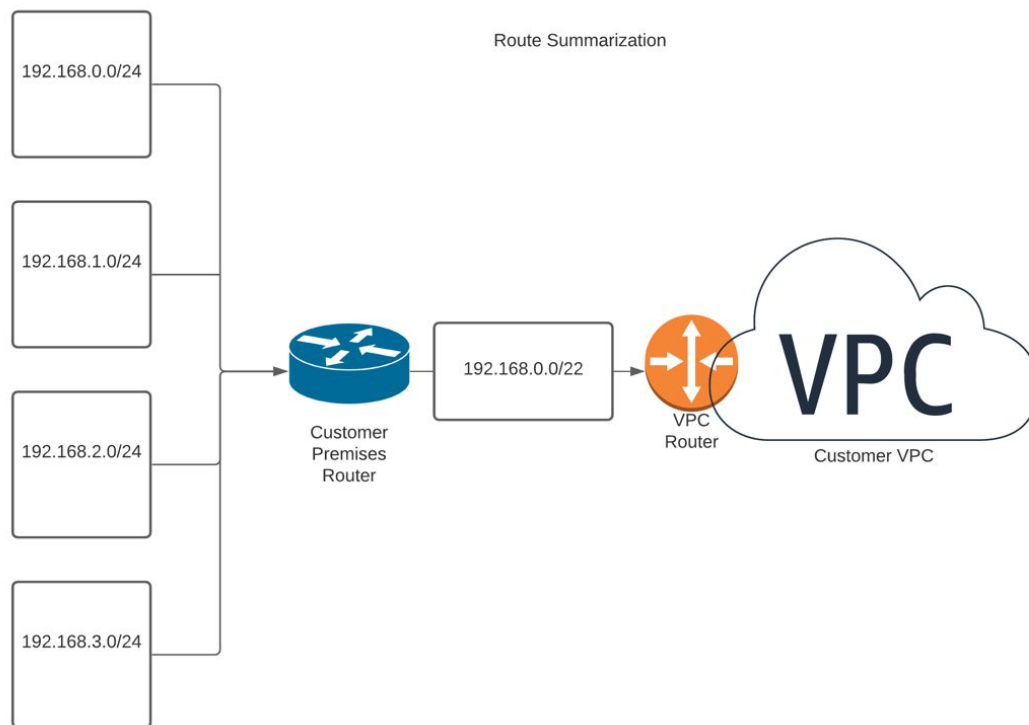
- Destination 192.168.0.0/24 target igw-123456789098765
- Destination 192.168.1.0/24 target igw-123456789098765
- Destination 192.168.2.0/24 target igw-123456789098765
- Destination 192.168.3.0/24 target igw-123456789098765

Instead of putting all of these routes in a routing table, the subnets can be supernetted and summarized into the following route:

- Destination 192.168.0.0/22 target igw-123456789098765

As you can see, we have taken four routes and converted them into a single summarized route. This provides full reachability while saving valuable resources in the routing tables. With AWS allowing only 100 routes, every route counts. So, it's best to use an IP addressing scheme that can be summarized and to use route summarization whenever possible. This can also be used for traffic engineering on AWS. To learn more about traffic engineering using the BGP routing protocol that is supported by AWS, we recommend the book *Internet Routing Architectures* by Sam Halabi.

The diagram below shows an example of route summarization with supernetting.



## IPv6 Addresses

When IPv4 was invented, no one could have imagined the growth of the internet. Very quickly the internet would be out of addresses, and something more scalable would be needed. The Internet Engineering Task Force invented a new version of the internet protocol to overcome the weaknesses with IPv4. This new protocol was IPv6. To overcome the address shortage, the 32-bit binary address used with IPv4 was changed to a 128-bit hexadecimal address. This provides infinitely more address space and scalability. Realistically speaking, IPv6 address capacity is likely sufficient to provide every internet enabled device an IP address.

While IPv6 is the future, IPv4 is still the main IP addressing scheme in use today. IPv6 addresses are becoming more popular. In modern times, most mobile phones have an IPv6 address. AWS automatically assigns an IPv6 address to every interface.

## Components of a VPC

Now that we have reviewed the basic elements of IP addressing, it's time to discuss the AWS specific components of a VPC. The key components of a VPC can be seen below. This section will cover these VPC components in detail:

- VPC routing tables
- Internet gateway
- Egress-only internet gateway
- NAT instances and NAT gateways
- Elastic IP addresses (EIPs)
- VPC endpoints
- VPC peering
- Network access control lists
- Security groups

## Routing Tables and Routing

All VPCs effectively have a virtual router provided by AWS. The virtual router is used to direct traffic to its ultimate destination. Routers determine how to make traffic forwarding decisions based upon their routing tables. A sample routing table can be seen below:

Routing Table Example	Target	
172.16.1.0/24	Local	
192.168.0.0/16	pcx-123456	
192.168.1.0/24	pcx-654321	* most specific
0.0.0.0/0	igw-123456	

In the routing table above, it's clearly visible that the routing table has a destination subnet and a destination interface/gateway to forward traffic. The destination is referred to as the target in AWS. Note that if there are multiple paths to a destination, the most specific route will be chosen.

## How Routing Tables Work

Routers build a map of the network. The map of the network will show which interface to use to send traffic to its ultimate destination. Traffic will then be sent to the next router, which will have its map of the network. Packets are forwarded from router to router until they reach their ultimate destination.

The map of the network is called a routing table. Routing tables can be built statically or dynamically. Static routes are user configured, where dynamic routes are dynamically learned via a routing protocol. Static routes are ideal when there are very few paths to reach the ultimate destination. Dynamic routes are learned, which is excellent for large networks. Additionally, dynamic routing enables high availability, as the routers can re-route traffic to a backup path if needed. Dynamic routing can re-route traffic by detecting when a router or link is down and calculating an alternate path. Most, if not all, large enterprise networks use routing protocols as part of their high-availability architecture. There are essentially two kinds of routing protocols, and they can be seen below.

### **Interior Gateway Protocols (IGP)**

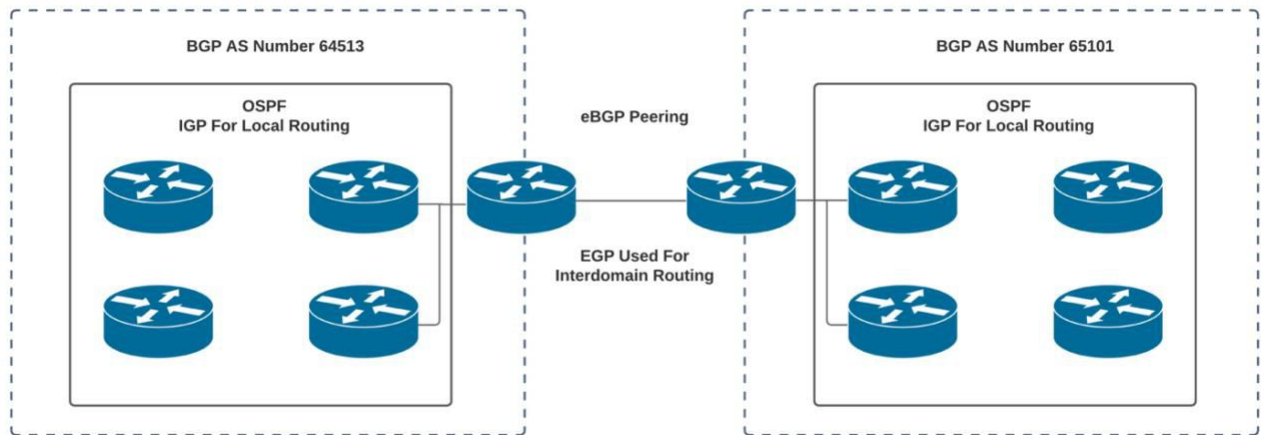
- Interior gateway protocols are used to exchange routing information inside of an organization.
- Interior gateway protocols provide a very detailed map of the organization's routes.
- Interior gateway protocols can detect outages and re-route traffic very quickly.
- Interior gateway protocols are tuned for performance at the expense of scalability.
- Some examples of interior gateway routing protocols include OSPF, IS-IS, and EIGRP.

### **Exterior Gateway Protocols (EGP)**

- Exterior gateway protocols are used to exchange routing information across organizations.
- Exterior gateway protocols provide extensive tuning tools, proving the means to engineer traffic, and filter routes for scalability, security, and proper routing.
- Exterior gateway protocols are slower to re-route traffic, as they are designed for scalability and tunability.
- Border Gateway Protocol (BGP) is the exterior gateway protocol used by the internet and AWS.
- Exterior gateway protocols are tuned to be able to store an enormous number of routes (assuming the routers have sufficient memory and CPU capacity). At the time of this writing, the internet routing table has over 800,000 routes.<sup>30</sup>



The diagram below shows how an IGP is used for internal routing and an EGP is used for interdomain routing.

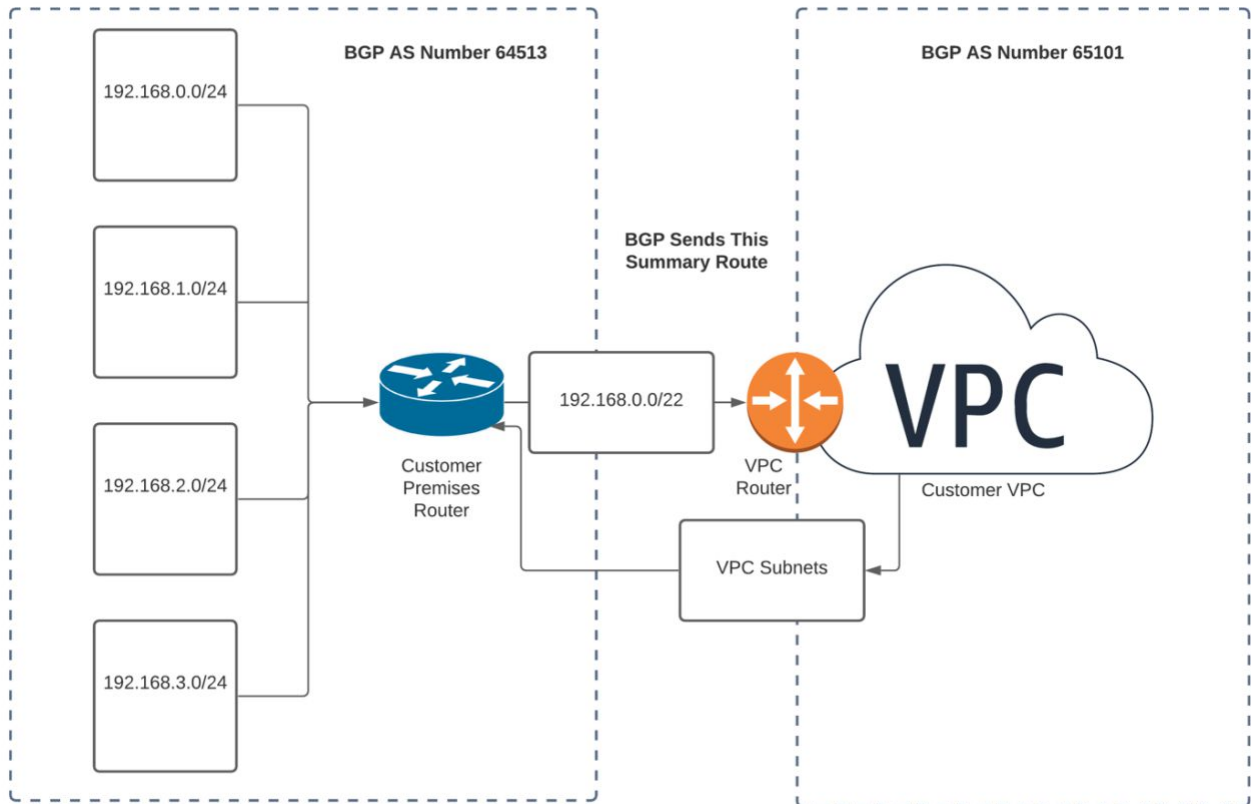


## Dynamic Routing with AWS

AWS supports connecting an organization to AWS with BGP. BGP is a highly tunable and scalable exterior gateway routing protocol. BGP runs on TCP port 179. It is essential when using BGP to connect to AWS that firewalls and network ACLs allow TCP port 179. BGP enables an organization to have multiple connections to the internet or AWS, and load share across these connections. As with all BGP routing, an autonomous system number is required to identify an organization to the AWS network.

BGP is required when using a direct connection to connect to AWS. AWS supports some of the available BGP tuning options like communities. The AWS BGP implementation supports the well-known community “no export”, which prevents a VPC from becoming a transit autonomous system for the internet (this means you won’t become an accidental internet service provider and have parts of the internet connecting through your VPC). AWS has a light BGP implementation when compared to core internet routers. AWS supports a maximum of 100 routes learned from BGP, therefore an IP addressing scheme capable of route summarization is necessary. AWS supports some of the BGP tuning options such as weight, local preference, and Autonomous System path (AS path). Please see below for an example of a system connecting with BGP to share routing information via BGP to AWS.

The diagram below shows how an organization would use BGP to connect to AWS to share routing information.



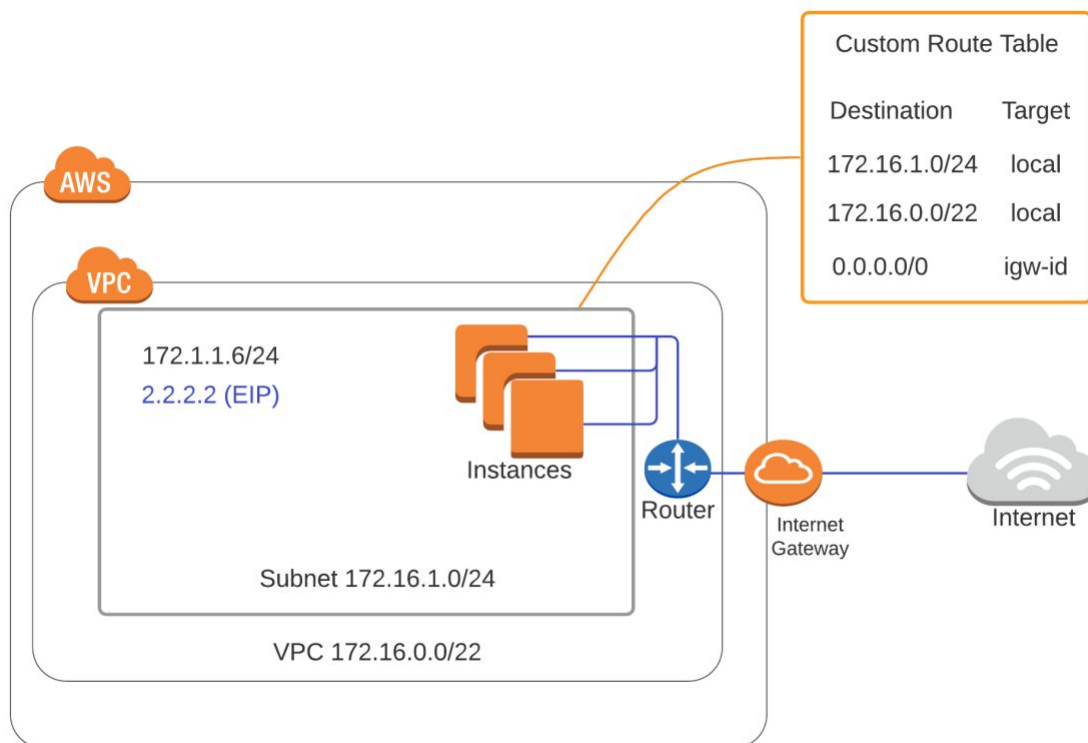
## Internet Gateways (IGW)

In order to connect to the internet, an internet connection and an internet gateway must be configured. The internet gateway is a router with an internet service provider connection. AWS provides internet service to VPC customers when the customer sets up an internet gateway. The AWS internet gateway is a high-availability, redundant internet router. When using the internet gateway, it will have a route to all internet destinations or a default route to an upstream provider. Additionally, the internet gateway will translate private IP addresses into a public address for internet connectivity.<sup>31</sup>

An internet gateway is created in the following manner:

1. Attach an IGW to the VPC.
2. Create a default route to send all internet destined traffic to the internet gateway.
3. Assign a public IP address to the gateway.
4. Configure security. Systems will be reachable from the internet, so it is essential that systems are patched for security vulnerabilities and firewalls; network ACLs and security groups are configured.

The diagram below shows an example of an internet gateway on the AWS platform.



### Egress-Only Internet Gateways

Egress-only internet gateways allow internet connectivity to IPv6 systems. IPv6 does not really use private address space. So essentially all IPv6 address are unique and globally routable. Therefore, IPv6 systems do not need NAT to connect to the internet, as there is no need to translate an internal address into an external address.

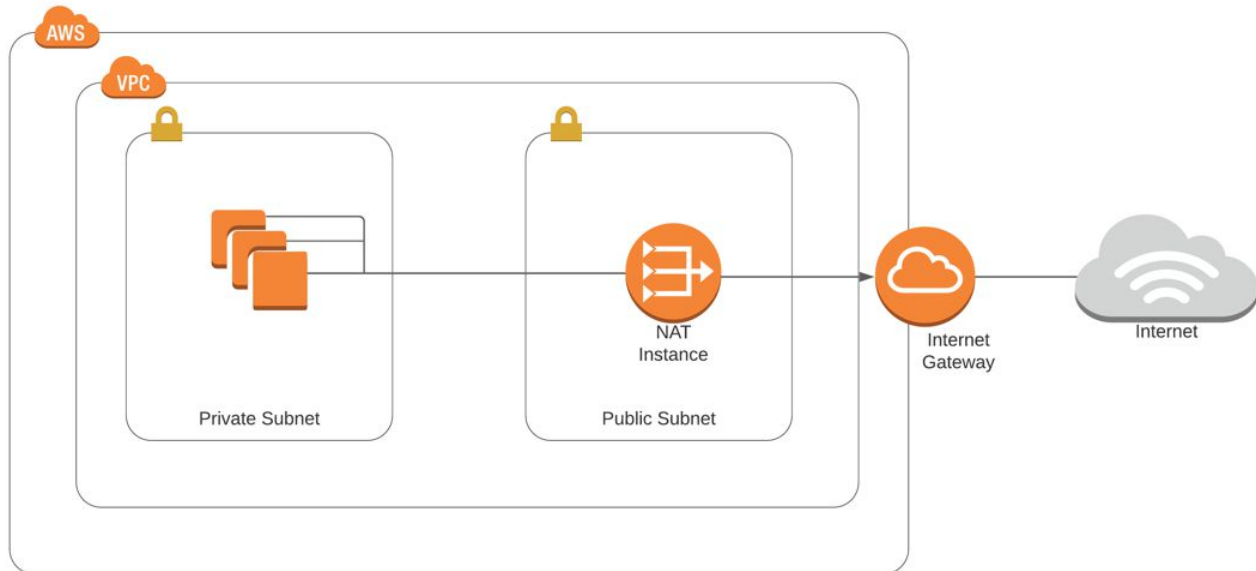
When using an egress-only internet gateway, systems will not be reachable from the internet. This type of internet gateway is stateful and allows traffic established by internal hosts to return to the AWS VPC. This allows internal systems to download software patches and upgrades from the internet while keeping outsiders from connecting into the AWS VPC from the internet. This is very similar in function to the NAT gateway without an internet gateway in IPv4.<sup>32</sup>

### NAT Instances

A NAT instance is a custom AWS virtual machine that translates private IP addresses into public IP addresses. The NAT instance is available as an AMI, and it runs on an EC2 instance. A NAT instance must be in a public subnet with a route to the internet gateway. This type of setup is

used for egress only, meaning internal systems can connect to the internet, but systems on the internet will not be able to connect to systems in the VPC. Additionally, the VPC routing table must have a default route to the internet gateway. This is really a legacy product and has largely been replaced with the AWS NAT gateway.<sup>33</sup>

The diagram below shows an example of a NAT instance on the AWS platform.

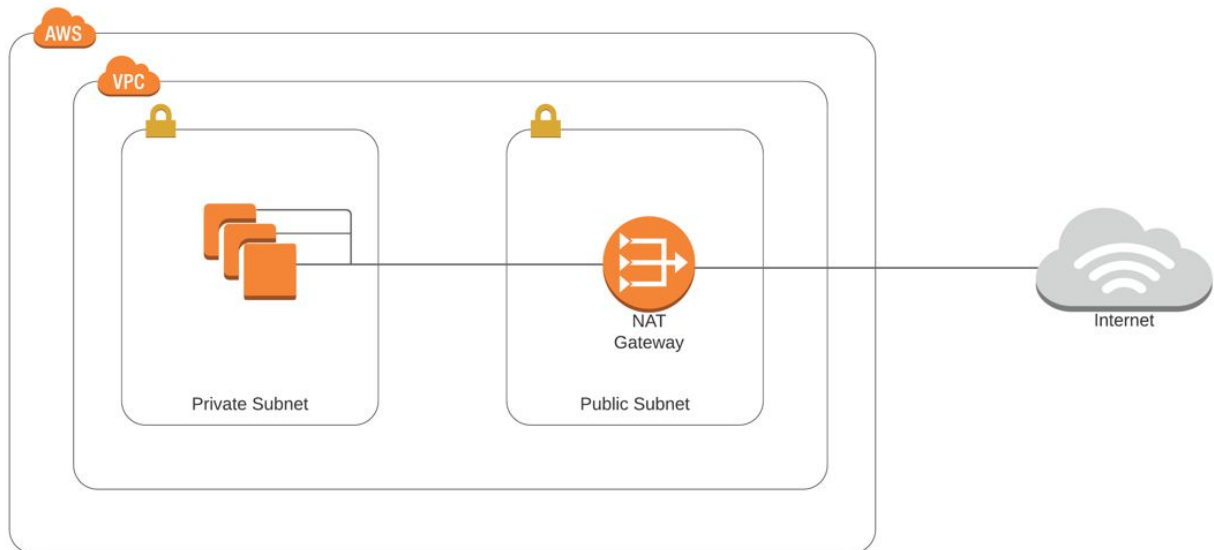


## NAT Gateway

A NAT gateway is a fully managed NAT service. The NAT gateway is highly available and redundant inside of an availability zone. A NAT gateway provides egress-only internet connectivity, similar to a NAT instance. This means that inbound connections from the internet will be refused, but internal hosts will be able to connect to the internet. A NAT gateway is optimal when hosts need to connect to the internet to download software patches but desire to keep their systems off the public internet for security reasons.<sup>34</sup>

The NAT gateway is configured in a public subnet, then a default route must be created to send internet-bound traffic to the NAT gateway. Note that the NAT gateway will use an elastic IP for the life of the NAT gateway.

The diagram below shows an example of a NAT gateway on the AWS platform.



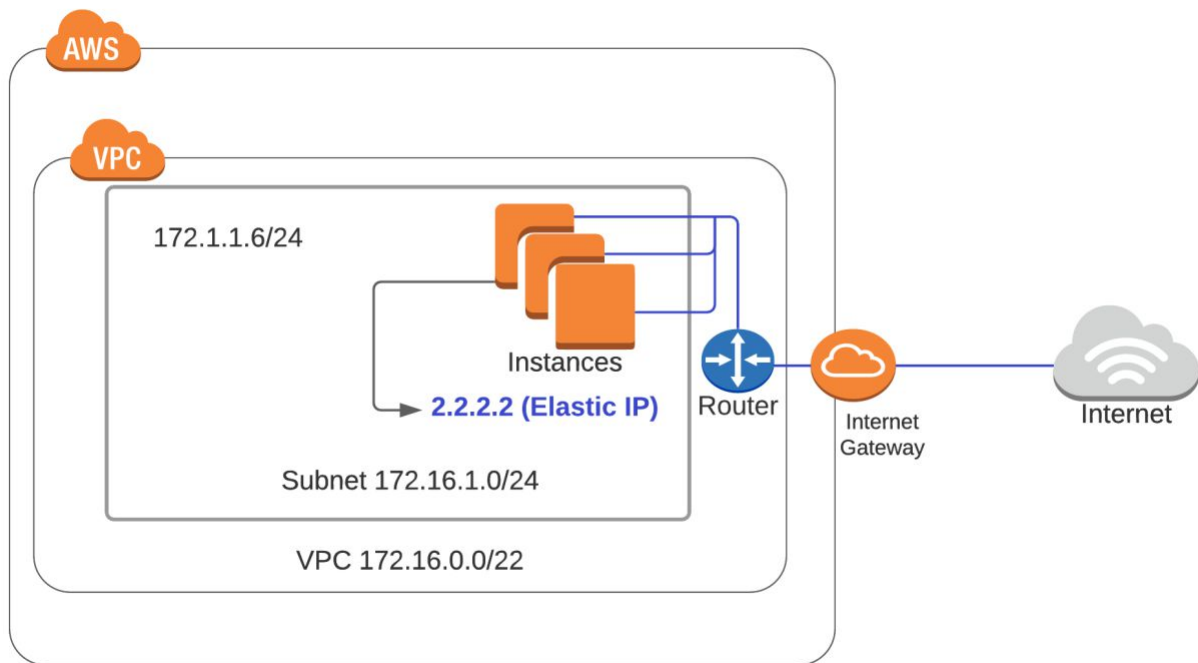
### Elastic IP Addresses (EIPs)

AWS maintains a pool of public IP addresses for their customers to use public IP addresses on the internet. These public IP addresses are called Elastic IP addresses. Elastic IP addresses work in the following manner: <sup>35</sup>

1. When an organization needs a public address, it sets up an Elastic IP address.
2. The EIP is taken from the AWS public address pool and dedicated to the customer's Elastic IP address.
3. The customer can keep this EIP as long as they are using the address.
4. When the customer no longer needs the EIP address, the customer closes the EIP, and the address is sent back to the AWS pool for future customer use.

An EIP can have a single public address that is mapped to multiple private IP addresses, with the main address being the primary address and the additional addresses being secondary addresses. Secondary IP addresses are useful during IP address migrations, as they allow for connectivity while IP addresses are changed. Secondary addresses are often used when an organization merges with another organization, and IP addresses need to be modified to allow for full connectivity. This often occurs when both organizations are using the same private IP address space.

The diagram below shows an example of using an Elastic IP address on the AWS platform.



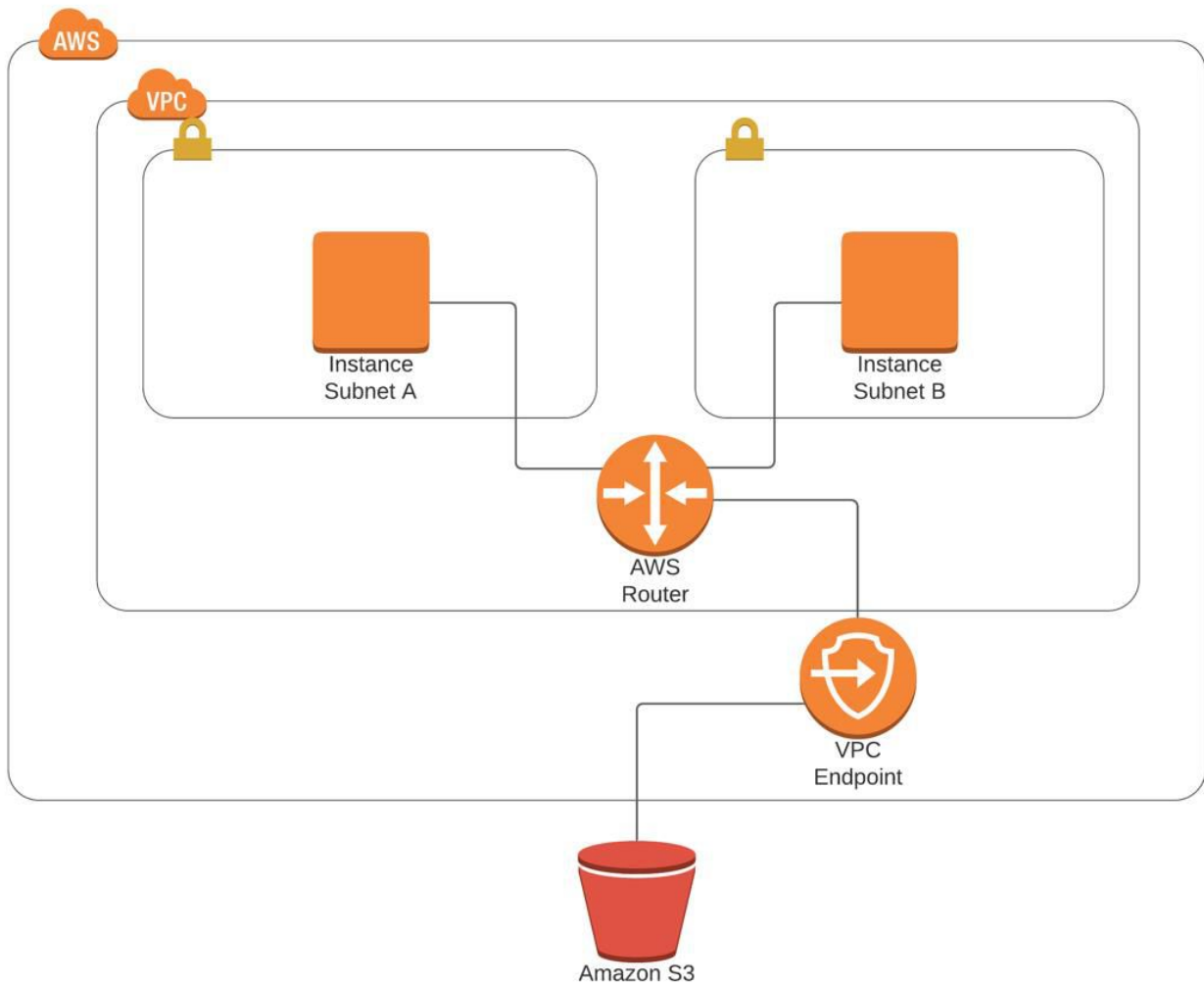
## Endpoints

Internet gateways and NAT gateways offer a solution to connect a customer VPC to the public internet. While it's possible to connect to many AWS services over the public internet, it's not the preferred method. Connecting to the AWS services over the Amazon network can offer significant benefits especially when connecting to other services or VPCs on the AWS platform. Connecting to another AWS service is performed with the creation of a VPC endpoint. When connecting with VPC endpoints, the connectivity is established over the AWS high-speed network backbone.<sup>36</sup> This can offer the benefits:

- Privacy and security – Sending data over the AWS network is much more private and secure than the internet.
- Performance – Internet gateways speeds are limited to about 45 Mbps; the AWS network has dramatically higher performance.
- The AWS network is fully managed by AWS, therefore it can have lower latency and lower congestion than the public internet. The internet has no performance guarantees across autonomous systems.
- Cost control – AWS charges for internet use. Sending data over the AWS network will cost less than internet use.
- Simplicity – VPC endpoints do not require a public IP address, internet gateway, or NAT gateway.

The way to connect your organization's VPC to AWS without traversing the internet is with a VPC endpoint. VPC endpoints are virtual devices used for routing within the AWS network. There are two types of VPC endpoints: interface endpoints and gateway endpoints.

The diagram below shows an example of a VPC endpoint on the AWS platform.



## Interface Endpoints

An interface endpoint is an elastic network interface, that uses a private address from the VPCs address pool. The interface endpoint serves as an entry point from your organization to supported services. Supported services include AWS services and other VPCs. Interface endpoints use the AWS PrivateLink service. The PrivateLink service restricts all access to between the VPC and the AWS services. Interface endpoints are compatible with most VPC services.<sup>37</sup>

## Gateway Endpoints

A gateway endpoint is a private endpoint that provides high-security access to an AWS service. It effectively places a route in the VPC's routing table for traffic destined to the AWS service. An example of a gateway endpoint is connecting Amazon S3 to DynamoDB.<sup>38</sup>

## VPC Peering

VPC peering is a technique to connect one or more VPCs without traversing the public internet. VPC peering also mitigates the need for direct or VPN connections between organizations that are hosted on the AWS network. VPC peering provides high-speed, high-availability connectivity by leveraging the AWS backbone for connectivity.<sup>39,40</sup>

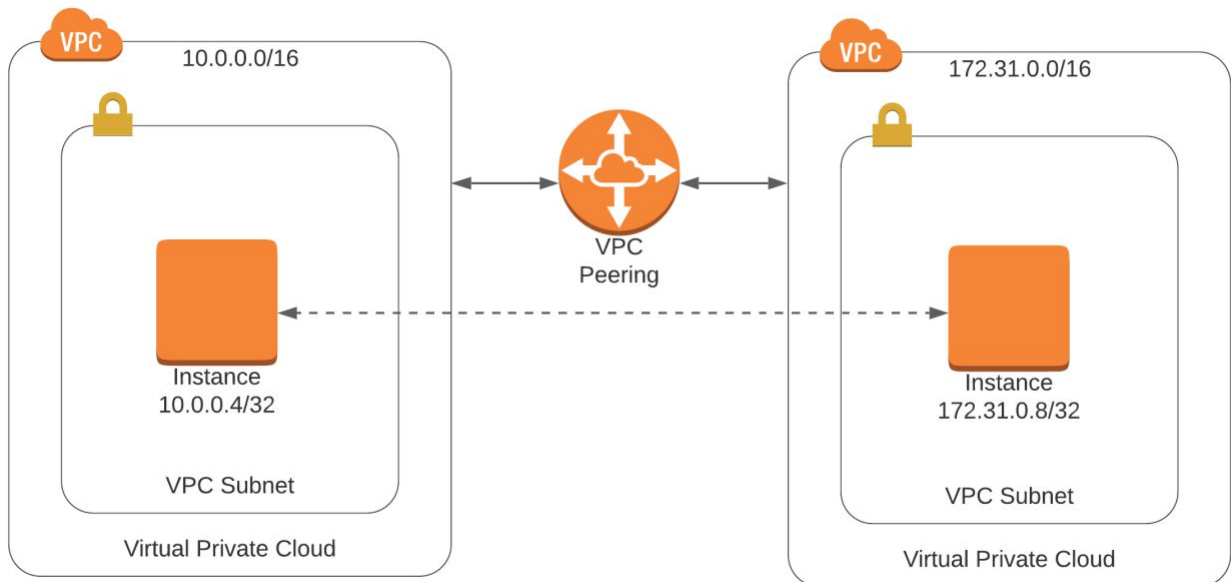
Some key things to know about VPC peering:

- VPC peering provides a nontransitive connection. This means that while VPC peering facilitates connectivity between VPCs, it does not facilitate routing traffic through a VPC to connect to another VPC.
- VPC peering uses the AWS network backbone, so there is need for internet connections, internet gateways, NAT gateways, or public IP addresses.
- Inter-region VPC traffic is encrypted for data privacy.

There are essentially two primary architectures for VPC peering: hub and spoke, and fully meshed.



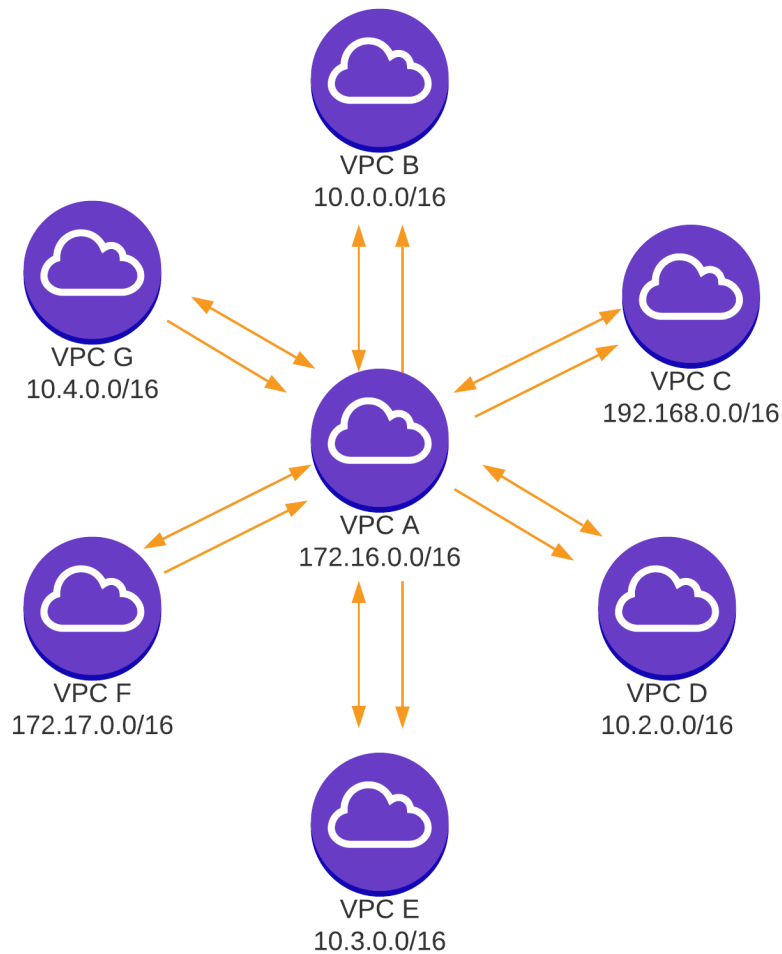
The diagram below shows an example of a VPC peering on the AWS platform.



### Hub and Spoke

In a hub-and-spoke environment, a hub is created with connections to all remote VPCs. This enables the hub to communicate with each remote VPC or spoke. However, since VPC peering is not transitive, VPCs will not be able to communicate with each other since communication is limited to hub-and-spoke VPCs.

The diagram below shows an example of hub-and-spoke VPC peering on the AWS platform.



### Fully Meshed

When all VPCs need to communicate with each other, every VPC will require peering to every other VPC. This is referred to as a fully meshed environment. Fully meshed architectures are required because VPC peering is not transitive. While fully meshed VPC peering works well, it is challenging from a scalability perspective. Scalability is challenged as the number of connections grows very rapidly as additional sites are added. The number of connections can be calculated with this formula:  $N * (N - 1) / 2$ , where N is the number of nodes or VPCs.

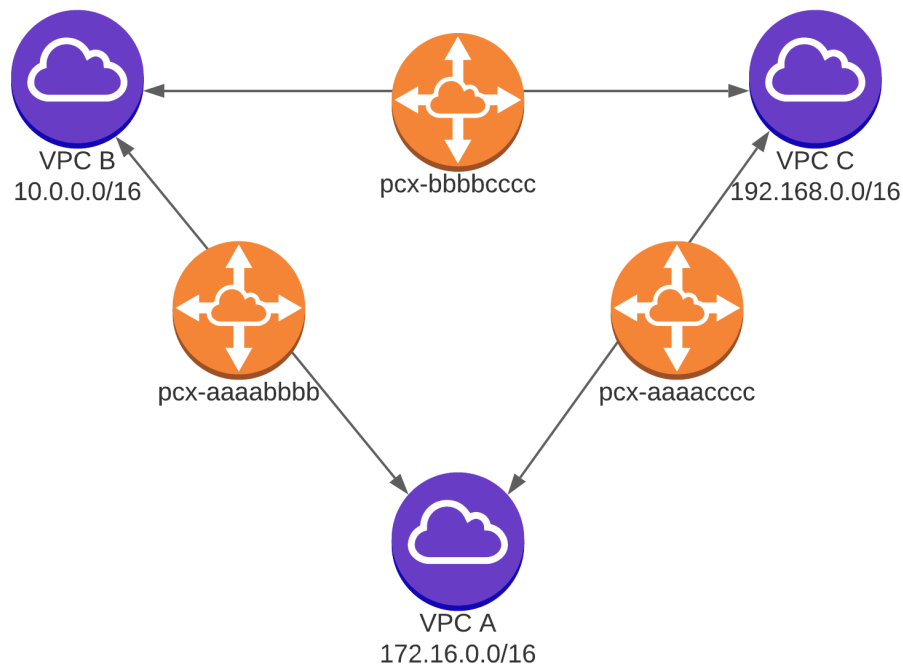
Let's examine how fast VPC connections grow in the fully meshed environment.

- With three VPCs, six connections are required:
  - $3 * (3 - 1) / 2 = 6$  connections
- With ten VPCs, forty-five connections are required:

- $10 * (10-1) / 2 = 45$  connections
- With twenty VPCs, only 190 connections are required:
  - $20 * (20-1) / 2 = 190$  connections
- With thirty VPCs, 435 connections are required:
  - $30 * (30-1) / 2 = 435$  connections

Fully meshed environments are excellent when a small number of VPCs require connectivity. However, fully meshed environments do not scale when a large number of VPCs require connectivity with each other. A solution to help create a more scalable VPC peering environment is with AWS CloudHub.

The diagram below shows an example of fully meshed VPC peering on the AWS platform.

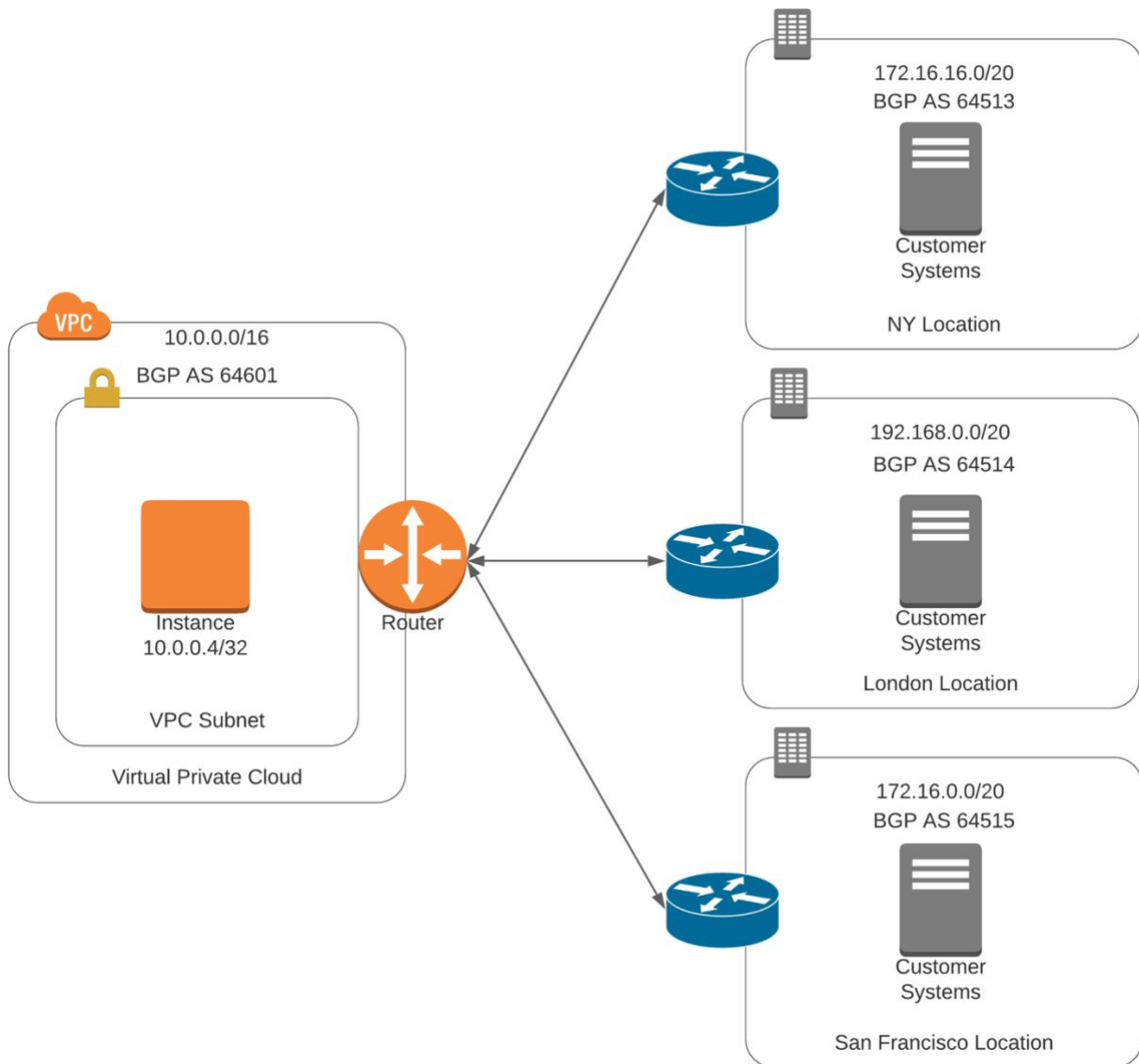


## AWS CloudHub

When it's necessary to establish connectivity between a VPCs and a large number of remote sites, CloudHub simplifies the process of VPC peering. CloudHub enables an organization to have transitive VPC connections in a hub-and-spoke environment. CloudHub uses BGP, specifically eBGP, to connect and share routing information across VPCs. Routing information is propagated via BGP, which provides network reachability to all remote locations or connected VPCs. Since all remote locations have knowledge of all subnets across organizations and VPCs,

full communication across VPCs is established. Since BGP is used for route sharing, connectivity can be limited to only desired resources by using route filters, access lists, and firewalls.<sup>41</sup>

The diagram below shows an example of using CloudHub to simplify VPC peering on the AWS platform.



## Enhancing the VPC Security Posture

AWS provides numerous options to enhance the security of a VPC. Two fundamental options to enhance the security of a VPC include network access control lists and security groups.

## Network Access Control Lists (NACL)

The NACL is a means to enhance security by keeping unwanted traffic out of a subnet. NACLs block or permit traffic in a manner similar to an access control list on a router or a stateless firewall.<sup>42</sup> There are some key things to understand about NACLs in order to use them effectively:

- Rules are created to determine what traffic is allowed or denied.
- The accept or deny rules must be written in a specific order.
- The rules are processed in order, therefore if you explicitly deny something, it won't be possible to permit something that is denied by a previous statement.
- The order is determined by the number attached to the rule statement.
- Lower numbers in the rule statement are processed prior to higher numbers.
- Network ACLs have an implicit deny, so you must specify any traffic that you want to allow, or it will all be blocked when you add a network ACL. The allowed traffic must be sequenced before any rule that denies the desired traffic.
- NACLs are written in both inbound and outbound rules. Inbound rules determine what's allowed into a subnet; outbound rules determine what traffic is permitted to leave a subnet.
- Remember, NACLs are stateless, so inbound rules and outbound rules need to match. If SSL is allowed in, then it must be allowed back to the requester. NACLs are stateless, therefore there is no means to allow return traffic like a stateful firewall.
- NACLs have an implicit deny, so all that is necessary to permit desired traffic.

The order of the rules in the NACL are so important. Below are two examples of network ACLs. One with the proper technique, and the other with an incorrect technique.

### Proper NACL Structure – Do This!

Proper Technique that allows desired traffic:

#### Inbound

Rule 110 Allow TCP Port 80 Source any

#### Outbound

Rule 110 Allow TCP Port 80 Destination any

### Improper NACL Structure – Don't do this!

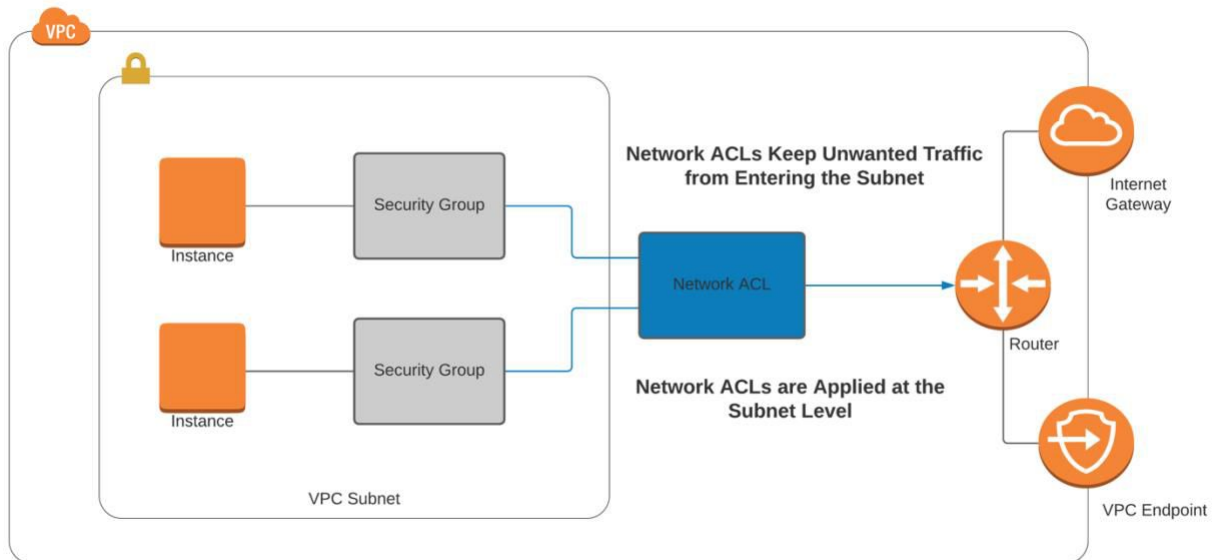
#### Inbound

Rule 100 – Deny all traffic

Rule 110 Allow TCP Port 80 Source any

Note that in the above example with improper technique, all traffic is blocked by the first rule in the NACL, therefore all traffic is blocked. This reinforces the need to use the correct order in NACL rule statements.

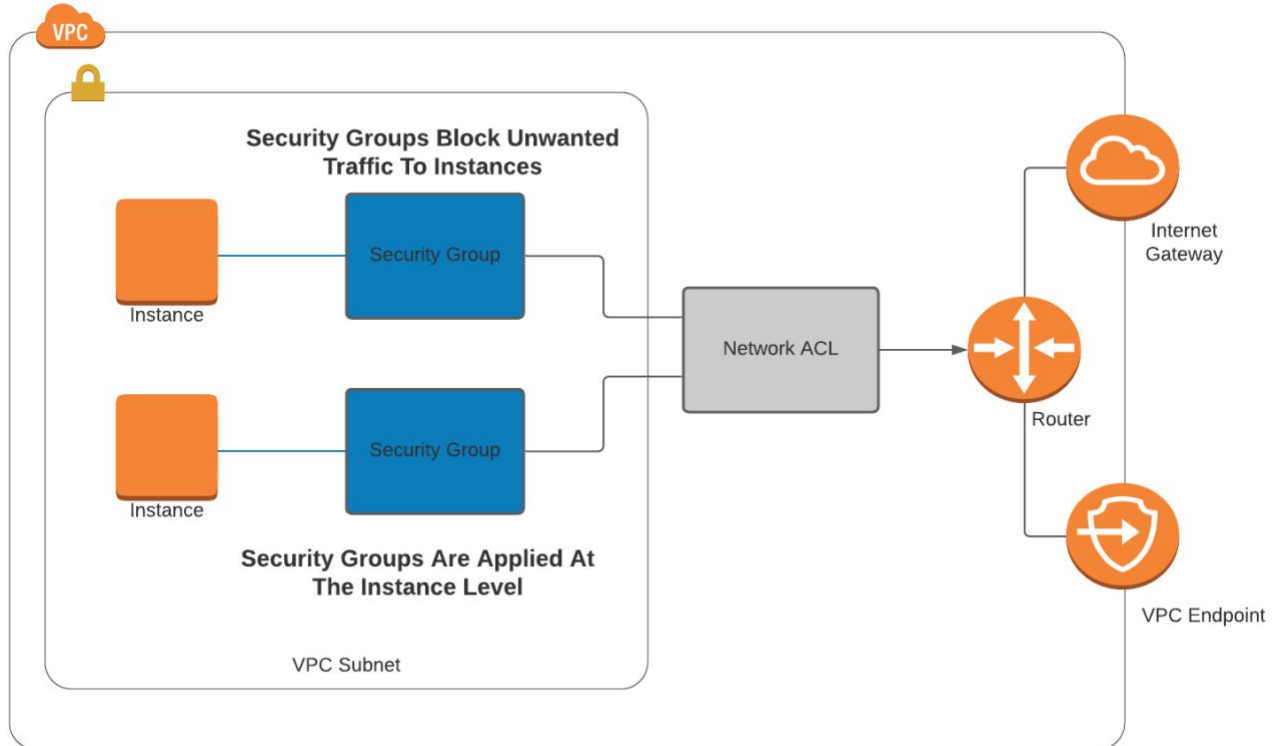
The diagram below shows how network ACLs keep unwanted traffic out of the subnet.



## Security Groups

A security group is essentially a stateful access control list (like a firewall) that is applied to a computing instance or AWS service. This is different than a NACL, which is applied to a subnet. Realistically speaking, a good security architecture will include NACLs at the subnet and security groups attached to the server. Security groups have an implicit deny, so only permit statements are required. All that is necessary is configuring the permit statements to allow desired traffic into the server. Since security groups are stateful, it is only necessary to permit inbound traffic, as outbound return traffic will be permitted. Security groups evaluate all rules prior to permitting or denying traffic, so the order of rules in a security group is not as critical as with NACLs.<sup>43</sup>

The diagram below shows how security groups keep unwanted traffic out of the instance.



## Labs

- 1) Create a VPC using private address space. Use 10.0.0.0/16 as block of IP addresses. Link to create a VPC below.  
<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-getting-started.html>
- 2) Create a new private subnet 10.0.1.0/24. Link on how to create a subnet below.  
<https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#AddaSubnet>
- 3) Create an EC2 Instance. Create an Elastic Network Interface (ENI). Attach the elastic network interface to the newly created instance. Link to create an ENI and attach to an instance below.  
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html>
- 4) Create an Internet gateway and set up routing for the internet. After the internet gateway is created set up a default route pointing the internet gateway. Link on how to create an internet gateway below.  
[https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_Internet\\_Gateway.html#Add\\_IGW\\_Attach\\_Gateway](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html#Add_IGW_Attach_Gateway)

- 5) Create a network ACL. Set up the NACL to block any traffic coming from the CIDR range 192.168.0.0/16 from entering the subnet. Prevent any traffic from leaving the subnet destined to the CIDR range 192.168.0.0/16. The NACL should allow all other traffic in and out of the subnet. Link on how to set up a NACL below.  
<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-basics>
- 6) Create a security group for the EC2 instance you created. Allow http, https traffic from any source into the instance. Link on how to create security group below.  
[https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html#CreatingSecurityGroups](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html#CreatingSecurityGroups)
- 7) Set up a NAT gateway. Set up a default route to the NAT gateway. Prior to setting up the NAT gateway delete the current internet gateway and current default route pointing to the internet gateway. Link on how to create the internet gateway below.  
<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html#nat-gateway-creating>



## Chapter 6

### AWS Network Performance Optimizations

There are three components of AWS networking that can have a substantial impact on performance and availability: placement groups, Route 53, and load balancers.

#### Placement Groups

Placement groups are simply where an organization places their equipment. Where an organization places its equipment can have a profound effect on performance and availability. There are three options for placement groups: clustered, partitioned, and spread groups.<sup>44</sup>

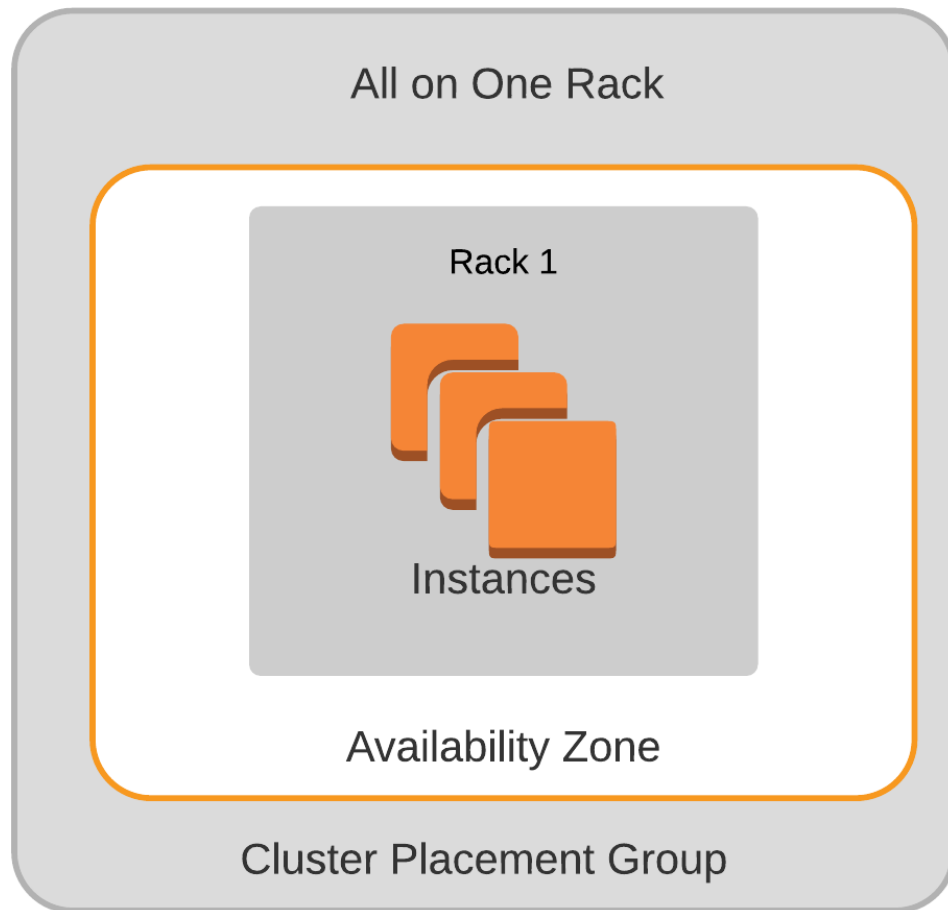
#### Clustered Placement Groups

A clustered placement group offers the best performance at the expense of availability. A clustered placement group means placing an organization's servers extremely close to each other to reduce latency and optimize performance.<sup>45</sup> Some key tenants of placement groups are:

- Proximity – Instances are very close in physical proximity.
- Instances are often in the same rack.
- Instances are often on the same physical server.
- Since devices are often on the same rack, on the same network switch, and on the same server, this offers the absolute best network performance

Clustered placement groups have a major drawback when it comes to availability. Since everything is close together, often in the same server, rack, switch, and power source. there are many single points of failure when compared with other architectures. This architecture is perfect for applications that are not tolerant of latency.

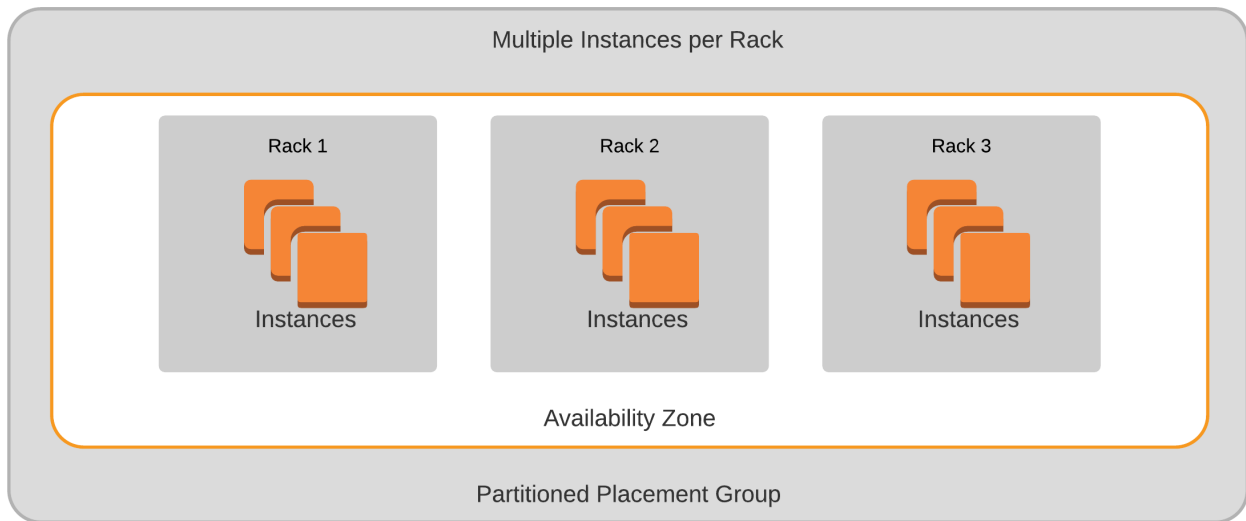
The diagram below shows an example of a clustered placement group.



### Partitioned Placement Groups

Partitioned placement groups provide a high level of performance, low latency, and with higher availability than a clustered placement group. Partitioned placement groups place all components in a single AZ (data center). However, the instances are grouped into partitions and spread across data center racks. Spreading the load across a data center minimizes the risk that a single server, power outage, or network switch failure will bring the entire system down.<sup>46</sup>

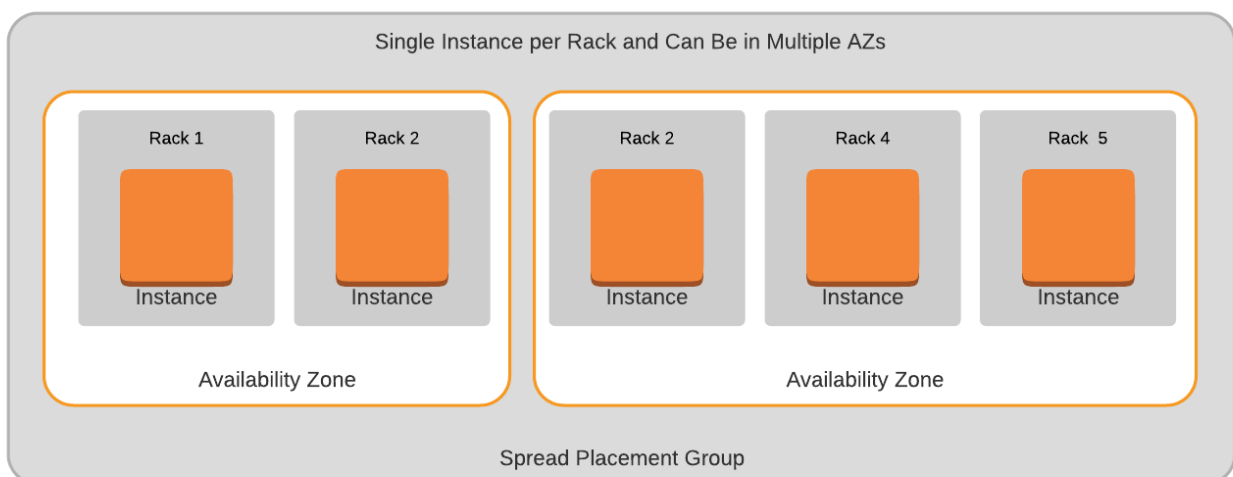
The diagram below shows an example of a partitioned placement group.



### Spread Placement Groups

A spread placement group is optimal when a high-availability design is required. In spread placement groups, instances are spread across hardware, racks, physical servers, power distribution units, and other system components. Groups can be spread across multiple availability zones. This design offers high availability, but with higher latency and lower network performance than clustered and partitioned placement groups.<sup>47</sup>

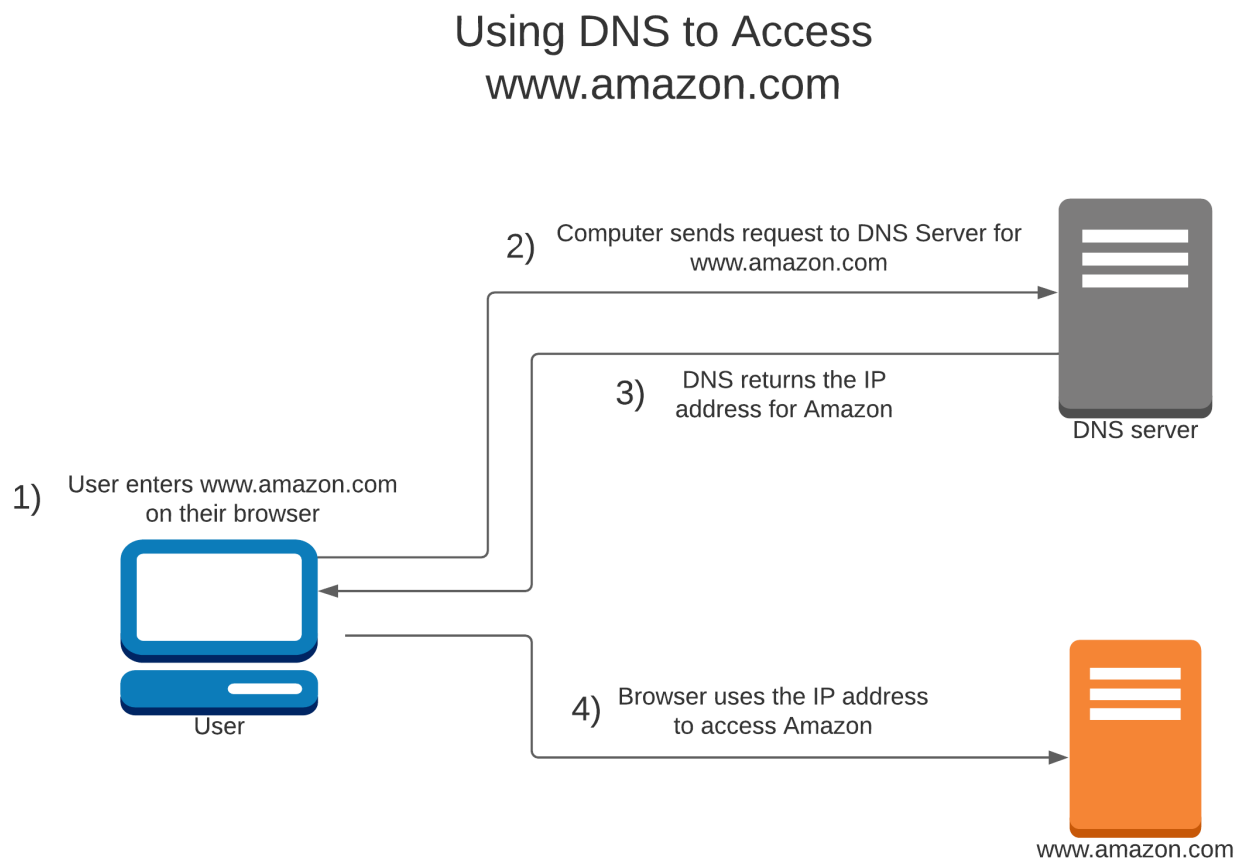
The diagram below shows an example of a spread placement group.



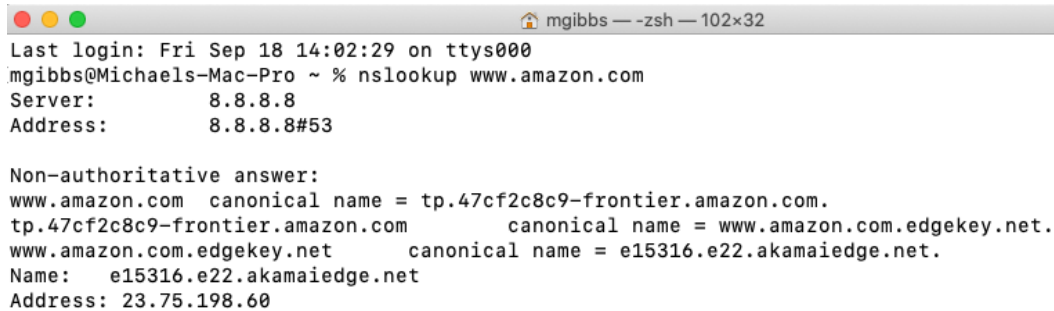
## Amazon Route 53

Every device on the network requires an IP address. While it's possible to connect directly to every system using just the device's IP address, it's infeasible to remember every system's IP address. Imagine remembering the IP address to every website on the internet. The solution that was developed to this challenge was the Domain Name System (DNS). The DNS system maps a name with an address. Amazon Route 53 is Amazon's implementation of DNS system.<sup>48</sup>

The diagram below shows an example of DNS mapping a name to an IP address.



For example, it's very easy to remember [www.amazon.com](http://www.amazon.com). When we enter [www.amazon.com](http://www.amazon.com), our systems ask the DNS server what the IP address is for this website. Then our request is forward to the IP address of the Amazon website. You can see the name to address mapping on any UNIX/Linux and Windows systems with the nslookup command.

A terminal window titled 'mgibbs — zsh — 102x32' showing the output of the 'nslookup www.amazon.com' command. The output displays the server used (8.8.8.8), the address (8.8.8.8#53), and a non-authoritative answer showing the canonical names and IP addresses for www.amazon.com, tp.47cf2c8c9-frontier.amazon.com, and www.amazon.com.edgekey.net.

```
Last login: Fri Sep 18 14:02:29 on ttys000
mgibbs@Michaels-Mac-Pro ~ % nslookup www.amazon.com
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
www.amazon.com canonical name = tp.47cf2c8c9-frontier.amazon.com.
tp.47cf2c8c9-frontier.amazon.com canonical name = www.amazon.com.edgekey.net.
www.amazon.com.edgekey.net canonical name = e15316.e22.akamaiedge.net.
Name:   e15316.e22.akamaiedge.net
Address: 23.75.198.60
```

Some key points to know about AWS Route 53:

- Route 53 provides name to IP address mappings just like any other DNS platform.
- Route 53 is a high-availability platform for DNS services.
- Route 53 is highly scalable platform for DNS services and server health checks.
- AWS uses anycast services, for which there are multiple servers with the same address placed over the internet.
- Anycast provides extremely high availability and low latency. As a host it will connect to the closest DNS server based upon its IP address. If a DNS server were to become unavailable, the host will connect to the next closest Anycast address of the DNS server
- Route 53 supports most of the available DNS record types.
- Route 53 uses TCP and UDP port 53.
- Route 53 works with health checks and can be used to create a high-availability solution.
- Route 53 supports most DNS record types.
- Route 53 has numerous options to optimize a web-based environment.

AWS supports the following DNS record types:

Amazon Route 53 Supported Record Types
A (address record)
AAAA (IPv6 address record)
CNAME (canonical name record)
CAA (certification authority authorization)
MX (mail exchange record)
NAPTR (name authority pointer record)
NS (name server record)
SOA (start of authority record)
SPF (sender policy framework)
SRV (service locator)
TXT (text record)

While a deep understanding of DNS is beyond the scope of this book, there are some DNS records that are so foundational we recommend learning them. These key record types are:

#### A – Record

- The most fundamental DNS record.
- A record mapping a name to an IP address.
- The IPv6 equivalent is an AAAA record.

#### CNAME – Record

- A record that maps a domain to another domain.
- It can map to another CNAME Record or an A record.
- CNAME records effectively redirect a request for one domain to another domain.
- i.e., map [www.a.com](http://www.a.com) to [www.b.com](http://www.b.com).

#### NS – Record

- Identifies the DNS servers that are responsible for your DNS zone.
- These authoritative name servers propagate an organization's official DNS information to the DNS servers across the internet.
- NS records can be several entities.

## MX – Record

- A MX record specifies which mail servers can accept mail for your domain.
- MX records are necessary to be able to receive email.

AWS has several policy-based routing options for Route 53. These policies can be as simple as a mapping a name to IP address to finding the server with the lowest latency. The AWS Route 53 policies and their functions are:

- Simple routing – Basic DNS that maps a domain name to a single location. This is the default policy, which is perfect with a single server for a domain.
- Failover routing – Sends the traffic to the main server. If that is not available, sends traffic to a backup server.
- Geolocation routing – Used when there are servers in several regions. To optimize performance, geolocation routing will look at the source IP address of the user (which will ultimately provide their location) and route them to the closest region so they have the best performance.
- Geoproximity routing – Used when an organization has servers in multiple availability zones. Geoproximity routing will send the requestor to the closest availability zone.
- Latency-based routing – Will send to the server with the lowest latency to optimize performance. Ideal when the website is in multiple availability zones or regions. It provides the optimal experience to the user.
- Multivalue answer – Route to any available server.
- Weighted – Provides a mean share traffic between servers at a percentage you chose, i.e., 75 percent server a, and 25 percent to server b. Great option to test an application functionality. Think of a CI/CD pipeline and blue green deployments. Send most of the traffic to the old website for testing and a percentage to the new website. When the new site is tested, move the traffic to the new server.

## Load Balancers

Load balancers are network devices that facilitate load sharing across servers. Load balancers can help greatly with scalability by allowing the application to be deployed across multiple servers. Load balancers can also increase availability by allowing multiple servers to be used simultaneously, removing single points of failure. Additionally, load balancers can use health checks to remove unhealthy servers from being used, which further enhances application availability.<sup>49</sup>

Since load balancers facilitate load sharing among servers, they can facilitate scaling out applications across multiple servers. Scaling out servers substantially increases performance.

There are essentially two kinds of load balancers: network load balancers and application load balancers.

### **Network Load Balancers**

Network load balancers operate at the transport layer (layer 4) of the OSI model. Network load balancers work with either the TCP or UDP transport protocol.

### **Application Load Balancers**

Application load balancers operate at the application layer (layer 7) of the OSI model. Application load balancers typically work with the HTTP or HTTPS protocol.<sup>50</sup>

The AWS implementation of a load balancer is called an elastic load balancer. There are three options for elastic load balancers. These options are application load balancers, network load balancers, and classic load balancers. There are some key things to know about elastic load balancers:

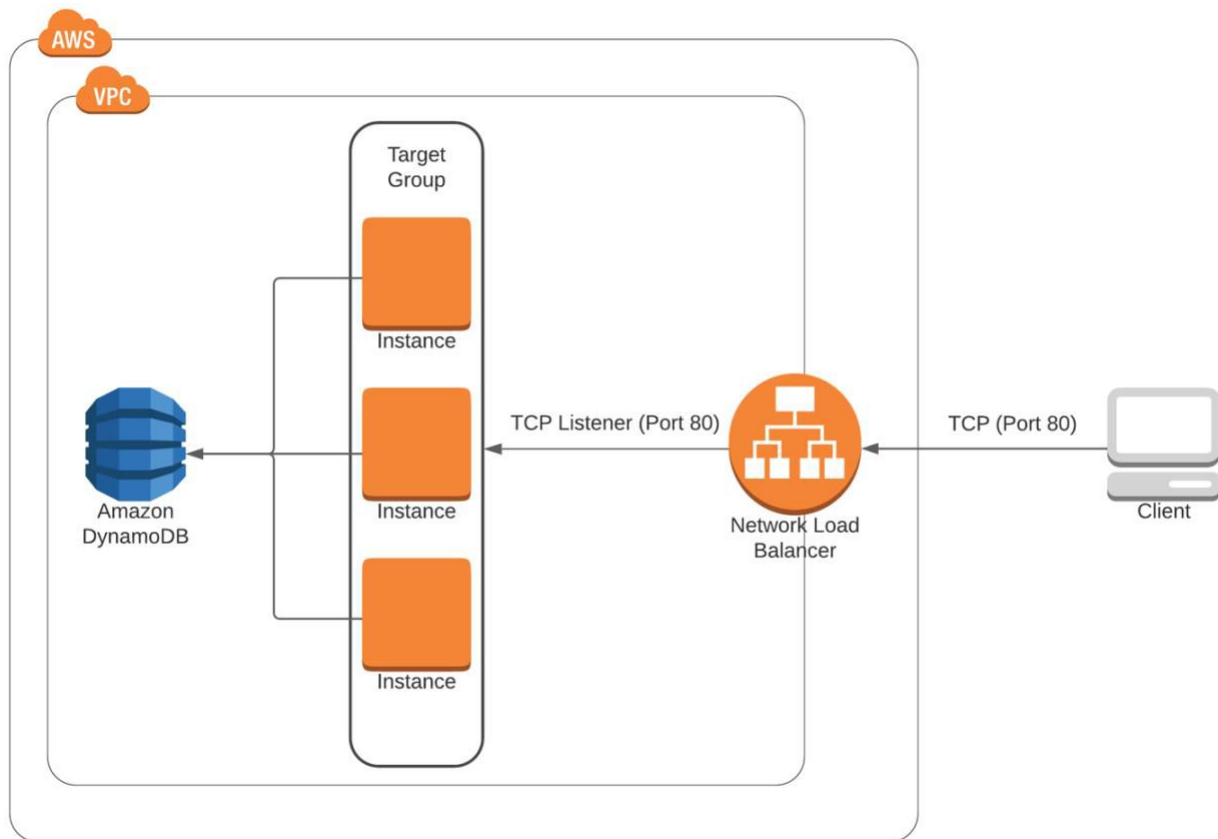
- Elastic load balancers automatically distribute traffic to multiple targets (i.e., EC2 instances).
- Elastic load balancers are autoscaling and can spin up more instances if necessary to meet the applications performance needs.
- Elastic load balancers use an IP address, and if autoscaling occurs, multiple IP addresses will be used.
  - Plan your addressing scheme carefully so you don't run out of addresses.
  - Remember, AWS reserves the first four IP addresses (network and first three usable addresses) and the last (broadcast) IP address of the subnet.
- Elastic load balancers can load balance across availability zones.
- Elastic load balancers support health checks so nonfunctioning servers can be removed from use.
- Elastic load balancers can terminate SSL connections, which can reduce load on servers.

### **Elastic Load Balancer – Network**

The network elastic load balancer operates at layer 4. Network load balancers perform routing based upon the destination port of the traffic they receive. Network load balancers are extremely fast and are an excellent option when ultimate speed is needed. Network load balancers can handle millions of requests per second and excel with rapidly changing traffic patterns. The elastic load balancer network version is stateful. This means that once a connection is established between the host and the server, the connection is maintained until the session is completed. The network ELB keeps state by maintaining sticky sessions that have a mapping of the source and destination of the connections.<sup>51</sup>



The diagram below shows an example of an elastic network load balancer.



### Elastic Load Balancer – Application

The application elastic load balancer operates at layer 7 of the OSI model. The application ELB can route traffic based upon many options and is ideal for web traffic. Application load balancers are an excellent means to load balance requests to microservices and container-based applications. Application ELBs can even load balance between a VPC and an on-premises data center.

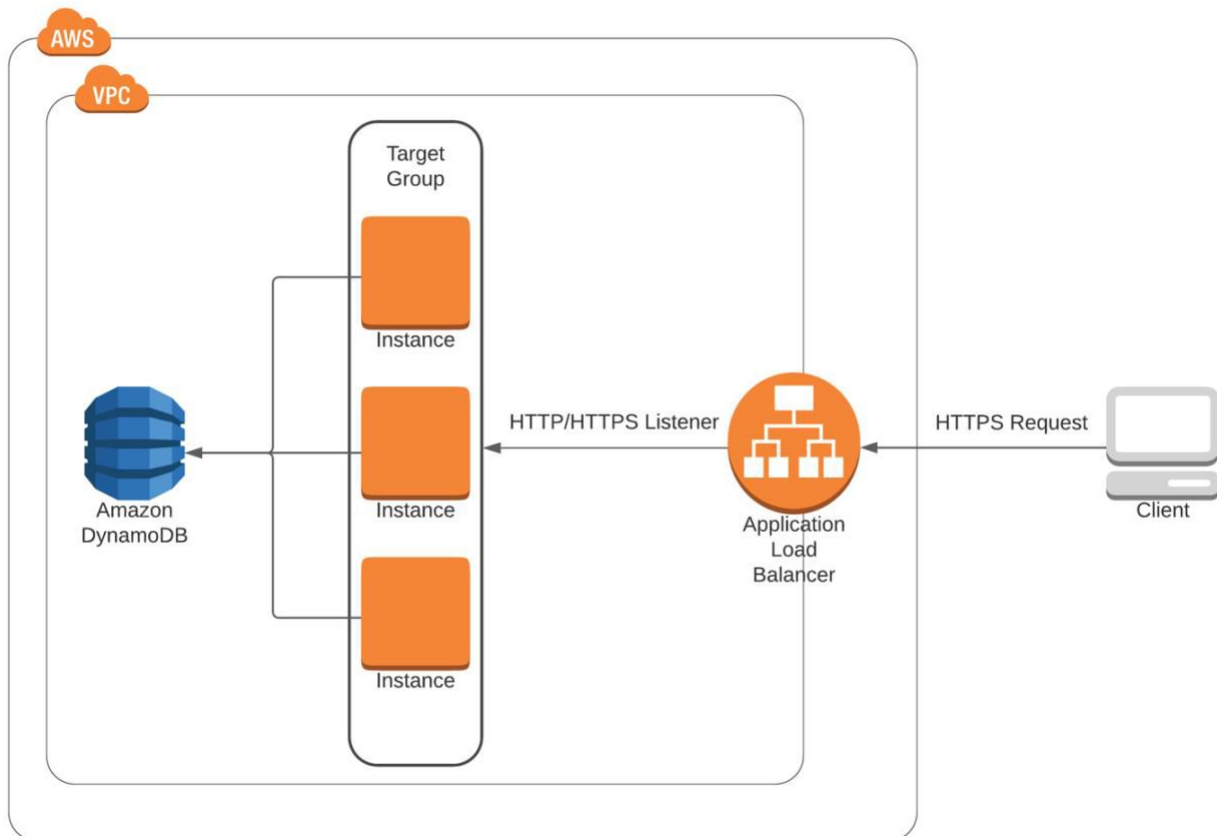
Application elastic load balancers are stateful. Once a connection is established between the host and the server, the connection is maintained until the session is completed. The application ELB keeps state by maintaining sticky sessions that have a mapping of the source and destination addresses of the connections.

Application ELBs make forwarding decisions based upon the following factors:

- Domain name
- Path provided by the URL

- Elements in the http, https header
- HTTP method-based routing (i.e., put or get)
- Source address

The diagram below shows an example of an elastic application load balancer.



## Classic Load Balancers

The AWS classic load balancer can be network based or application based. This is a legacy platform and can work with both EC2-classic and VPCs.<sup>52</sup> Classic load balancers:

- Have autoscaling capabilities.
- Can support single or multiple availability zones.
- Can terminate SSL connections to reduce server load.
- Are stateful by using sticky sessions.
- Provide logs to analyze traffic flows and can be used with CloudTrail for auditing.

## Labs

- 1) Create a Centos based EC2 instance. Install an apache web server. Place an index.html file in the webserver as the main web page. Link on how to install a web server on Centos is below.  
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html>
- 2) Create a hosted zone in route 53 pointing to the EC2 server you created. Link on how to set up a hosted zone in route 53 below.  
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/CreatingHostedZone.html>
- 3) Set up a Route 53 health check on the web server you created. Link to set up a Route 53 health check is below.  
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html>
- 4) Create an AMI of the webserver you just created. Launch a new instance from this AMI. Link on how to create AMI below.  
<https://docs.aws.amazon.com/toolkit-for-visual-studio/latest/user-guide/tkv-create-ami-from-instance.html>
- 5) Set up a route 53 failover routing policy with health checks on both EC2 instances. Then shut off one of the web servers and make sure DNS fails over. Link on how to set up failover routing policy below.  
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-configuring.html>
- 6) Shut down and delete all instances and services created in this lab.

## Chapter 7

### Security

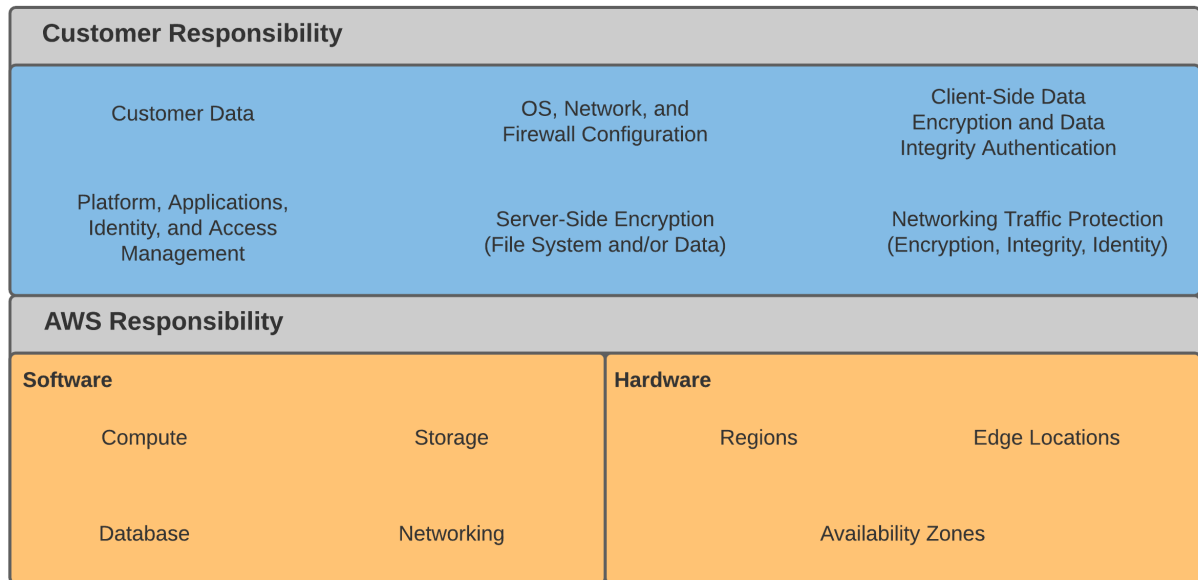
Security is a critical component for the success of any organization. A complete guide to full security architectures is beyond the scope of this book and should be explored by any organization that would face serious consequences if a security breach occurs. This chapter breaks down security into the following components:

- Who is responsible for what parts of the VPC?
- Principle of least privilege.
- Industry compliance.
- Identity and access management.
- Multiple account strategies.
- Network ACLs, security groups, WAF.
- Intrusion detection and prevention
- Distributed denial of service attacks and prevention.
- Service catalogs.
- Systems manager parameter store.

### AWS Shared Security Model

While outsourcing the data center to the cloud can help with costs, agility, scalability, and even security, there is still a substantial amount of security that must be performed by the customer. When outsourcing an organization's data center to AWS, security and compliance responsibilities are shared between AWS and the customer. This is called the shared security model. In the shared security model, AWS maintains the security of the cloud, and the customer maintains the security of their VPC.<sup>53</sup>

The diagram below shows the AWS shared responsibility model.



## Securing the Cloud

AWS manages keeping the cloud secure. Keeping the cloud secure is really about managing the following functions:

- Physical security – Keeping the facility locked, keeping unauthorized users out of the AWS data centers.
- Principle of least privilege – Limiting who from AWS can manage assets in the cloud.
- Security of the cloud – Keeping the cloud secure (firewalls, system patching, routing, IDS/IPS, change management).
- Keeping all AWS applications secure with patching and maintaining the underlying components of serverless applications offered by AWS.
- Keeping the AWS network secure with secure routing, VLANs, route filtering, firewalls, and intrusion prevention and detection (IDS/IPS).

## Securing the VPC

The customer is responsible for securing all aspects of their VPC. This means the customer is responsible for the following security components:

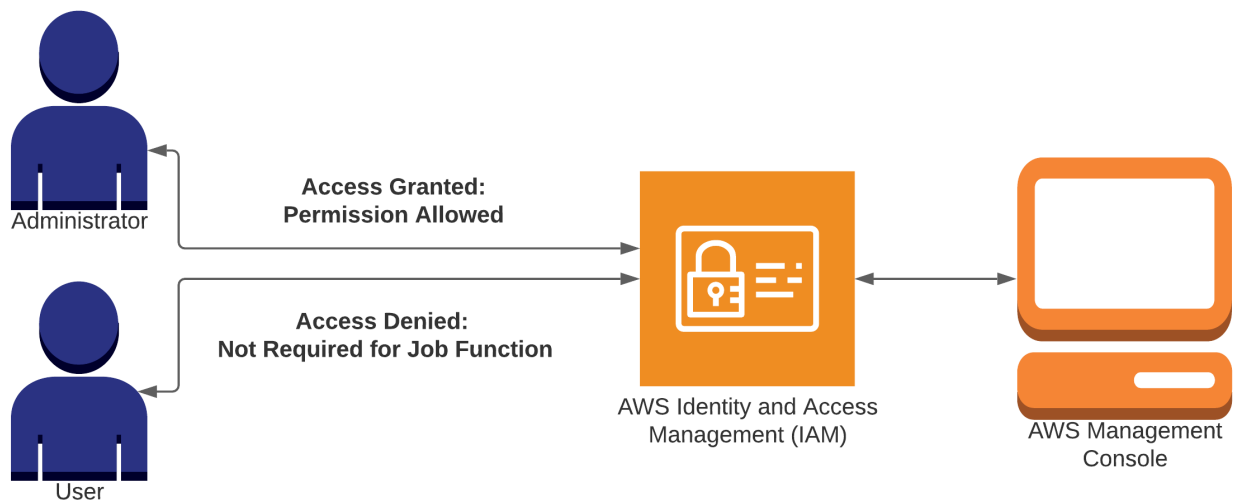
- Identity and access management – Determine who is allowed in the VPC and define their functions.

- Principle of least privilege – Grant the least privileges necessary for employees and partners to perform their functions effectively.
- Data security – Manage encryption.
- Maintenance of customer-designed applications.
- Management of the VPC routing tables.
- Managing traffic allowed into the VPC – Firewalls, NACLs, security groups.
- Maintenance of the operating systems and applications stored on EC2 compute instances.
- Physical security – Keep the devices that connect to the cloud secure from unauthorized users.

### Principle of Least Privilege

One of the most critical components of security is the principle of least privilege. The principle of least privilege is really about making sure individuals and systems using the cloud can access only the functions necessary to perform their role effectively. Granting more than the minimal level of privileges can enable users or hackers to intentionally or accidentally damage the VPC. Additionally, privileges should be revoked when no longer needed, i.e., when an employee leaves the company.<sup>54</sup>

The diagram below shows the principle of least privilege by allowing access to the management console only to individuals who need access for their job function.



## Industry Compliance

Many industries throughout the world are highly regulated. Often these industries have a legal requirement that requires a level of security, data retention, and auditing policies. AWS supports many international compliance requirements.<sup>55</sup>

Some key compliance standards are:

- PCI DSS – for payment cards
- ISO 9001. 27001, 27017, 27018
- Fed ramp
- HIPAA – US health care privacy

A full list can be seen at <https://aws.amazon.com/compliance/programs/>.

## Identity and Access Management

Identity and access management is a key component of any security architecture. Identity and access management is about identifying the user and giving the user access to the resources necessary to perform their functions.<sup>56</sup> Identity and access management is also referred to as AAA. The key components of AAA:

- Authentication – Identify the user.
- Authorization – Determine if the user is allowed to access the resource.
- Accounting – The ability to see what the user has done.

AWS divides IAM into users and roles. An IAM user is a person accessing the AWS cloud. Generally speaking, an IAM role is used by an AWS service to access another service, i.e., EC2 accessing a DynamoDB.

AWS has some specific components of its IAM systems. AWS uses the concept of principals. In AWS a principal is an IAM entity that is permitted to access AWS resources. AWS further breaks down the principal concept into root users, IAM users, and roles.

The diagram below shows the functions of authentication, authorization, and accounting.

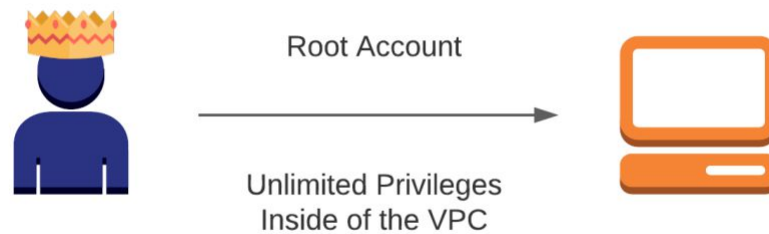


### Root User

The root user is the person who created the AWS account. The root user has full system access. The root user can access the console and has programmatic access to AWS resources. Since the root user can do anything, including deletion of the VPC, it's best to use the root account to set up the VPC and then immediately create an IAM user with appropriate access to the VPC. This is similar in practice as not using the root account to log in to a UNIX or Linux system to prevent accidental system damage.



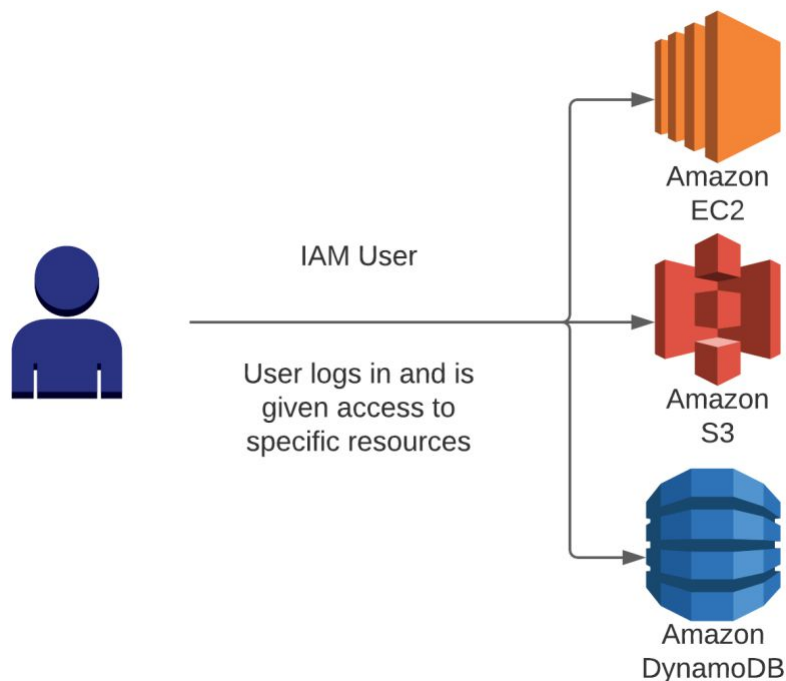
The diagram below shows an example of the root user privileges.



## IAM Users

IAM users are identities that have permissions to interact with AWS resources. IAM users are created by principals with administrative access. IAM users can be created with the AWS management console, CLI, or SDKs. IAM users are permanent unless deleted by an administrator.

The diagram below shows provides an IAM user accessing an AWS VPC.



## Roles and Security Tokens

Roles are used to provide access to AWS services. There are three types of roles in the AWS environment:

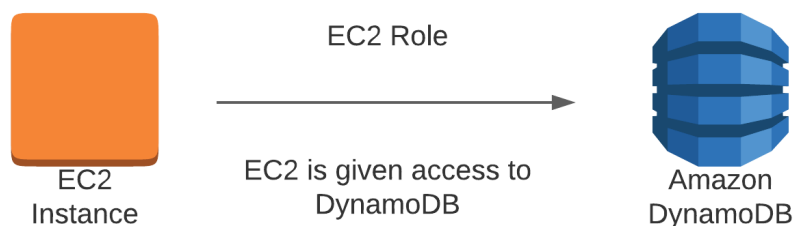
- EC2 roles
- Cross account roles
- Identity federations

### EC2 Roles

EC2 roles enable EC2 computing instances to access AWS services, i.e., S3 and DynamoDB. To set up an EC2 Role, an IAM role is created and then applied to the EC2 instance. By creating the EC2 role, there is no need to store AWS credentials on the EC2 instance, which further enhances security.<sup>57</sup> Here is how EC2 roles work:

1. An EC2 role is created.
2. The EC2 role is applied to an EC2 instance.
3. When the EC2 instance attempts access AWS services, a temporary token is provided to allow access.
4. The AWS service recognizes the tokens and grants access.
5. As temporary tokens expire, new tokens are generated frequently.
6. By rotating tokens, security is enhanced as no password (key) needs to be passed to the application.
7. This greatly enhances security. If an EC2 instance were to be hacked, no passwords are given to the hacker. Since the tokens expire and are rotated frequently, even if hackers were to gain access to a token, it could not be used for long.

The diagram below shows an example of an EC2 instance accessing DynamoDB with an EC2 role.



## Cross-Account Roles

In the modern technology environment, it is frequently necessary for an organization to share resources with other business partners. In order to connect with organizations outside the VPC, cross-account roles are used. Connecting to other organizations can create significant business opportunities, but that connectivity also brings security challenges. The partner company may need access to certain resources but should not have access to any resource that could compromise the organization if lost or stolen. While it's always essential to provide access with the principle of least privilege, nowhere is it more critical than with connecting to external organizations.<sup>58,59</sup> Therefore, be very strategic in assigning permissions to cross-account roles. Cross-account roles work in the following manner:

1. A role is created for the external user
2. The external user connects to the AWS Secure Token Service (STS) and receives a temporary token.
3. The external user then provides the temporary token to AWS and is authorized to access the VPC.

The diagram below shows a cross-account role being used to access external VPCs.

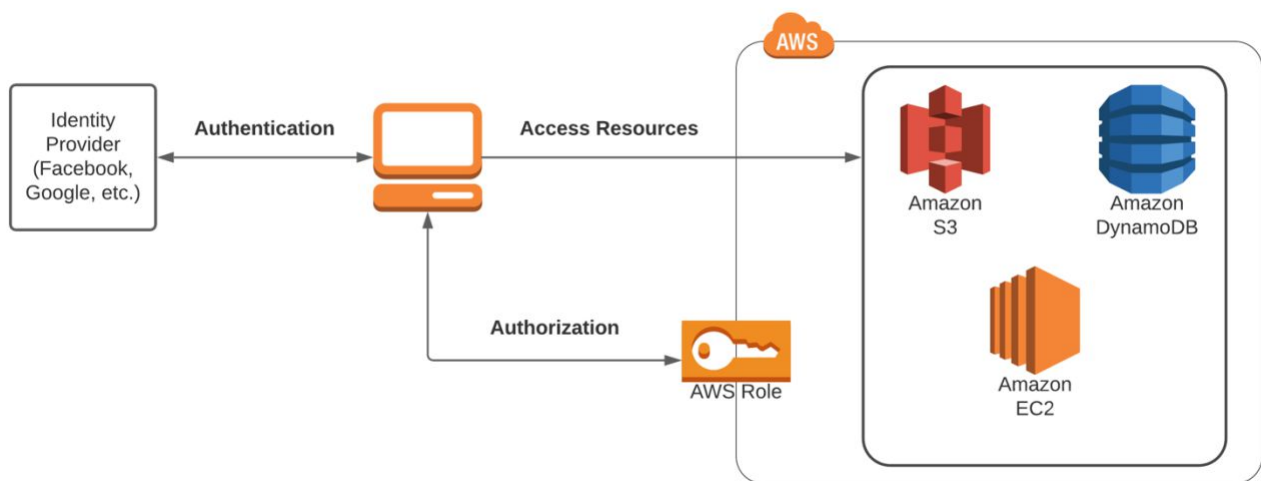


## Identity Federations

IAM is such a critical function for organizational security. As organizations grow in size and complexity, IAM can become challenging to manage. Often the best way to scale IAM systems is to connect (federate) with an identity provider. A VPC can connect to an identity provider and use its IAM database within AWS. Connections with an identity provider are built by building a trust relationship with the identity provider. After the trust relationship is established, a connection is made with OpenID connect (OIDC) or Security Assertion Markup Language 2.0 (SAML).<sup>60</sup>

Identity providers can be an organization's active directory or LDAP systems or external providers such as Google, Amazon, Facebook, Twitter, or LinkedIn. AWS has three choices for authentication with identity providers that are single sign on, Federated IAM, and AWS Cognito.

The diagram below shows how identity federations work with the AWS platform.

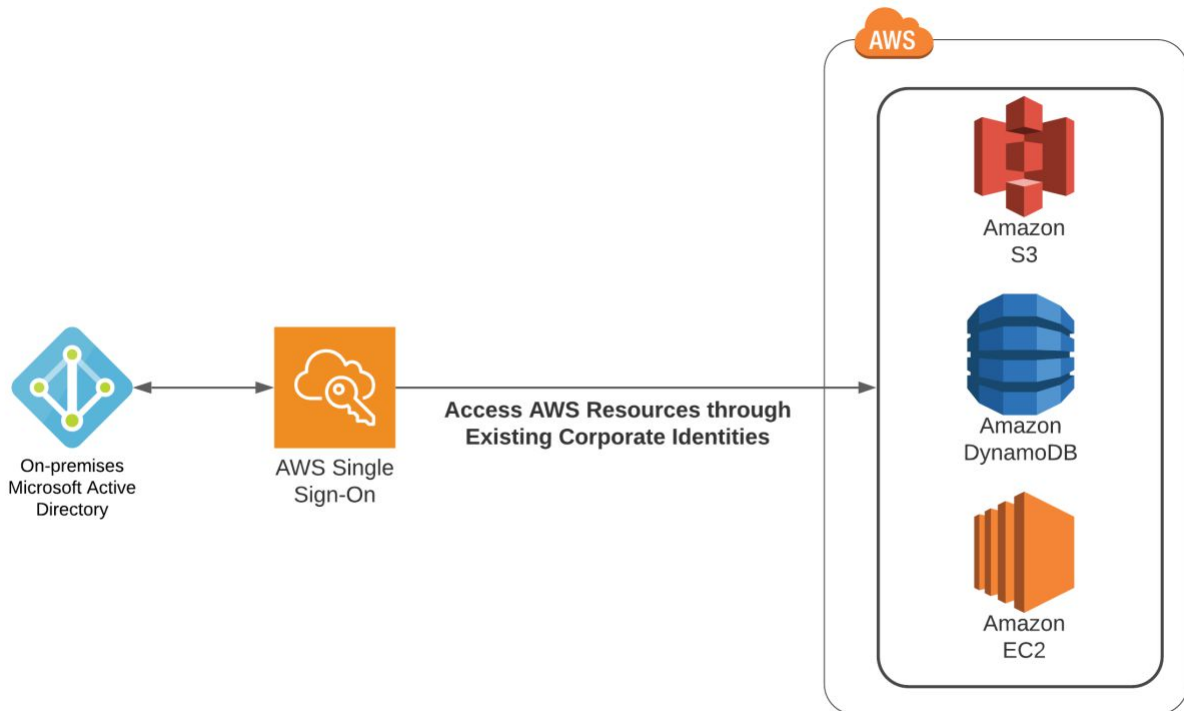


## AWS Single Sign-On

AWS Single Sign-On enables the user to authenticate once to the identity provider, and then they will not need to sign on to access AWS services.<sup>61</sup> IT works in the following manner:

1. The user signs on to the identity provider.
2. The user is authenticated by the identity provider.
3. The identity provider determines what group (permissions) to give the user.
4. The user is given permissions and is authenticated and authorized to use AWS services.

The diagram below shows how Single Sign-On works with the AWS platform.



## Federated IAM

Federated IAM provides a means to authenticate with an external identity provider. Federated IAM enables significant and granular control over user functions. Federated IAM works in the following manner:

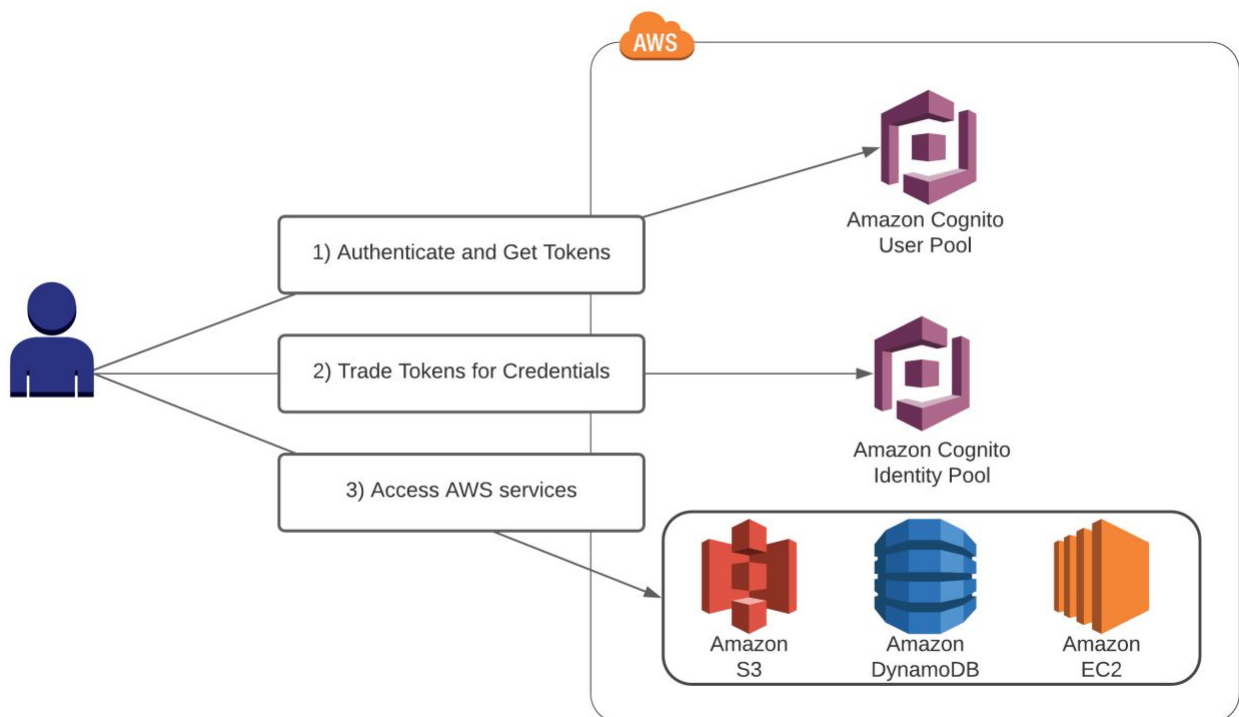
1. A user attempts authentication.
2. The request is forwarded to the identity provider.
3. The identity provider authenticates the users.
4. The identity provider determines the user's privileges.
5. The identity provider grants privileges based upon job role, the organization's cost center, and other factors.

## AWS Cognito

AWS Cognito is an identity and data synchronization service. AWS Cognito enables organizations to synchronize identity management and data across mobile devices. Cognito provides authentication, authorization, and user management for web and mobile apps. AWS Cognito users can sign in directly with a username and password, or with a third-party identity provider such as Facebook or Google.<sup>62</sup> AWS Cognito is simple and efficient. Cognito works in the following manner:

1. The user attempts authenticate against Cognito.
2. Cognito authenticates the user.
3. Cognito provides a token for the user.
4. User device trades token for credentials.
5. The credentials are then used to access AWS services.

The diagram below shows how AWS Cognito is used to authenticate mobile devices to access the AWS platform.



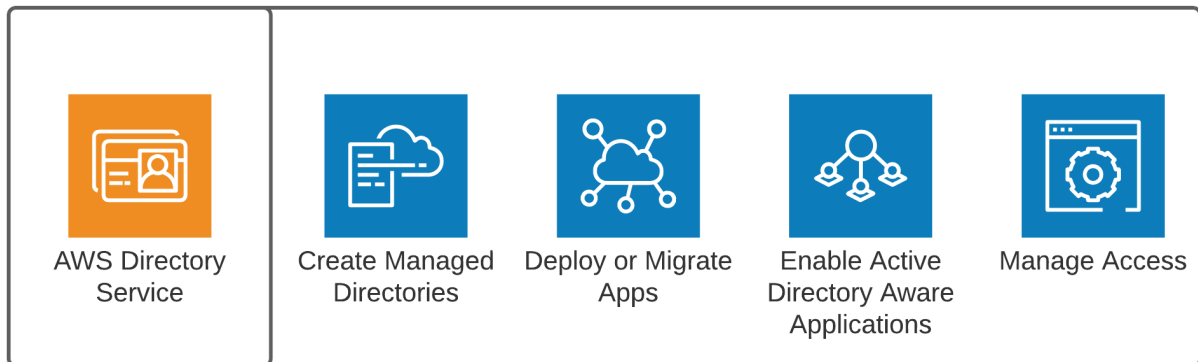
## AWS Directory Service

Another means to create a scalable IAM solution is with the AWS Directory Service. The AWS Directory Service provides hosted, dedicated tenant, Windows Active directory (AD) servers. These are high-availability servers spread across two availability zones with the default configuration. The AD servers are actual Microsoft AD servers hosted by AWS. Being actual Microsoft AD servers, Microsoft dependent workloads can function in the AWS VPC.<sup>63</sup>

AWS Directory Service can also be integrated with customers on premises Microsoft AD domain controllers. AWS Directory Service can also be used by AWS services such as EC2, RDS for SQL server, end user computing, and AWS WorkSpaces for IAM functions. The hosted AD servers

can also be used by EC2, RDS for SQL server, AWS End User Computing, and AWS workspaces for IAM functions.

The diagram below shows how AWS Directory Service is used to facilitate Microsoft applications in the AWS environment.



## Authentication Process

Now that we have discussed the available IAM options, it is necessary to understand how the authentication process works under the different options.

### Username and Password

- 1) User logs in to the console with username and password.
- 2) AWS verifies the user's identity.
- 3) AWS provides an authorization based upon the user's privileges.

### Access Key

An access key is a combination of a twenty-character key ID and forty-character secret. The access key is used for authentication and facilitates connections to AWS via an API. This is generally preformed with the Software Development Kit (SDK).

### Access Key and Security Token

When an IAM authentication needs to occur for an assumed role, a secure token is provided to the requesting application. The secure token, along with the access key, is used for authentication. This provides additional security over other methods.

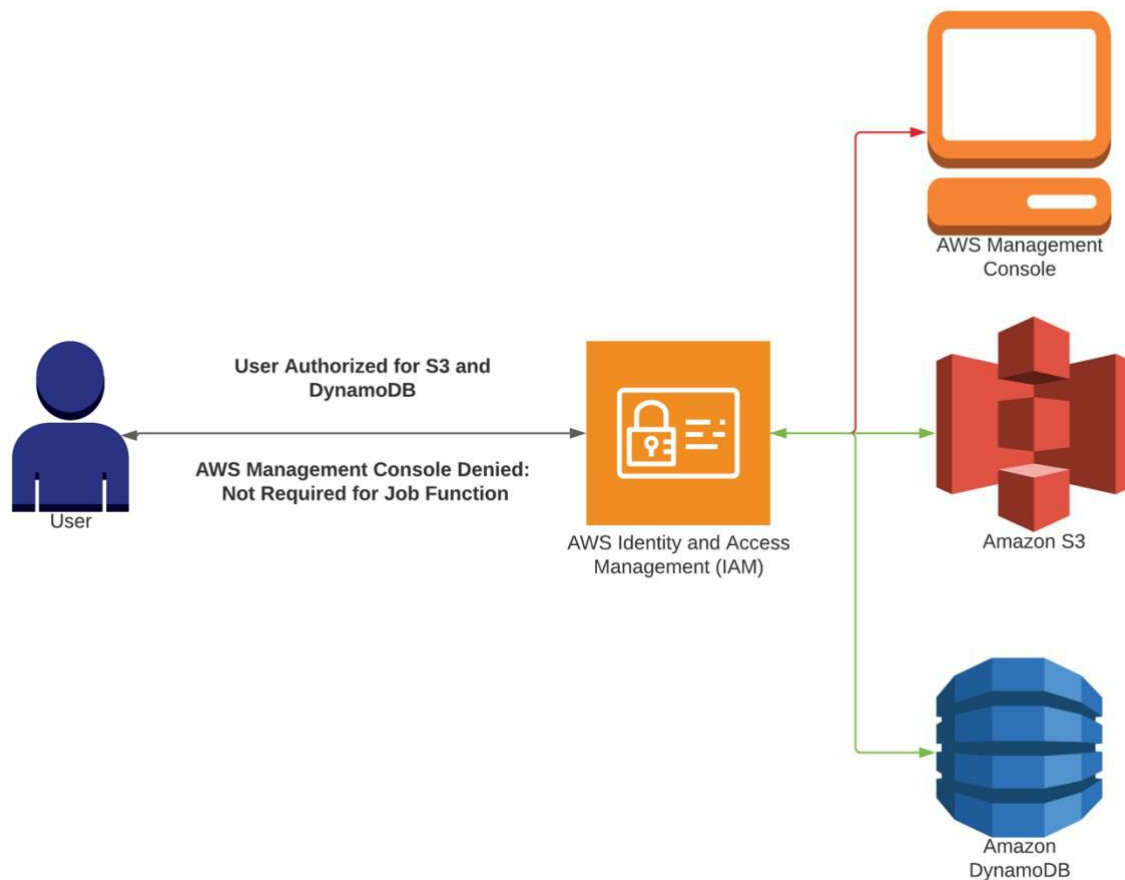
## Authorization

After authentication, it is necessary to authorize the user to perform whatever functions are necessary for their job function. To keep the VPC secure, the default policy is to deny access to all services. Authorization is really about granting permissions to necessary services that have been defined in the IAM policy.

IAM policy documents are written in JavaScript Object Notation (JSON). A policy document defines the following attributes:

- Effect – Allow or deny.
- Service – What service is being requested.
- Resource – What the resource is being made available. This is the full Amazon Resource Name.
- Action – This determines the permissions of the user, i.e., read only, read write.
- Condition – The condition component of IAM is optional. It enables very granular controls, i.e., to allow access from a specific IP subnet, time of day.

The diagram below shows the user authorization on the AWS platform.





## Creating IAM Policies

IAM policies determine who is allowed into the VPC and what actions they can perform. When creating an IAM policy, permissions can be applied to a specific resource or all resources. Providing access to specific resources is based upon the Amazon Resource Name (ARN). Providing access to all resources is accomplished with an asterisks (\*) wildcard. There are two types of policies available in AWS: AWS managed policies and customer-managed policies.<sup>64</sup>

### AWS Managed Policies

AWS managed policies are standalone policies created by AWS. These policies have several key attributes:

- Provide permission for services and functions within AWS.
- Are optimized for common use cases.
- Can be attached and moved to different entities and accounts in AWS.
- Can be based on job role to provide different levels of access.
- Have two major predefined roles: administrator access and power user.
- Administrator access provides full access to every service.
- Power users essentially have full access with the exception of IAM and organization management.

### Customer-Managed Policies

Customer-managed policies are managed by the customer for their account. These policies have several key attributes. They:

- Are not visible outside the customer's organization.
- Are custom made for the organization's specific needs.
- Can be attached to entities within the AWS account.

### How to Create an IAM Policy

There are several ways to create an IAM policy. To make an IAM policy from the console, perform the following steps:

1. Sign in to the IAM console from a user account with administrator privileges – <https://console.aws.amazon.com/iam/>
2. In the navigation pane, choose policies.
3. You will see a list of AWS managed policies. These are simple to use, updated by AWS as needed, and can help avoid configuration errors.

- Alternatively, you choose to create a customer-managed policy starting with an AWS policy and then customize and use the policy generator. Or you can create one from scratch.

## Copying an AWS Managed Policy

If an organization elects to create its own policy, the easiest method is to copy an AWS managed policy and customize. This is the simplest method, and it helps to avoid configuration errors by starting with a known good configuration.

## AWS Policy Generator

The AWS Policy Generator is an easy-to-use questionnaire that will generate an IAM policy.<sup>65</sup> The policy generator works as follows:

- Go the policy generator page.
- Answer the questions.
- Assign permissions to specific resources. Multiple permissions can be created as statements.
- A policy document is then created that can be edited.

The diagram below shows how the AWS Policy Generator can be used to create custom IAM policies.

The screenshot shows the AWS Policy Generator web interface in a browser window. The URL is `awsolicygen.s3.amazonaws.com`. The page has the Amazon Web Services logo at the top. Below the logo, it says "AWS Policy Generator" and provides a brief description: "The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#)." The interface is divided into three main steps:   
**Step 1: Select Policy Type** - A section titled "Select Type of Policy" with a dropdown menu set to "IAM Policy".   
**Step 2: Add Statement(s)** - A section titled "Add Statement(s)" with a sub-header "A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements." It includes radio buttons for "Effect" (selected: "Allow", unselected: "Deny"). Below this, there are two options for "AWS Service": "Amazon SQS" (selected) and "All Services (\*\*)" (unselected). A note says "Use multiple statements to add permissions for more than one service." Below that, there are two options for "Actions": "Select Actions" (selected) and "All Actions (\*\*)" (unselected). A note says "Use a comma to separate multiple values." Below that, there is a text input field for "Amazon Resource Name (ARN)" with a placeholder "ARN should follow the following format: `arn:aws:sqs:<region>:<account_id>:<queue_name>`". Below the input field are two buttons: "Add Conditions (Optional)" and "Add Statement".   
**Step 3: Generate Policy** - A section titled "Generate Policy" with a sub-header "A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements." Below this, it says "Add one or more statements above to generate a policy." At the bottom of the page, there is a disclaimer: "This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies." and a copyright notice: "©2010, Amazon Web Services LLC or its affiliates. All rights reserved." and "An amazon.com company".

## Create an IAM Policy from Scratch

IAM policies can be generated from scratch in JSON format. When creating an IAM policy, it is essential to make sure to use the proper grammar and syntax. Creating an IAM policy from scratch is ideal for organizations that have individuals with JSON programming expertise and require significant policy customization.

### Sample IAM Policy

Now that we have discussed the methods to create an IAM policy, let's evaluate a sample policy. A sample policy can be seen below:

```
{
  "Version": "2020-09-01",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "ArnEquals": {"ec2:SourceInstanceARN": "arn:aws:ec2:*:*:instance/instance-id"}
      }
    }
  ]
}
```

Now let's evaluate this IAM policy.

1. The "version" and "date" explain when the policy was created.
2. The first string allows mounting and unmounting of a volume on EC2.
3. The next statement provides the locations to the volumes.
4. The third line specifies conditional elements that allow the policy to be in effect and is optional.

## **Applying the IAM Policies**

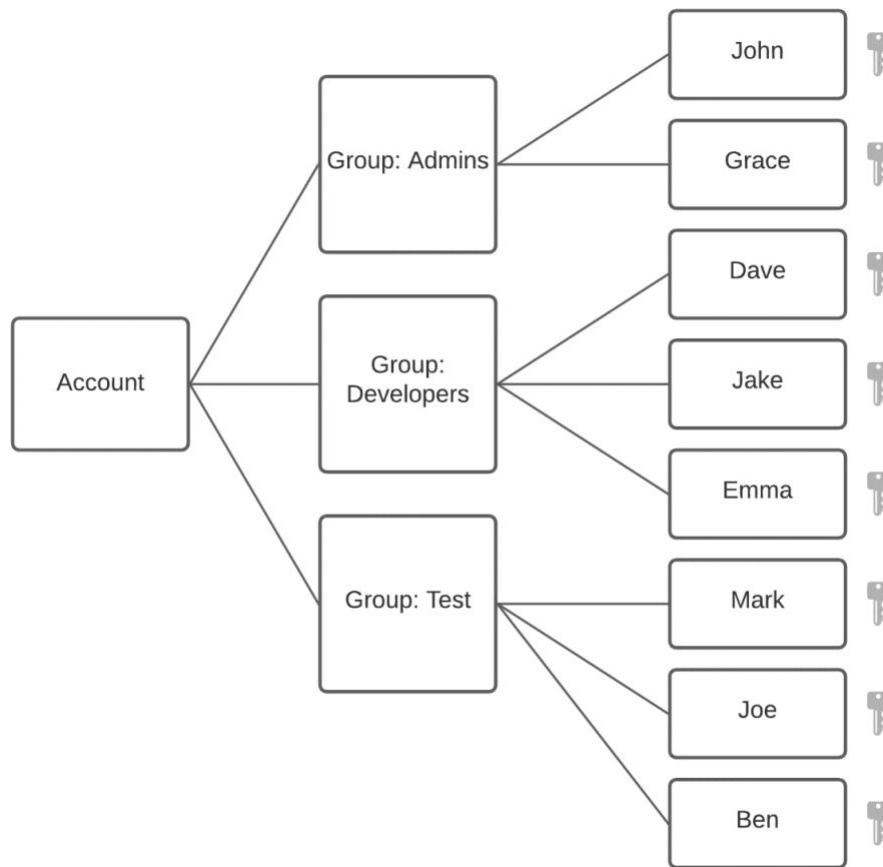
After the IAM policies are created, they need to be applied in order to function. IAM policies can be applied in several manners. Policies can be applied as a user policy, managed policy, or a group policy.

User policy – A user policy is applied to individual users. This works well but quickly becomes unscalable when many users exist in an organization. Imagine configuring individual policies for 200,000 employees.

Managed policy – With managed policies, a policy is created and exists independently of the user. The policy can then be attached to users or groups (collections of users). This method is effective and scalable.

Group policy – In this method, an organization creates a group and attaches a policy to the group. Users are then added to groups and inherit the policy from the group. For example, a group is created for developers. The group is given the permissions necessary to perform their roles. The developers are then added to the group. Another group could be created for finance that would allow for obtaining cost metrics. Individuals from the finance department would be added to the finance group. This is a highly effective and scalable method for IAM policies.

The diagram below shows how group policies are used for IAM with the AWS platform.



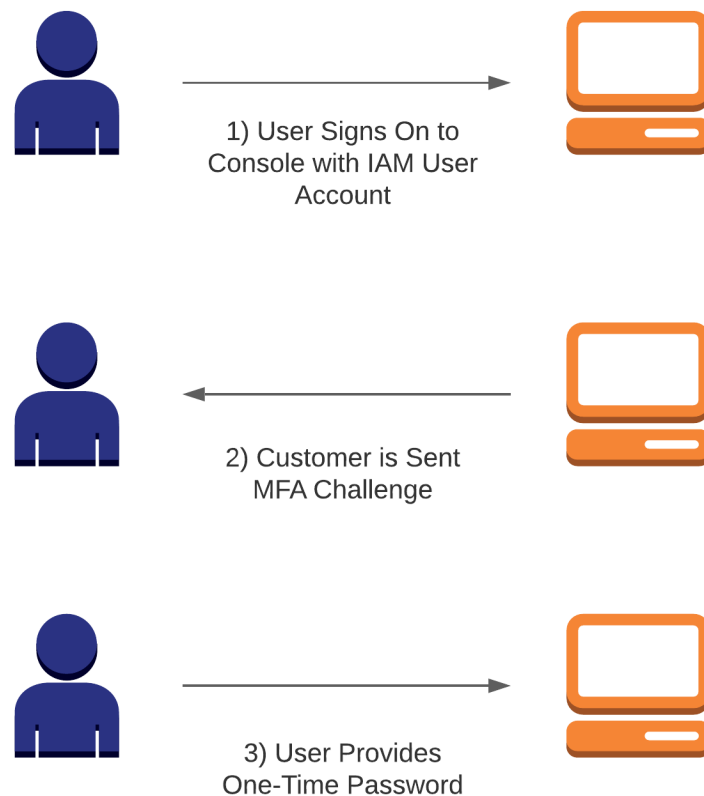
### Further Securing IAM with Multifactor Authentication

Organizations looking for enhanced security around IAM can use multifactor authentication (MFA). MFA applies to the security concept of something you have and something you know. A perfect example is a debit card. To access money in your account, you need the card (something you have) and a pin number (something you know). This combination greatly enhances the security posture. MFA works in the following manner:

1. The organization sets up an authenticator app or device with a key.
2. The authenticator device creates a one-time password that changes every few seconds.
3. When the user logs in with their username and password, AWS will provide a challenge asking for the one-time password.
4. If the user provides the correct one-time password, they will be authenticated into the system.

This setup provides substantial security. Even if a hacker were to obtain the username and password, they would not be allowed into the system without knowing the constantly changing one-time passwords.

The diagram below shows how multifactor authentication is used in the AWS platform.

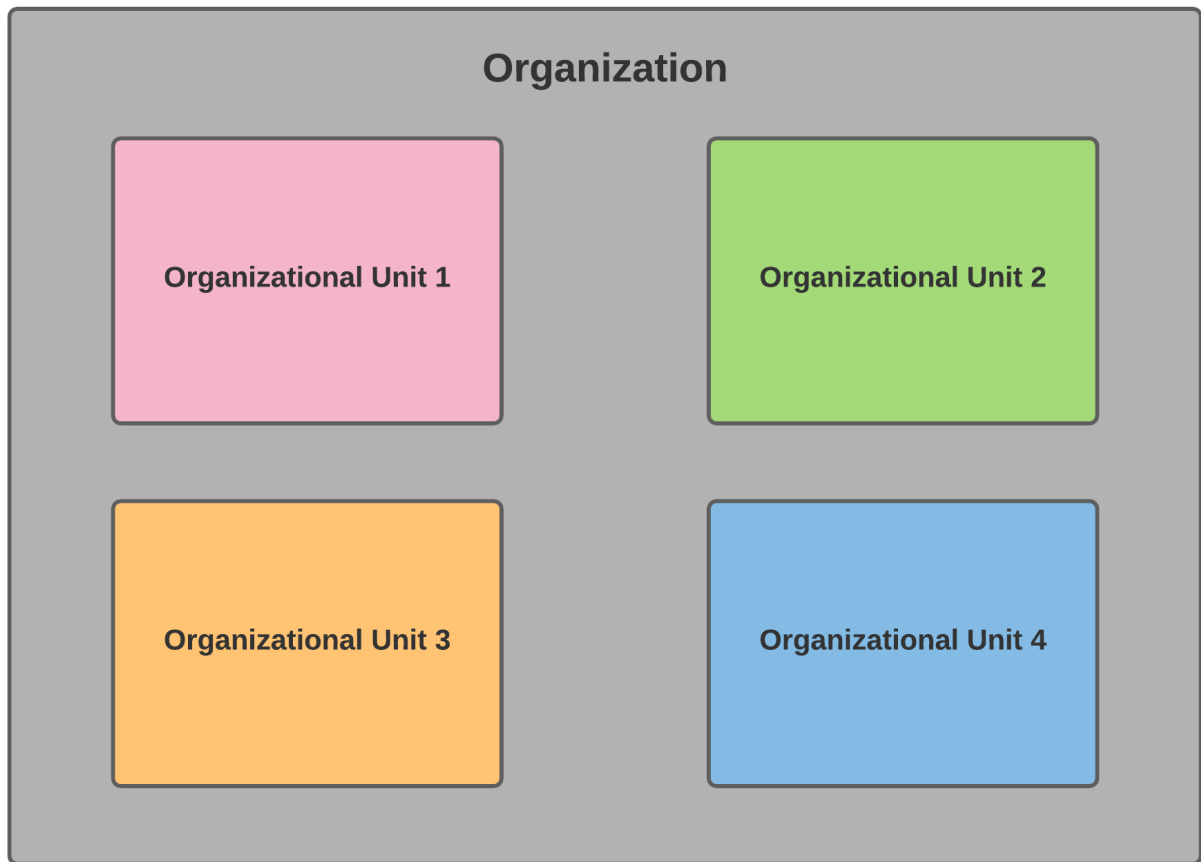


### Multi-Account Strategies

Another method to increase the security of the VPC is to partition the organization into multiple small accounts and share information between accounts. Each small account will be placed into a single billing unit called an organization unit (OU). Since all organization units are placed into a single billing organization, the organization can still benefit from volume discounts on the total services they consume.<sup>66</sup> Multi-account strategies are especially beneficial from a security perspective because of the following security enhancements:

- Isolation between organizational units.
- The ability to share only the necessary information between units.
- The ability to reduce the visibility of workloads between organizations.
- The ability to reduce the blast radius (meaning that if a problem happens in a single OU it won't affect another OUs).
- The ability to truly compartmentalize data.

The diagram below shows an example of a multi-account strategy on the AWS platform.



## Restricting Network Access

A strong security posture involves allowing traffic needed for business operations while keeping all unwanted traffic out of the network and services. Traffic filtering is accomplished with network ACLs, security groups, and firewalls. Network ACLs and security groups are covered in depth in the networking section of this book.

Firewalls are widely used in enterprise networking. Traditional firewalls keep unwanted traffic out of the network like a network ACL does, but firewalls are stateful. Since firewalls are stateful, they know the state of every connection that has passed through the firewall. Being stateful, firewalls default policies block all incoming traffic but allow return traffic initiated by internal users that pass through the firewall. Modern firewalls can recognize common attack patterns and stop them by dynamically applying new firewall rules. AWS has a modern firewall solution that can be used with CloudFront, application load balancers, and API gateways.

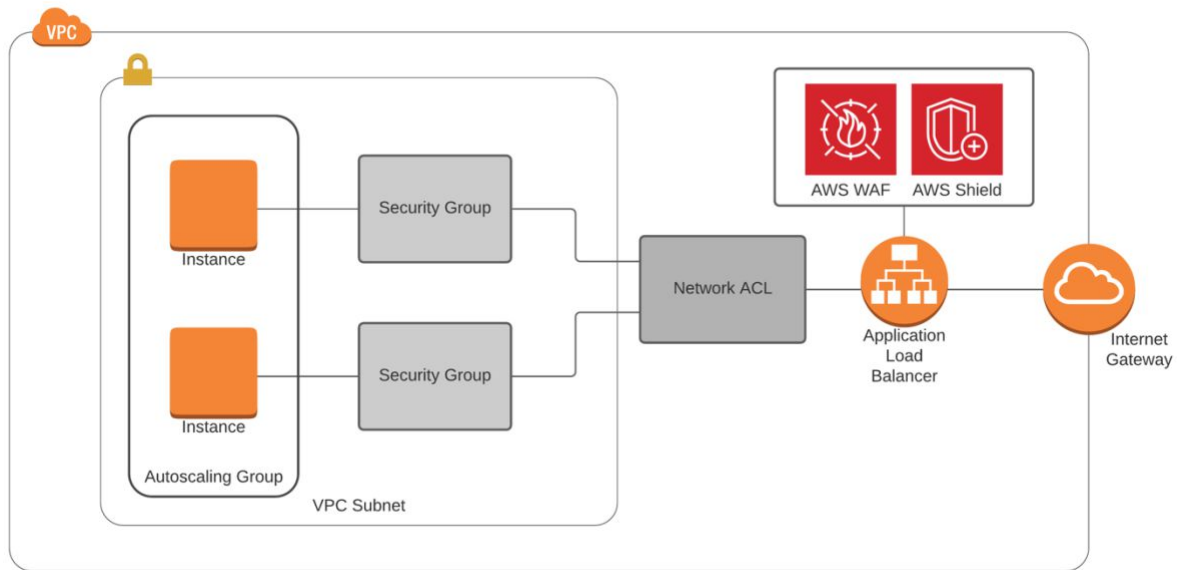
## Preventing Distributed Denial of Service Attacks

Distributed denial of service (DDoS) attacks are a common assault against an organization's systems. A DDoS attack is designed to interrupt the normal function of server, application, or network by overwhelming the service or its surrounding infrastructure. A DDoS attack is implemented by flooding traffic or server requests from multiple computers on the internet. Preventing a DDoS attack takes a full security posture. The key elements of DDoS prevention within AWS are:

- Block unwanted traffic with network ACLs, which reduces the options the attacker can use to attack the network or server.
- Keep unwanted traffic out of servers and AWS services with security groups.
- Use a firewall. Adding AWS Web Application Firewall (WAF) can recognize common attacks and can dynamically apply policies to mitigate these attacks.
- Leverage AWS Shield, which provides enhanced DDoS protection. There are two versions of AWS Shield—Standard and Advanced. AWS Shield Standard is provided at no additional cost for organizations using AWS WAF. AWS Shield Advanced is an enhanced option at an additional cost that provides protection to EC2, ELB, CloudFront distributions, Route 53, and AWS Global Accelerator.
- Leverage autoscaling. Since the goal of a DDoS attack is to overwhelm the network or computing platform, autoscaling can help mitigate against DDOS attacks. During a DDoS attack, autoscaling can help increase compute capacity to offset the loss of computing capacity from the attacker.



The diagram below shows how multiple network security measures are combined to thwart a distributed denial of service attack.



### Amazon Web Application Firewall (WAF)

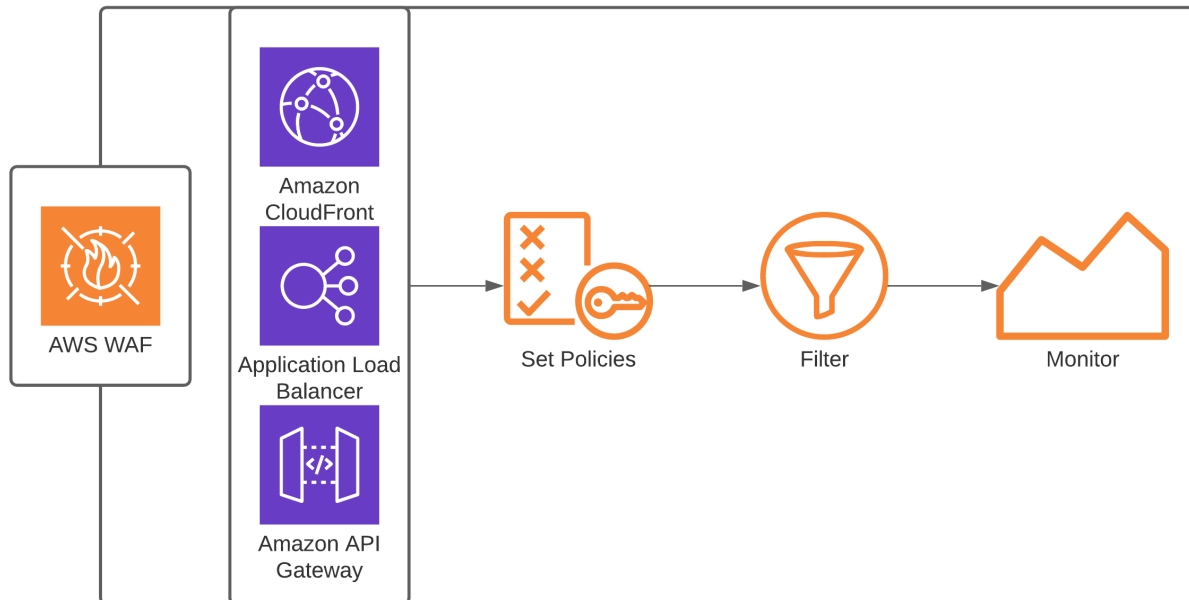
AWS WAF is a web application firewall that protects against attacks. WAF monitors HTTP(S) requests looking for exploits and can mitigate an attack if it occurs. WAF controls access to content based upon the firewall policy. WAF can assist with protection of many AWS services including CloudFront distributions, Amazon API Gateway, or application load balancers.

AWS WAF provides granular control to protect an organization's resources. It provides a means to control access through web ACLs, rules, or rules groups. Web access control lists (web ACLs) are lists that either allow or deny traffic depending upon the configured policy. Rules are statements that allow or deny traffic based on the criteria in the policy. Rule groups are groups of individual rules that can be reused in other places.<sup>67</sup>

Setting up and using WAF is performed in the following manner:

1. Enable WAF on the application or device.
2. Create a policy that filters access to the application.
3. WAF analyzes the traffic depending on the policies created.
4. WAF will permit or deny the traffic depending on the traffic's adherence to the WAF policy.
5. If an attack occurs, new rules can be created to mitigate the attack.
6. WAF integrates with CloudWatch to provide increased visibility into network traffic and potential or actual attacks.

The diagram below shows how the AWS Web Application Firewall functions in the AWS environment.



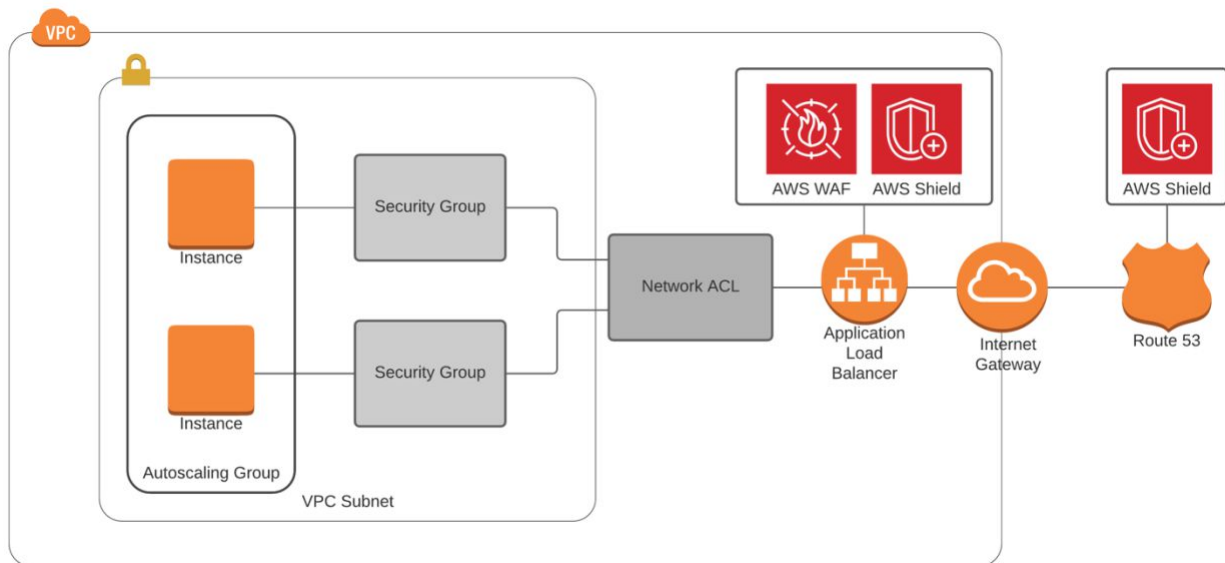
## AWS Shield

AWS Shield is an AWS service to protect against DDoS attacks. AWS Shield is available in two versions: AWS Shield Standard and AWS Shield Advanced.<sup>68</sup>

### AWS Shield Standard

AWS Shield Standard is a free DDoS protection service for AWS customers using WAF. AWS shield standard protects against the most common attacks. According to AWS, Shield Standard blocks against 96 percent of the most common attacks including SYN/ACK floods, reflection attacks, and HTTP slow reads. AWS Shield Standard works based upon the logic contained in its policy.

The diagram below shows how AWS Shield is used to defend against distributed denial of service attacks.



## AWS Shield Advanced

AWS Shield Advanced is a paid DDoS protection service for AWS customers. It has many advantages and additional features above AWS Shield Standard. AWS Shield Advanced has a rich feature set and functionality, including:

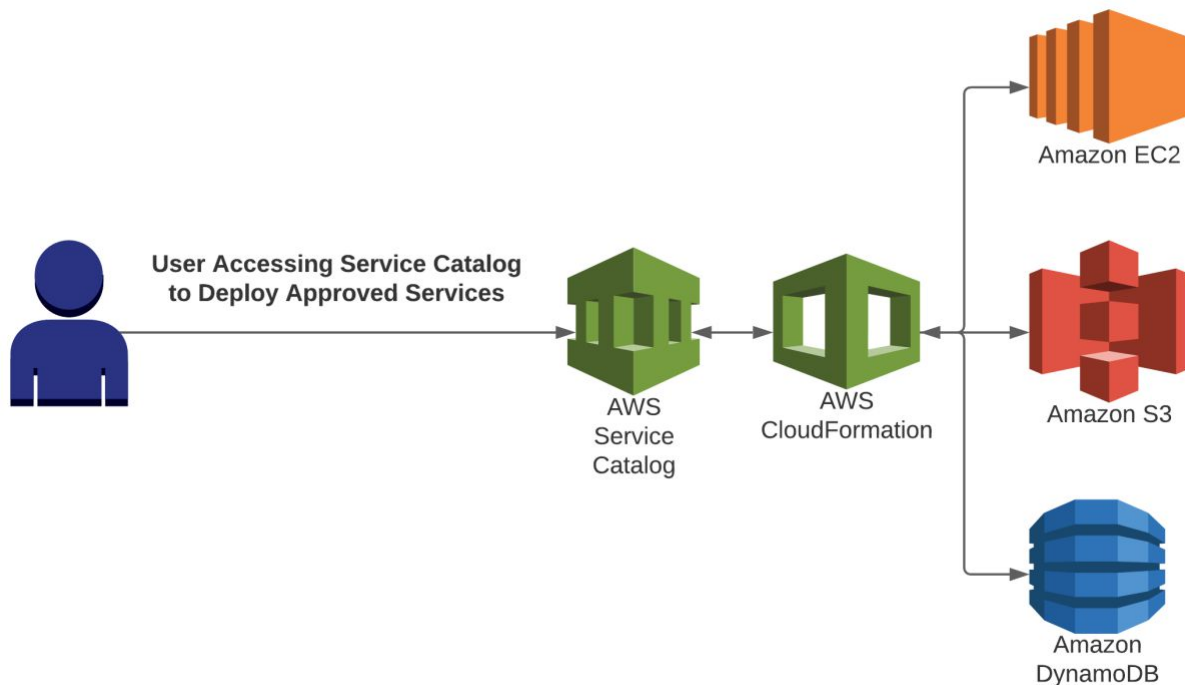
- Additional protection for volumetric attacks by adding intelligent attack detection and mitigation tools.
- Dynamic solution that can look at traffic patterns to determine if an attack is occurring.
- Ability to detect an attack and can automatically deploy ACLs to mitigate the attack.
- Visibility and notification for layer 3/4/7 attacks.
- 24/7 access to a DDoS response team, assuming the customer is a member of the business or enterprise support.
- Protection for ELBs, EC2 instances, CloudFront distributions, Route 53, and AWS Global Accelerator.

## AWS Service Catalog

As previously discussed, a full security posture includes physical security, access lists, security groups, firewalls, DDoS protection, IDS/IPS, and controlling and optimizing what is placed on the network. While there are many ways to control what is placed on the network, AWS makes it easier with the use of the AWS Service Catalog. A service catalog is a means to create a list of approved services. The service catalog is defined by the customer, so they can allow what services they desire on the organization's network. The service catalog can include specific

AMIs, servers, software, databases, and multitier application architectures. Therefore, the service catalog can help ensure compliance with corporate security standards by limiting what system admins can place on the network. For example, the service catalog can be configured to allow only security hardened AMIs on the network (fully patched, disabling unnecessary services, etc.). AWS Service Catalog simplifies deployments, as administrators can allow only approved and compliant services.<sup>69</sup>

The diagram below shows how the service catalog is used to select and deploy approved services.

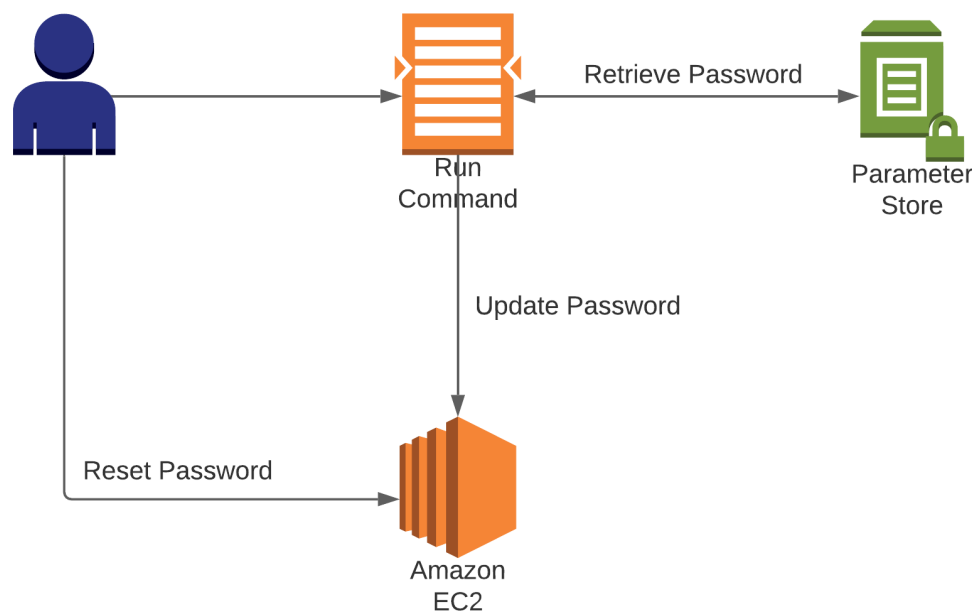


### **AWS Systems Manager Parameter Store**

Another component of an overall security posture is secure management of passwords, database strings, and other critical components. Proper management of passwords is essential for a secure environment. If passwords are compromised, the organization would be open to attacks by hackers. Stolen passwords can lead to serious data loss and exploitation by hackers. AWS provides a solution for securely storing passwords, which is the AWS Systems Manager Parameter Store, which increases the security posture by separating an organization's code from their secret information.<sup>70,71</sup>

The AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management, and secrets management. It scans your managed instances and will report any policy violations if are detected. The AWS Systems Manager Parameter Store is a scalable, hosted serverless environment optimized for storing passwords, database strings, license codes, and API keys. For extremely sensitive information, it's advisable to encrypt data in the store. It provides a means to encrypt sensitive data and provides an excellent means to track password use, as well as audit who has been accessing the system.

The diagram below shows how the Systems Manager Parameter Store can be used for secure password storage and maintenance.



## Labs

1) Set up Multifactor Authentication (MFA) on the root account. Link on how to set up (MFA) below.

[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_credentials\\_mfa\\_enable\\_virtual.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_virtual.html)

2) Create a new IAM user. Link on how to create a new user below. Give the user full permissions. Link on how to create IAM user and assign permissions below.

[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_users\\_create.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html)

3) Create an IAM group. Create a new IAM user. Add the new IAM user to the group. Assign any permissions to the group you desire. Link on how to create IAM group is below.

[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_groups\\_create.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups_create.html)

## Chapter 8

### AWS Applications and Services

AWS has several key services that can be used to enhance VPC based applications and services. This chapter will cover these key AWS services:

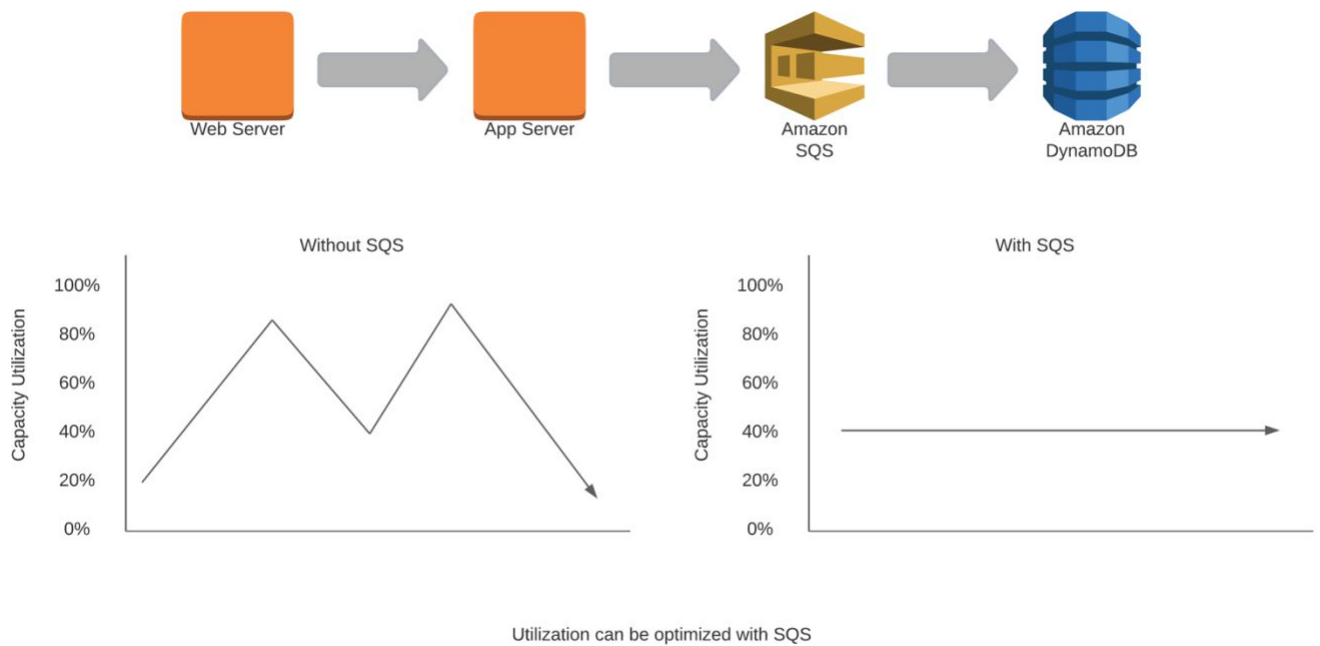
- Simple Queueing Service (SQS)
- Simple Notification Service (SNS)
- Simple Workflow Service (SWF)
- Kinesis
- Elastic Compute Service (ECS)
- Elastic Kubernetes Service (EKS)
- Elastic Beanstalk
- CloudWatch
- Config
- CloudTrail
- CloudFront
- Lambda
- Lambda@edge
- CloudFormation
- AWS Certificate Manager (ACM)

#### AWS Simple Queueing Service (SQS)

To design a highly available and scalable application, it is sometime necessary to decouple the components of an application's architecture. By decoupling the application's architecture, it is possible to minimize system bottlenecks and optimize performance of the entire system. A great option for decoupling application architectures is the Amazon Simple Queueing Service (SQS).<sup>72</sup>

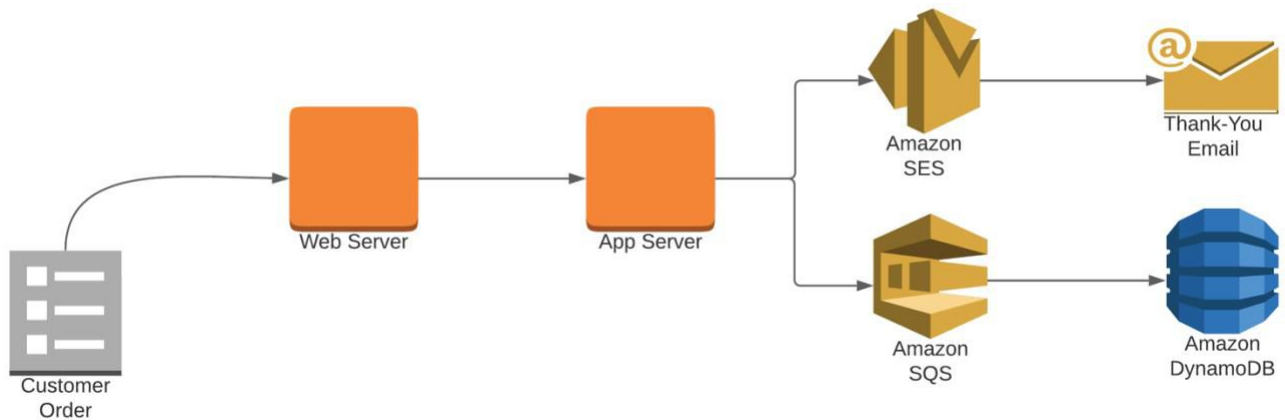
SQS is a message queuing service that provides temporary message storage prior to the message being transmitted to its ultimate destination. SQS enhances application availability by providing a way to retain messages when a part of the application's architecture is busy or unavailable. Since SQS can hold messages on the way to the destination, the organization can optimally size their architecture. If a traffic spike occurs, SQS will store and deliver the message when it's ready.

The diagram below shows how SQS can optimize performance and help promote scalability.



SQS is used to decouple application architecture components, which promotes scalability. SQS can further increase scalability and elasticity of services when SQS is used to autoscale instances based on the message's depth of the SQS queue. SQS can take the place of messaging middleware in multitiered applications. SQS is highly available, and multiple copies of every message are stored redundantly. SQS has integration with KMS to allow for end-to-end encryption. SQS is highly tunable transient storage. The SQS message queue retention period can be adjusted for up to fourteen days, with the default storage time being four days.

The diagram below shows how SQS queues are used to decouple application architecture components.



SQS can be configured for standard queues (the default), first in, first out (FIFO) queues and dead-letter queues. The key attributes of these queue types are:

### Standard Queues

- Super fast and support a nearly unlimited number of requests per second.
- Offers the fastest throughput and message delivery.
- Every message is delivered at least once.
- Best effort delivery, and messages may be delivered out of order.

### First In, First Out Queues

- High throughput but much slower than standard queues.
- First in, first out delivery guarantees the messages will be processed once and in the order they are received.
- Since messages are sent in the order they are received, it is possible the FIFO queue will increase latency because all new messages will be waiting for the previous message to be processed.

### Dead-Letter Queues

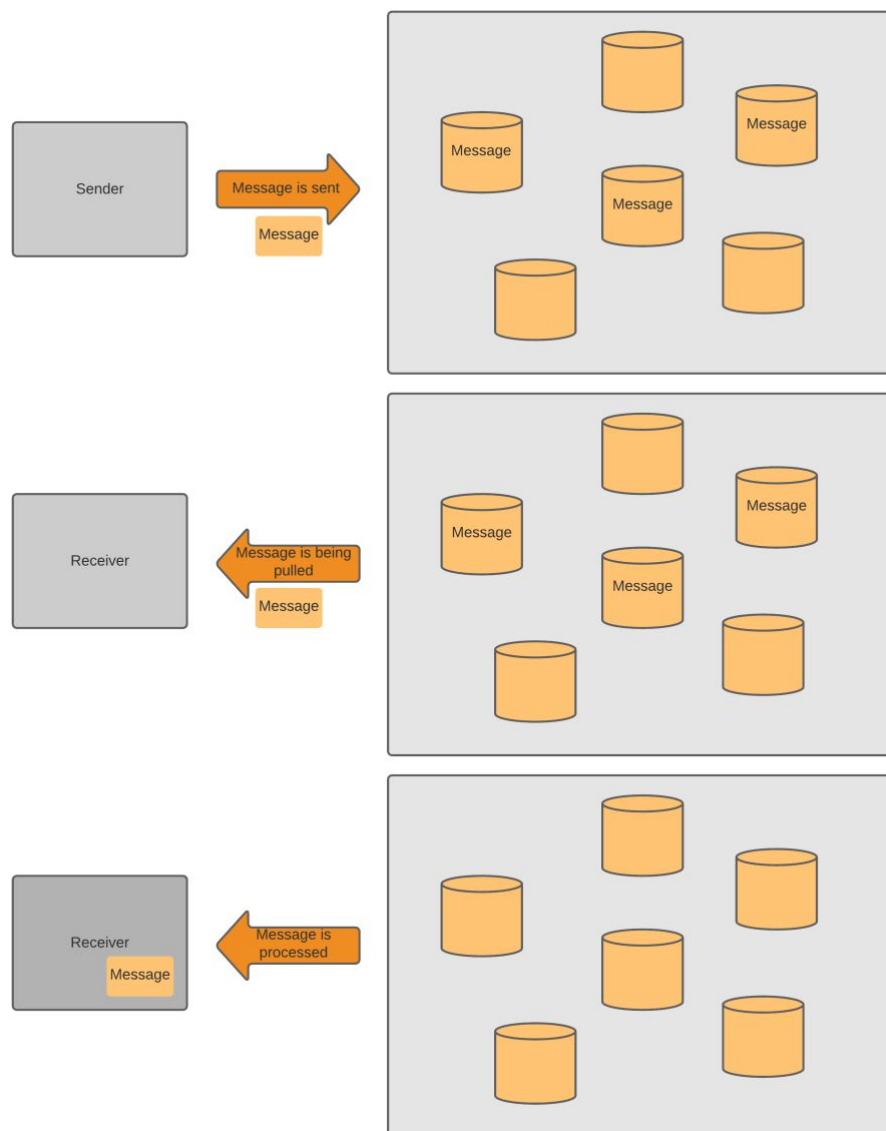
- Dead-letter queues (DLQ) can be set up to retain undeliverable messages if an error occurs in the system.



## How SQS Works

1. Messages are sent from the computing platform to the queue as a step to their ultimate destination.
2. After the message is inside the queue, it can be scheduled for delivery based upon the capacity of the ultimate destination for the message.
3. If the ultimate destination is busy, the message can stay in the queue until it is processed or times out. The message can stay in the queue for up to fourteen days based upon the SQS configuration.
4. The message is pulled from the queue to be processed.
5. After the message is completely processed, the message is deleted from the queue.

The diagram below shows how SQS is used to reliably deliver messages to their intimate destination.



## When to Use SQS

SQS can be extremely beneficial in designing a highly-available and scalable architecture. Some key situations where SQS can make a notable difference:

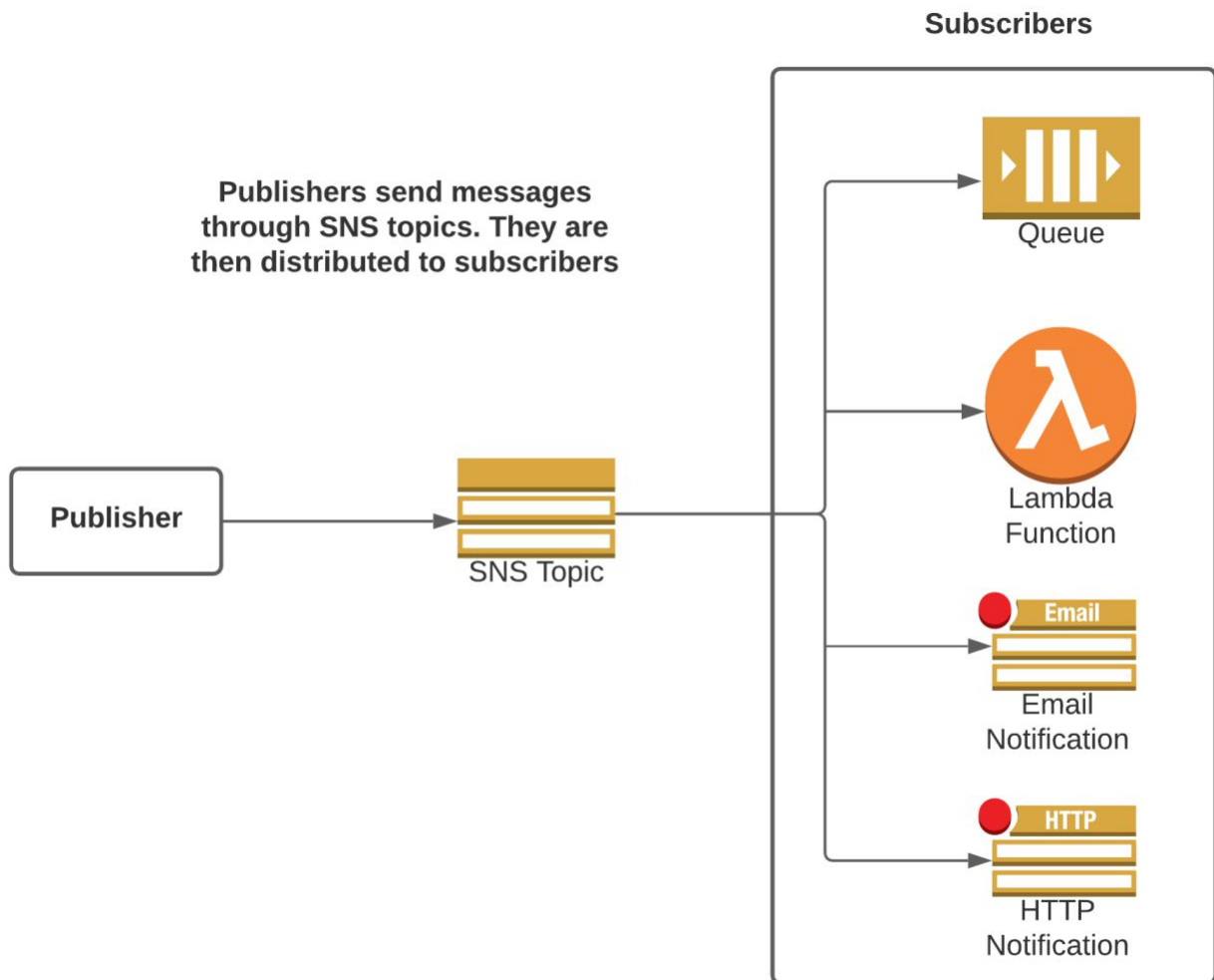
- With capacity planning and application scalability.
- To make sure messages (orders) are not lost if part of the system is overloaded (i.e., database on multitiered application).
- For cost optimization, as it offers the ability to right-size the instances supporting an application.
- For autoscaling, with its ability to trigger autoscaling based upon queue depth, as opposed to a less direct metrics, i.e., CPU utilization.
- To support an application's ability to handle large spikes in traffic, without having to scale or make changes to the platform
- To handle increased traffic destined for databases, often without the need to increase write capacity on the database.

## AWS Simple Notification Service (SNS)

Amazon Simple Notification Service is a managed messaging service to deliver messages between systems, or between systems and individuals. SNS is used to decouple messages between microservice applications. SNS is also used to send SMS and email, and push messages to devices.<sup>73</sup>

SNS facilitates communication between senders and recipients via a publish-subscribe model. The publish-subscribe (pub-sub) messaging model enables notifications to be delivered to clients using a push mechanism. Push notifications notify clients of message updates. SNS consists of publishers and subscribers. Publishers communicate by sending a message to a topic. A subscriber subscribes to a topic and receives messages that have been published to the topic. It functions like an email list—you subscribe to the list and receive messages the sender (publisher) sends to the list.

The diagram below shows how the SNS publisher/subscriber model works for message delivery.



SNS is a highly-available platform that by default runs across multiple availability zones. SNS can be used to fan out messages to a large number of subscriber systems or customer endpoints. Endpoints can be many things; some examples are SQS queues and Lambda functions. SNS allows message filtering through policies so that only desired notifications are received. SNS encrypts messages immediately to protect from unauthorized access.

SNS can be used in a variety of situations. Some common SNS use cases can be seen below.

### Application and System Alerts

- SNS can send a notification when a predefined event occurs (i.e., a limit is passed).

- For example: When a CPU's utilization goes over 80 percent, notify system administrators.

### **Email and Text Messages**

- SNS can push notifications to people email and/or send them text messages.
- For example: a company's CEO is going to be on TV and a broadcast link is sent to employees' email and phone.

### **Mobile Notifications**

- SNS can send push notifications directly to mobile applications.
- For example: notify a customer of a flash sale on your app.

### **AWS Simple Workflow Service (SWF)**

SWF is a workflow management solution. SWF enables the coordination of tasks across distributed application components. SWF enables you to create a workflow of tasks that take multiple steps for completion. SWF then coordinates the execution of tasks across the platform. Normally it would take substantial application development to coordinate tasks across multiple systems. As SWF is a prebuilt workflow management solution, all that's necessary is to tell SWF the necessary workflow steps, and SWF manages all coordination of steps until completion.<sup>74</sup>

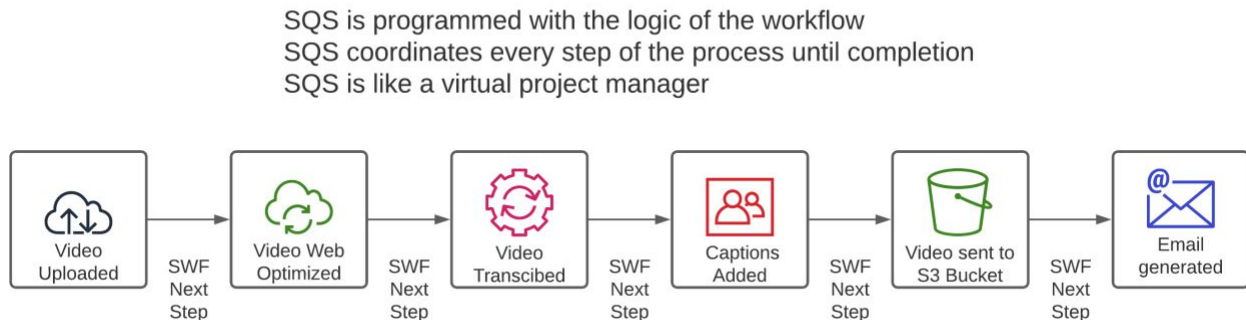
SWF manages workflows using workers that carry out the steps. These workers are programmed to perform, process, and confirm the completion of each step. Workers can be deployed using EC2, Lambda, or on a local system.

SWF controls the flow of tasks using deciders to keep track of the workflow. The decider receives decision tasks from SWF and then determines and schedules the next step to complete the task.

Let's explore a common multistep workflow that would benefit from SWF.

1. Video is uploaded.
2. Video is processed and converted to an optimized format.
3. After formatting, the video is transcribed.
4. After transcription, the video's transcriptions are added as subtitles.
5. After final processing, the video is stored on a server.
6. After the video is stored on the server, the user gets a notification that their video is ready for download.

The diagram below shows how SWF can be used in the above multistep workflow.



Without SWF, it would be necessary to develop software to manage coordination of all steps. With SQF, all that is necessary is to configure the workflow in SWF. SWF will manage the process from the time the video is uploaded until the time the user gets notified that the video is ready for download.

## AWS Kinesis

Amazon Kinesis is an AWS service for collecting, processing, and analyzing streaming data. Amazon Kinesis can collect and analyze streaming data in real time from sources, including video, audio, logs, website clickstreams, and internet of things (IoT) devices. Unlike traditional environments where you collect, store, and then analyze the data, Kinesis can do this in real time. By analyzing data in real time, the organization can receive a competitive advantage by not having to wait for data storage and processing to obtain actionable insights.<sup>76</sup>

### Why Use Kinesis

Kinesis is ideal for situations when large amounts of streaming data needs to be rapidly moved and processed. Some application examples:

- Weather sensors located across the globe that report current conditions every five minutes.
- A fleet of airplanes sending information about their status every few minutes.

These applications generate a large volume of streaming information, and Kinesis makes managing these types of data streams in real time feasible.

## Kinesis Platforms

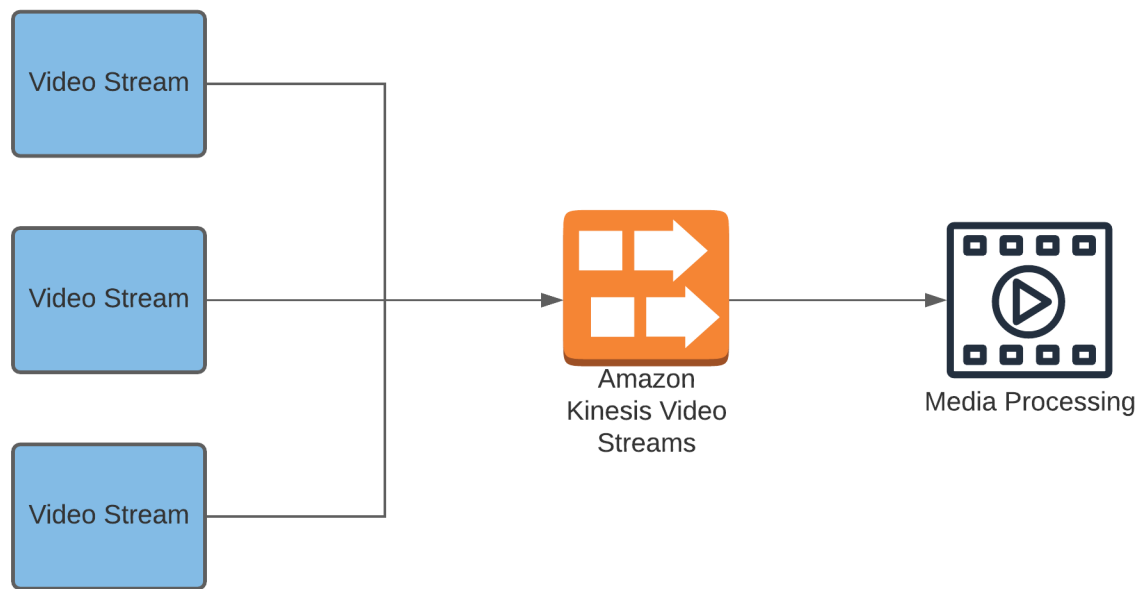
There are four Kinesis platforms:

- Kinesis Video Streams
- Kinesis Data Streams
- Kinesis Data Firehose
- Kinesis Data Analytics

### **AWS Kinesis Video Streams**

Kinesis Video Streams is a Kinesis application specifically for video data. Kinesis Video Streams enables collection of videos from multiple sources, as well as providing ingestion, storage, and indexing of multiple streams. The videos obtained by Kinesis streams can be sent for media processing or be used by machine learning applications.

The diagram below shows how AWS Kinesis Video Streams is used to capture large amounts of real time streaming data.



## AWS Kinesis Data Streams

Kinesis Data Streams is highly scalable platform for real-time data. Kinesis Data Streams can capture gigabytes per second of data from hundreds or thousands of sources. This includes financial transactions, location-tracking, database event streams, and other streaming data. Kinesis Data Streams can ingest an organization's data and provide meaningful insights using business intelligence tools.

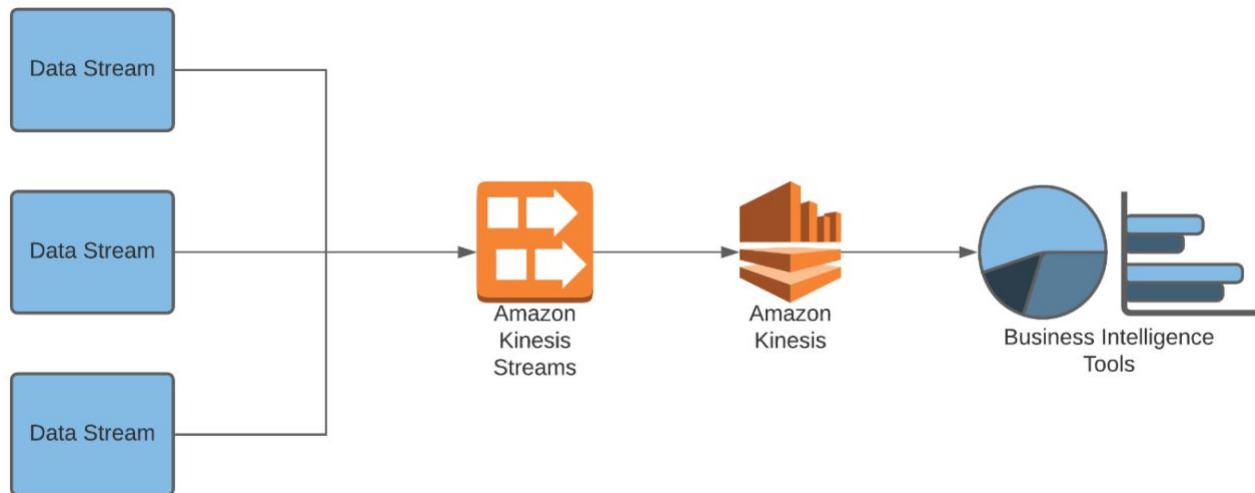
Kinesis Data Streams can be used in a variety of situations. Some common use cases are:

- Large event data collection.
- Real-time data analytics.
- Capturing gaming data.
- Capturing mobile data.

## How It Works

1. Streaming data is captured by AWS Kinesis Streams.
2. Streaming data is then sent for processing via EC2 and/or Kinesis Data Analytics.
3. After data is processed, the data can be sent to business intelligence tools.

The diagram below shows how AWS Kinesis Data Streams are used to capture large amounts of real-time streaming data.



## AWS Kinesis Data Firehose

Amazon Kinesis Data Firehose is a managed service to load streaming data into data stores, data lakes, and data analytics services. Amazon Kinesis Data Firehose can capture streaming data and put it in S3, Redshift, as well as other services. Amazon Kinesis Data Firehose scales to match the throughput of your data and supports autoscaling and data monitoring.

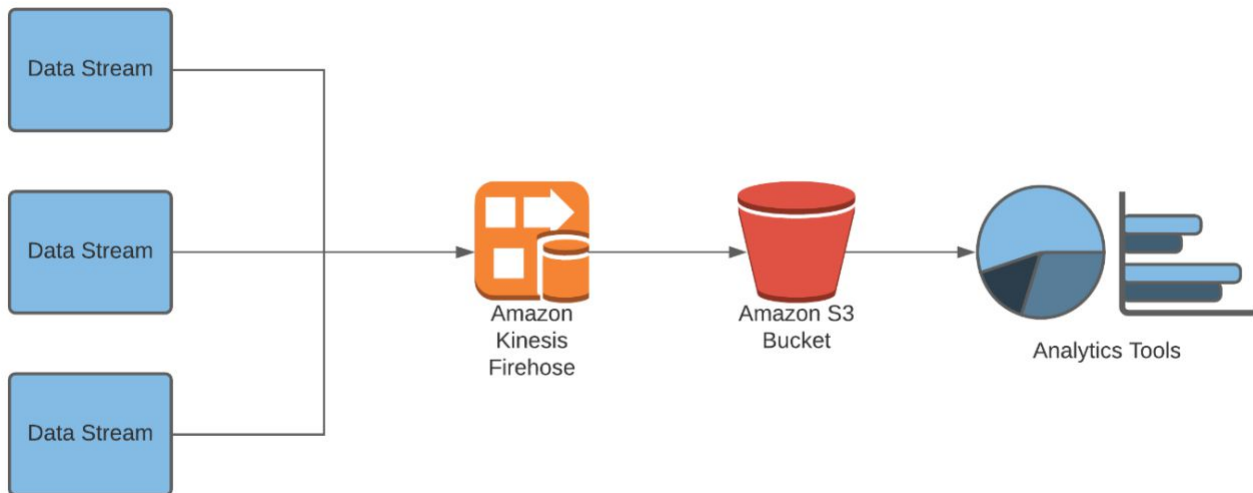
Amazon Kinesis Data Firehose pricing is based upon throughput. Throughput is based upon the number of shards. A shard is considered to be a throughput unit, with a capacity of 1 megabit per second. During setup, the administrator configures the capacity by the number of shards. Shards are increased as capacity requirements increase. A policy can be set up to autoscale the number of shards based upon utilization—up to ten shards per region, per account can be created. If additional shards are required, they can be obtained by contacting AWS support.

## How It Works

1. Streaming data is captured by AWS Kinesis Data Firehose.
2. Streaming data is sent for storage, i.e., S3.
3. Stored data can be analyzed with analytics tools.



The diagram below shows how AWS Kinesis Data Firehose is used to capture large amounts of real-time streaming data and store the data in S3.



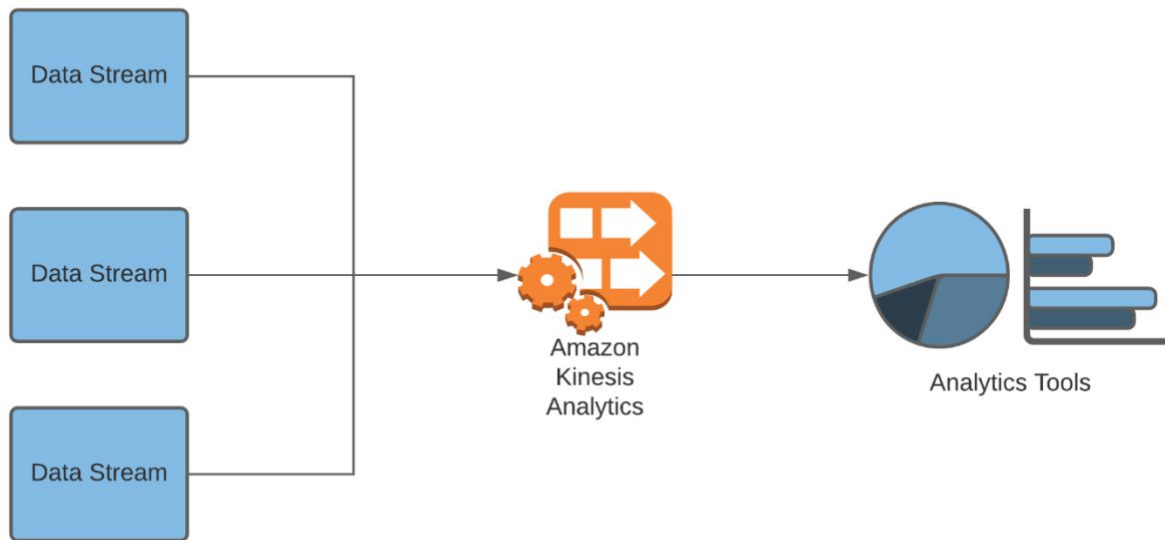
## AWS Kinesis Data Analytics

Kinesis Data Analytics is a managed service to transform and analyze streaming data in real time. Kinesis Data Analytics uses Apache Flink to process data streams. Kinesis Data Analytics can autoscale to meet an organization's needs. Kinesis Data Analytics can be queried with standard SQL queries.

### How It Works

1. Streaming data is captured by Amazon Kinesis Data Streams, Firehose, Elasticsearch, S3, DynamoDB, and other data sources.
2. Amazon Kinesis Data Analytics analyzes data in real time.
3. Amazon Kinesis Data Analytics sends processed data to analytics tools.

The diagram below shows how AWS Kinesis Data Analytics is used to capture and process large amounts of data in real time.

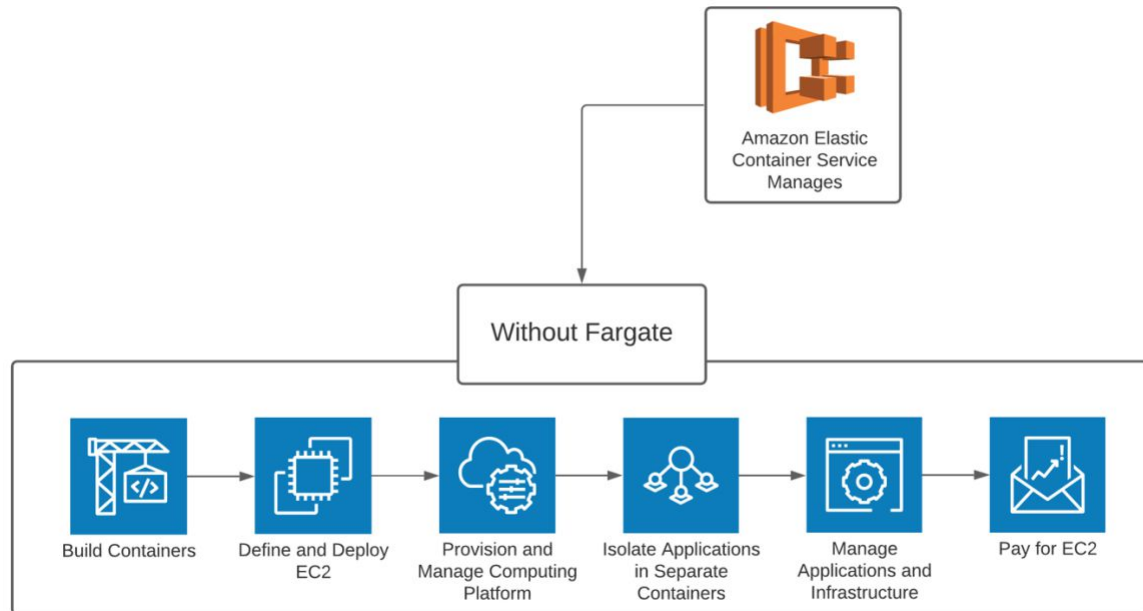


### **AWS Elastic Container Service (ECS)**

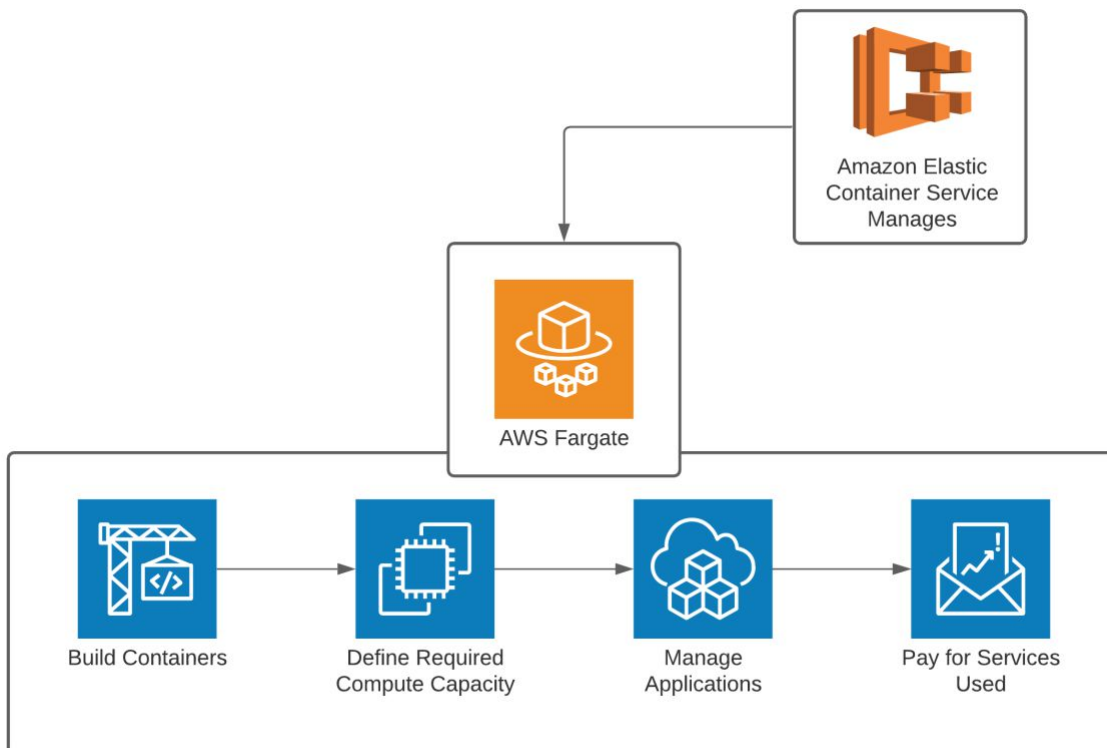
AWS Elastic Container Service (ECS) is a fully managed container management service. ECS is a high-availability (99.99 percent) and high-security container management solution. ECS is deployed in a VPC, which facilitates using AWS security features like network ACLs and security groups. ECS works to manage containers on EC2 or AWS Fargate.

ECS is often used with AWS Fargate, which is a serverless computing engine for containers. This enables ECS and Fargate to create a completely serverless container platform. ECS is used with Fargate, where ECS provides orchestration and management of the Fargate container service. When using ECS with Fargate, there is no need to configure computing instances, install, and manage operating systems, or manage computing instances. Since Fargate is serverless, there is almost limitless scalability.

The diagram below shows how AWS ECS is used to manage containers on the EC2 platform.



The diagram below shows how AWS ECS is used to manage Fargate containers.

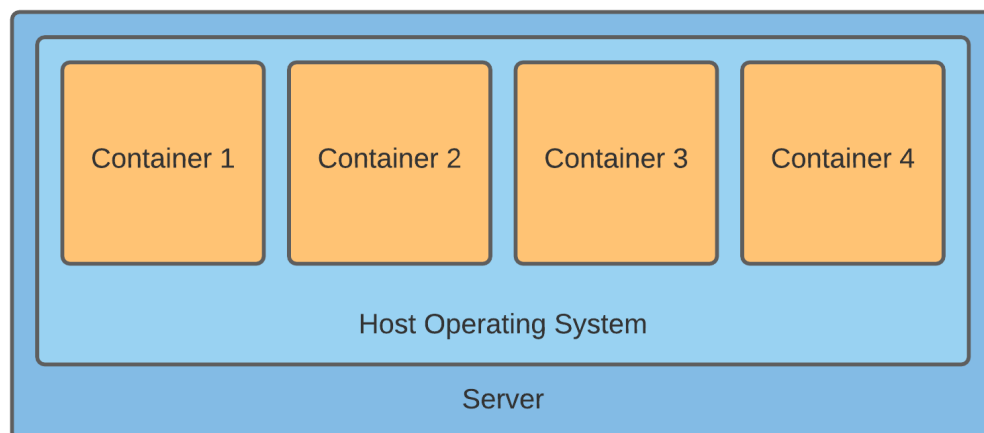


## What Is a Container?

A container is a lightweight, self-contained software package that acts as a modern version of a virtual machine. Virtual machines have a lot of overhead, as they require an entire copy of an operating system to run. By comparison, a container contains just enough of an operating system for the application in the container to run. Since the container runs only a small percentage of operating system packages, a container requires much less memory and CPU resources than a virtual machine. Therefore, many more containers can run at higher performance than virtual machines on a server.<sup>77, 78</sup>

Container images are logically isolated from each other, promoting a secure environment. The host of the containers can support only clients that use the host's operating system. This means Linux containers are hosted on a Linux host and Windows containers are hosted on a Windows host. The reason containers must be on the same operating system is that containers depend upon packages from the host operating system.

The diagram below shows how a server can support numerous containers in a logically isolated manner.



## Choosing Between EC2 and Fargate for Containers

As discussed, containers can be hosted on EC2 instances or Fargate. Both of these approaches have their merit. Choosing the right host is dependent upon an organization's requirements. Please see the list below for guidance on choosing the best container approach:

- If complete control over the device hosting the container is needed, choose EC2.
- If there are specialized requirements and specific customization options required, choose EC2.

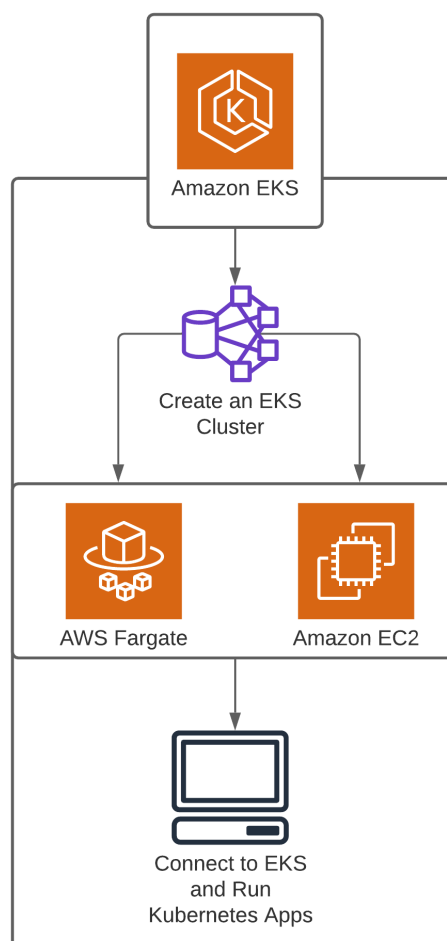
- If a highly scalable platform with minimal management overhead is desired, choose Fargate.
- If near limitless scalability is required, choose Fargate.
- Overall, Fargate is likely a better solution for most customers' needs.

## AWS Elastic Kubernetes Service (EKS)

AWS Elastic Kubernetes Service (EKS) is a fully managed Kubernetes container management service. EKS is a full Kubernetes service, which means Kubernetes containers can be moved to EKS without modification. Kubernetes is an open-source container management service and is the industry standard for containerized applications.<sup>79</sup>

EKS is extremely similar to ECS, but it uses the Kubernetes container platform. Like ECS, EKS is often used with AWS Fargate, creating a completely serverless container platform. EKS can also be used with EC2 in the same manner that ECS can be used with EC2.

The diagram below shows how AWS EKS is used to manage Kubernetes containers.



## **AWS Elastic Beanstalk**

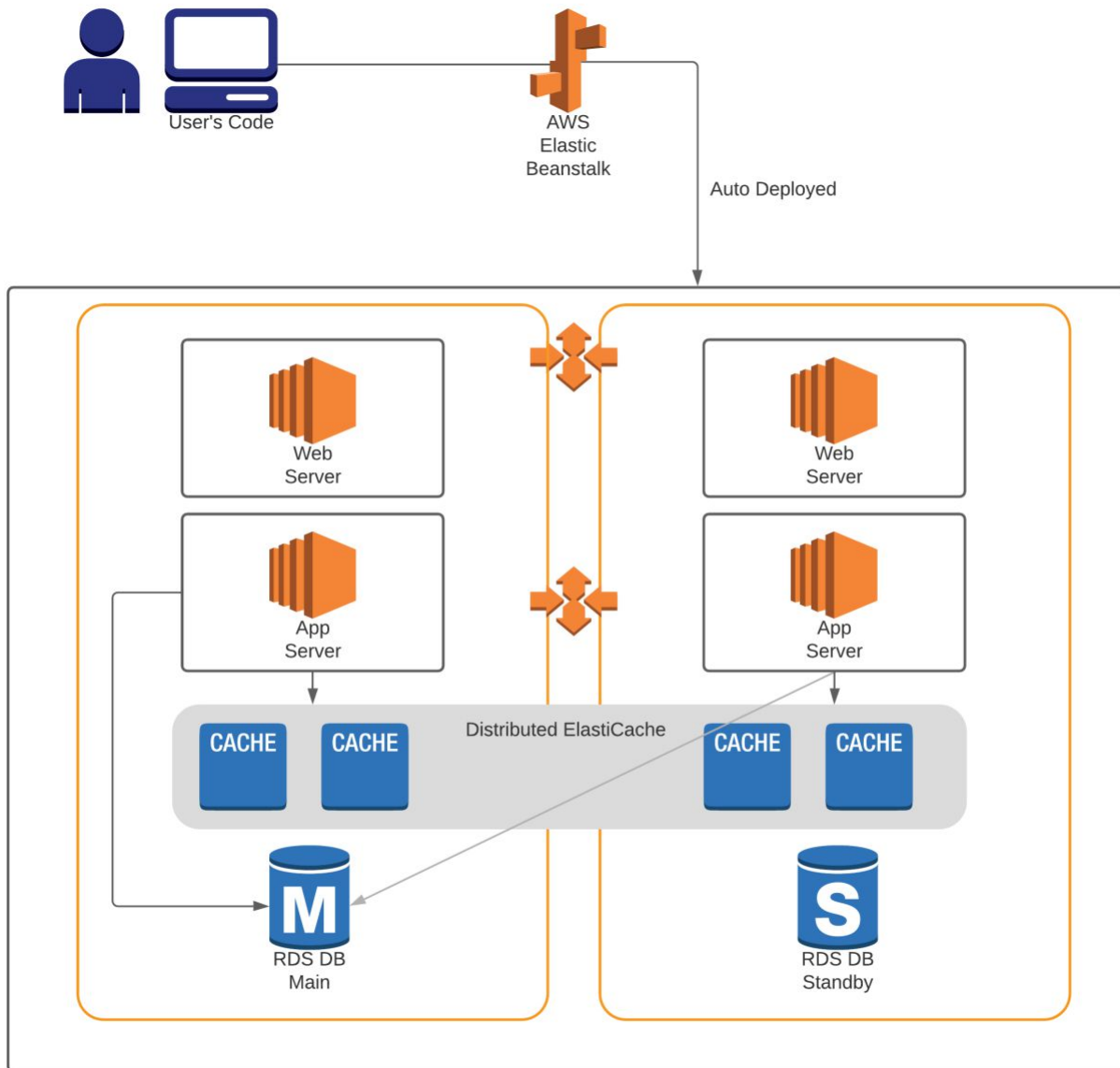
AWS Elastic Beanstalk is a service for provisioning, deploying, and scaling web applications and services. When using Elastic Beanstalk, the administrator simply uploads the code, and Elastic Beanstalk automatically deploys the necessary infrastructure (EC2, containers, load balancers). All infrastructure deployed by Elastic Beanstalk is autoscaling, so it grows with customer requirements. Additionally, infrastructure deployed from Elastic Beanstalk is automatically load balanced.<sup>80</sup>

Elastic Beanstalk provides the necessary tools for web deployment and automatically applies them to the customer, enabling the customer to focus on code development and not infrastructure management. Elastic Beanstalk supports the following programming languages:

- Go
- Java
- .NET
- Node.js
- PHP
- Python
- Ruby

Elastic Beanstalk provisions and manages the environment while allowing the administrator to manage the environment if desired after the computing platform is deployed. Elastic Beanstalk monitors application health and is integrated with CloudWatch logs for performance monitoring.

The diagram below shows AWS Elastic Beanstalk automatically deploying the application environment based upon inputting the organization's code.



## AWS CloudWatch

Amazon CloudWatch is a monitoring service to monitor AWS resources and applications deployed on AWS. CloudWatch provides metrics to monitor performance and troubleshoot issues.<sup>81,82</sup>

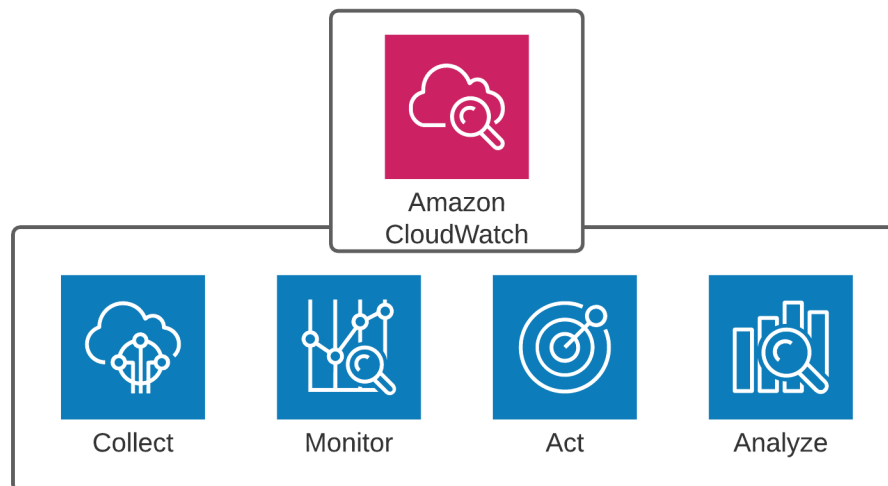
CloudWatch can work with built-in metrics and custom metrics. CloudWatch has several built-in default metrics, which include CPU utilization, disk utilization, and network utilization.

CloudWatch custom metrics can be set up to monitor factors critical to the application's performance, such as memory utilization, API performance, or other metrics.

CloudWatch has a notification service that notifies the organization when certain metrics have been reached. Organizations can set custom metrics and alert notifications. CloudWatch events can also be used to trigger autoscaling, Lambda functions, SNS notifications, actions on containers, and many other functions.

AWS CloudWatch is available in two versions for EC2 instances: basic monitoring and advanced. Basic monitoring automatically provides information every five minutes at no charge. Detailed monitoring provides information every one minute at an additional cost. When using detailed monitoring, it must be enabled at the EC2 instance. As pricing is subject to change, please reference the AWS website for current pricing information for CloudWatch services at <https://aws.amazon.com/cloudwatch/pricing/>.

The diagram below shows AWS CloudWatch can be used for VPC monitoring and optimization.



## AWS Config

AWS Config is a service that enables assessment, auditing, and evaluation of configurations in AWS. AWS Config provides a means to see what changes were made and who made these changes in the VPC. AWS Config provides monitoring of any changes that have occurred. When a change is made, AWS config can send an SNS alert to systems administrators.

AWS Config also provides constant monitoring of configurations and checks configurations against an organization's policies. If a change is made that violates the organization's policy a SNS alert is sent, and a CloudWatch event will occur.

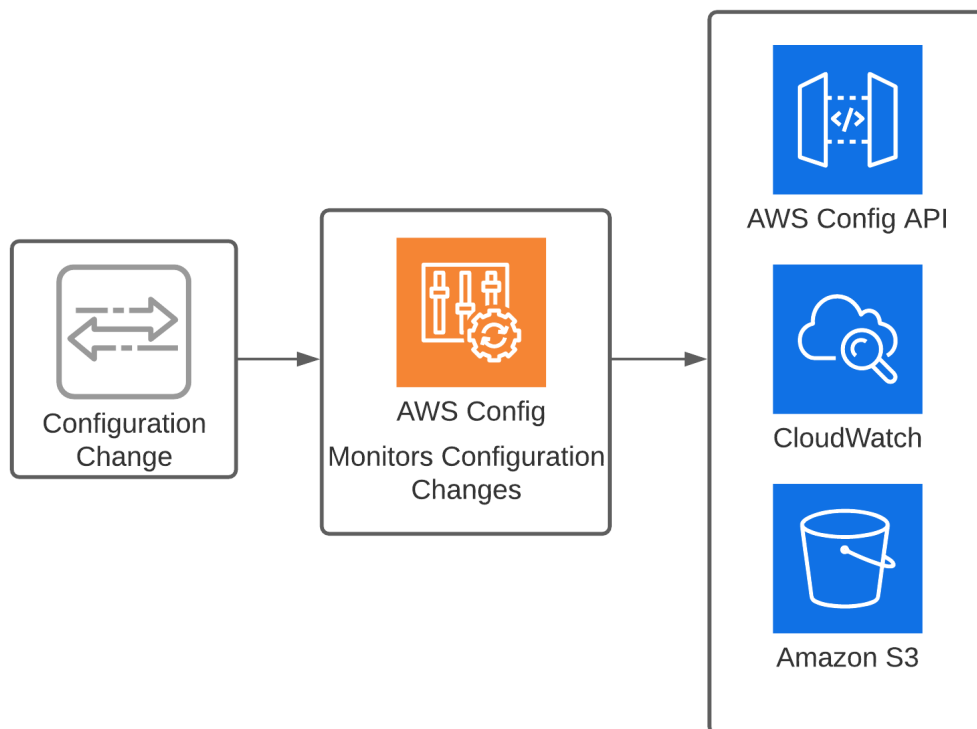


AWS Config provides a means to assist with change management. Config can track relationships between resources, so if changes are made, it will be easy to determine which systems will be affected by the changes. AWS Config can help with troubleshooting, as it can integrate with CloudTrail and track changes made. In this manner if a configuration change causes a problem, AWS config will show which changes should be reverted to go back to a fully functional environment.

### How It Works

1. A configuration change is made.
2. AWS Config notes the change, and records the change in a consistent format.
3. AWS Config will then check the change against an organization's policies.
4. AWS Config will notify the services that can notify the system administrator of configuration changes. Notifications can be sent as a CloudWatch event, SNS, or other AWS service.

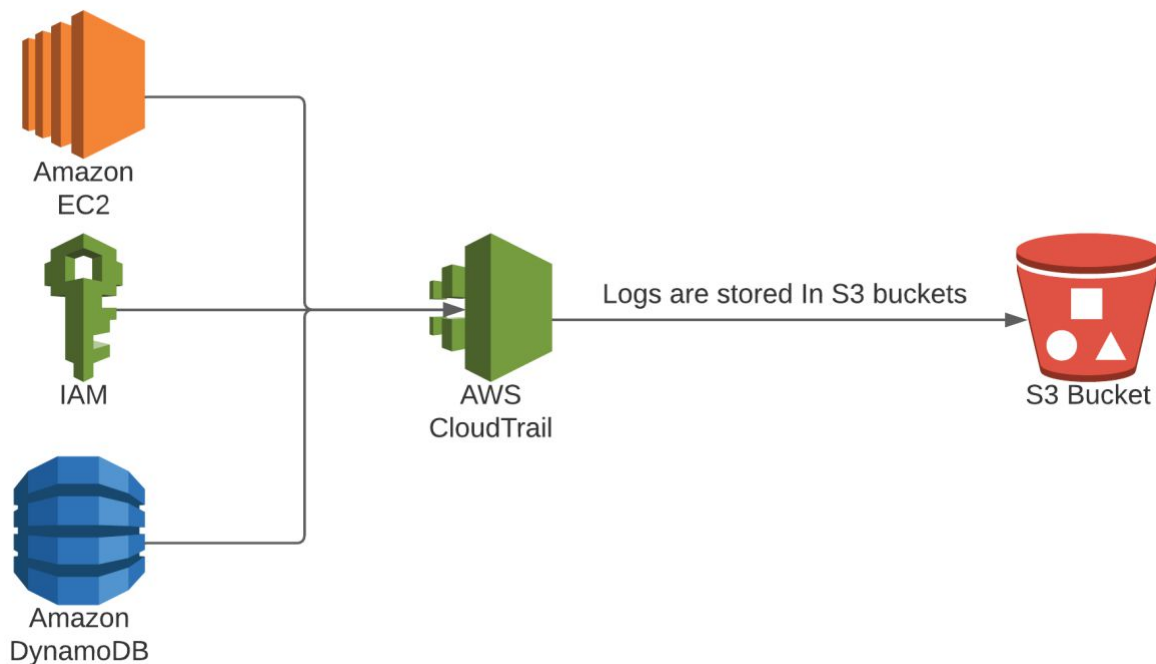
The diagram below shows how AWS Config monitors for configuration changes.



## AWS CloudTrail

CloudTrail is an AWS service that assists with the auditing process. CloudTrail provides an audit log that supports with risk management and compliance endeavors. CloudTrail provides a means to track changes made to an AWS account by user, role, or AWS service.<sup>83</sup>

The diagram below shows how AWS CloudTrail is used for logging and auditing of an organization's VPC.



CloudTrail is enabled when the AWS account is created. To start using CloudTrail, create a trail with the CloudWatch console, CLI, or CloudTrail API. CloudTrail records events, and these events are visible in the CloudTrail console under event history. The CloudTrail event history shows events that have occurred in the last ninety days. Additionally, CloudTrail can be configured to store logs in an S3 bucket for long-term storage.

There are two types of CloudTrail trails can be created, and they can be seen below.

### Local Trail

A local trail is a tied to single region trail. CloudTrail logs are put into a single bucket. This is the default option when CloudTrail is configured by the CLI or API.

## **A Trail that Applies to All Regions**

A CloudTrail can be set up to monitor all regions. This provides the most comprehensive logging and auditing options available. This provides a record of all events that occur inside an organization's infrastructure. This type of trail can help correlate events across an organization's global infrastructure and provide insight on fixing problems.

## **CloudTrail and Compliance Audits**

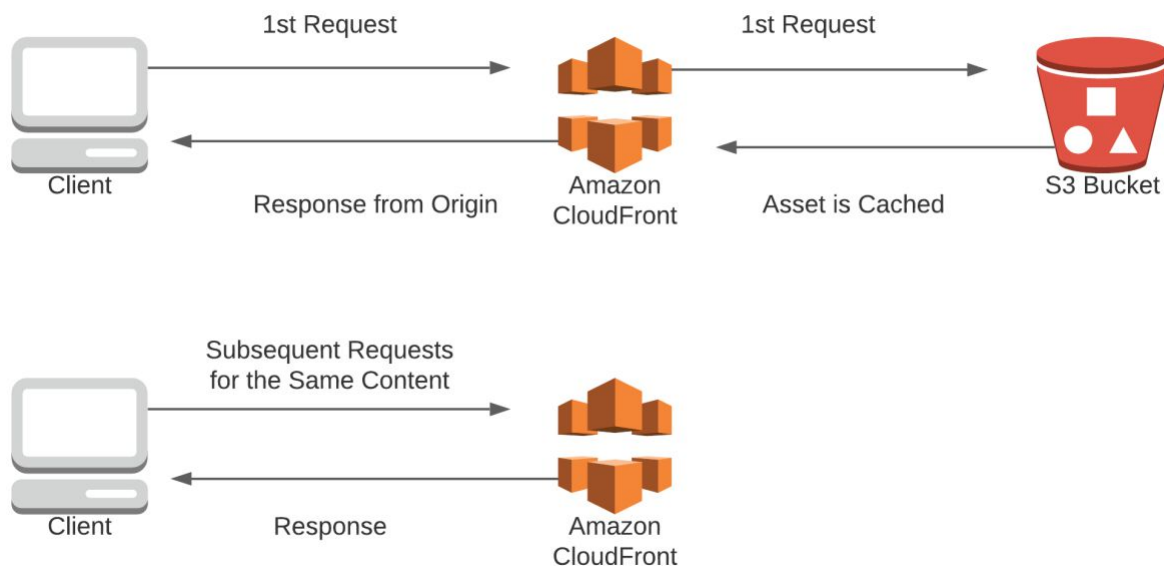
Many industries are highly regulated and have specialized data storage, protection, privacy, and auditing requirements. CloudTrail can be very beneficial in highly regulated environments, as CloudTrail can help show how the organization is adhering to legal requirements.

## **AWS CloudFront**

AWS CloudFront is an Amazon-branded content delivery network. CloudFront can dramatically improve web hosting and is integrated with numerous AWS services. Effectively, CloudFront is a network of caching servers spread throughout the world. When a request is made to a webpage, its location is determined and the web request is sent to the closest CloudFront server. Local CloudFront servers cache website content and speed the delivery to remote locations throughout the world.<sup>84,85,86</sup> The caching server works in the following manner:

1. The web request is sent to the CloudFront caching server.
2. If the website has been requested prior to the cache timeout, the content is sent straight to the user.
3. If the website data is not stored on the cache, the cache reaches out to the original website.
4. When the data is received, the information is stored on the cache until the cache's expiration, and the data is sent to the requestor.

The diagram below shows how CloudFront caching can be used to improve the scalability and performance of web applications.



Caching can assist with website scalability and performance by offloading frequent requests to the cache instead of the actual website. Caching is very helpful for frequently requested content. If the content is very dynamic and user requests are all for new data, then the caching server will not help to improve performance or scalability.

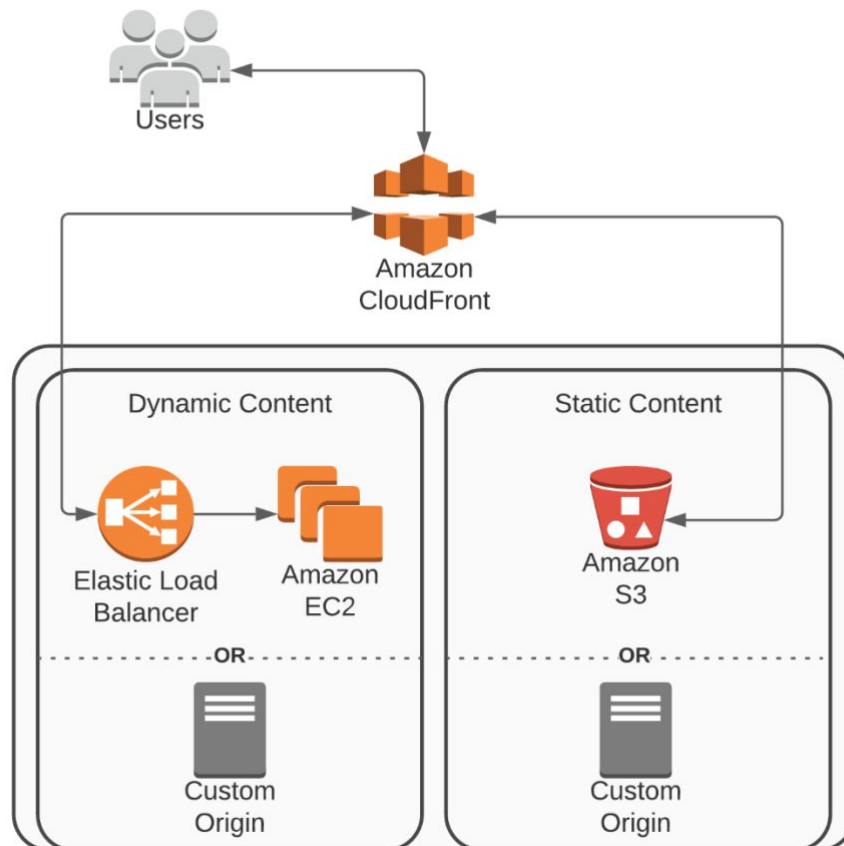
CloudFront integrates with numerous AWS services including S3, EC2, elastic load balancers, and Route 53. CloudFront is typically used as a front end to static websites stored on S3. CloudFront can also be a front end to an EC2-based website as long as an elastic load balancer is part of the architecture.

CloudFront can help website performance through the following mechanisms:

- **Cached content** – The request does not need to go to the web server because it is cached.
- **Global reach** – There are over 217 points of presence for CloudFront. So CloudFront can get content much closer the user's location.
- **Routing efficiency** – When CloudFront is used, requests that go to the original source (i.e., S3 bucket) traverse the AWS backbone and not the public internet. Therefore, performance can be enhanced, as AWS can manage their network for lower latency than when traversing an unknown number of internet service providers.

- Persistent connections – CloudFront maintains connections to the source. This minimizes the number of connections required on the web server, which reduces server load.

The diagram below shows how AWS CloudFront can be used to enhance the performance of a web application with static and dynamic content.



CloudFront can also make a significant impact on an organization's security.

### CloudFront Integrates with Web Application Firewall (WAF)

- WAF adds firewalling capabilities to protect against common web attacks.

### CloudFront Can Help Prevent Distributed Denial of Service Attacks

- CloudFront distributes requests through multiple points of presence.

- CloudFront forwards only legitimate http/https requests to the server that aren't already in the cache. This means the attacker cannot launch a DDoS by sending a large number of invalid requests to the server.
- AWS Shield Standard is included with CloudFront to provide additional layers of DDoS protection.

### **CloudWatch Can Provide Encryption in Transit**

- CloudFront can enforce SSL/TLS protocols.
- CloudFront integrates with the AWS Certificate Manager.
- CloudFront supports Server Name Identification (SNI) as well as custom certificates.

### **Tuning CloudFront**

CloudFront is highly tunable to meet an organization's needs. CloudFront can be modified by changing the Time to Live (TTL) for objects in the cache. The minimum, maximum, and default TTL for objects are configurable options. If problems occur in the cache, it is possible to clear the cache. Clearing the cache is performed via the API or with the command line with the following command structure:

- `aws cloudfront create-invalidation --distribution-id distribution_ID --paths "/"`

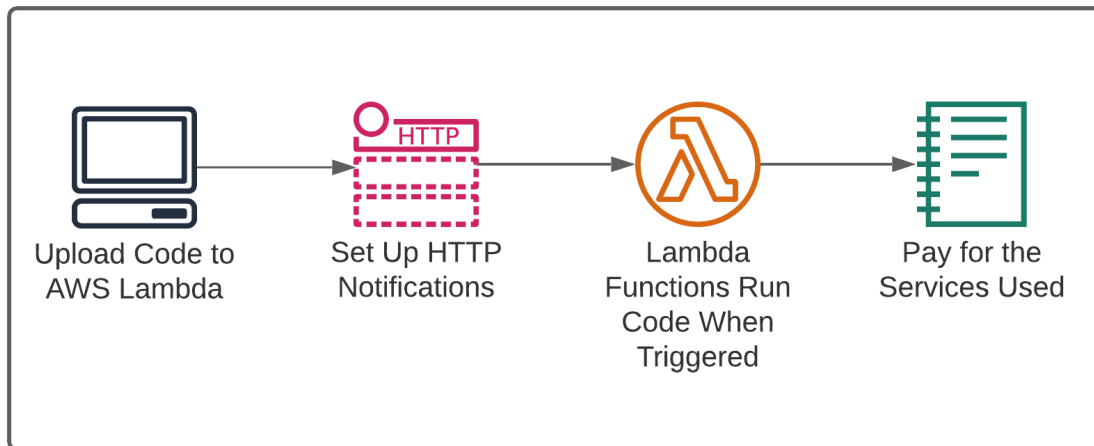
### **AWS Lambda**

AWS Lambda is a serverless computing service to enable automation across an organization's infrastructure. AWS Lambda is useful in many situations where automation can increase the efficiency of technology by decreasing manual intervention. Some examples of automation with the Lambda platform include processing data across multiple systems, patching operating systems, and remediation of security events.<sup>87</sup>

Using Lambda functions is much simpler than deploying custom automation applications. To get started using Lambda, just upload the code for the Lambda function. Since Lambda is serverless, there is no need to manage servers and operating systems. Lambda supports the following programming languages:

- C#
- Go
- Java
- Node.js
- Python
- PowerShell
- Ruby

The diagram below shows how Lambda functions are deployed on the AWS platform.

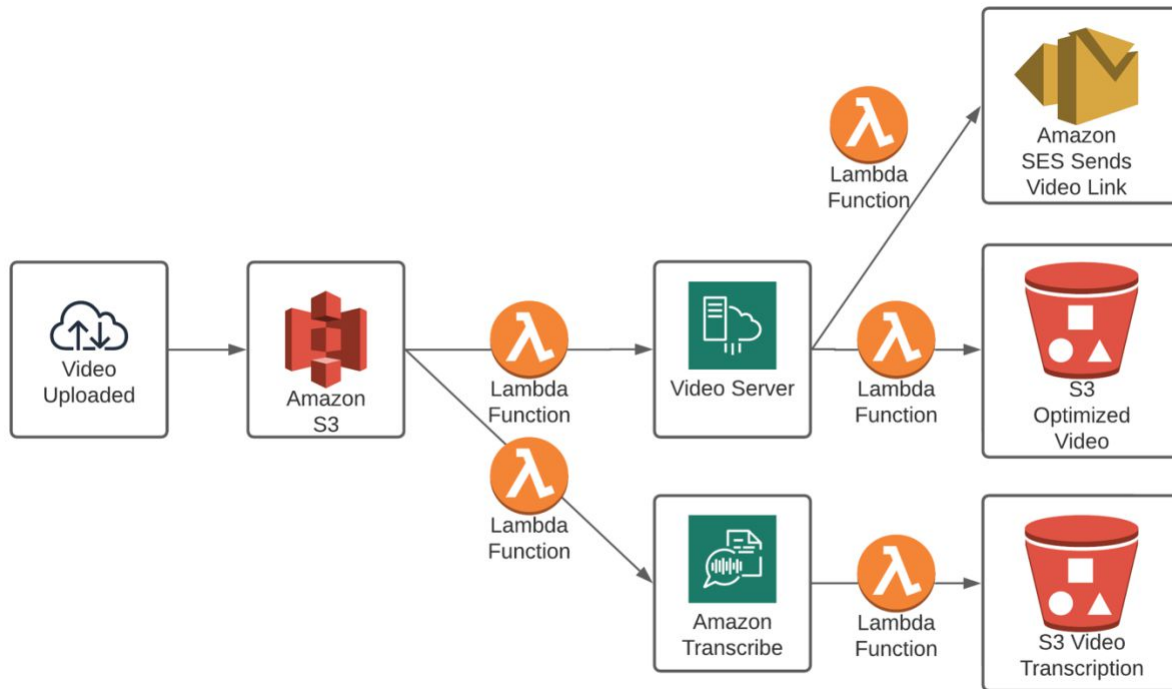


Lambda is stateless, meaning that once the function is performed the function has been completed. If additional functions are needed, it will be necessary to set up additional Lambda functions. Lambda functions can be run in response to events in a VPC. A sample video optimization using Lambda can be seen below:

In this example, there is a video-processing application that optimizes and transcribes a video after being uploaded into S3 using Lambda functions.

1. A new video is uploaded into S3.
2. When S3 detects a new video, a Lambda function is triggered to inform the transcription application that a new video is ready to be transcribed.
3. The video gets transcribed.
4. After the video is transcribed, a Lambda function triggers AWS SNS/SES.
5. An email is then sent to the customer, telling them their video is ready for download.

The diagram below shows how AWS Lambda can be used to automate workflows on the AWS platform.



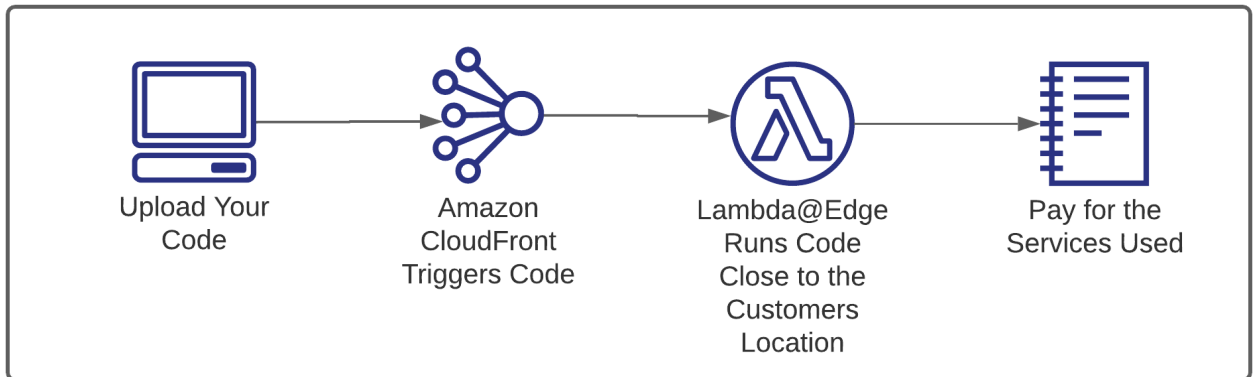
### AWS Lambda@Edge

AWS Lambda@Edge is a serverless CloudFront feature. Lambda@Edge works with the CloudFront content delivery network. Lambda@Edge enables the content to be closer to the customer and achieve higher performance. Additionally, Lambda@Edge allows for running Lambda functions closer to the user. Setting up is a matter of the following:

1. Upload code to a Lambda function.
2. Set up the Lambda function to be triggered by CloudFront.
3. The Lambda@Edge code is run where your users are located.



The diagram below shows how AWS Lambda@Edge can be used to perform functions close to the customer's location.



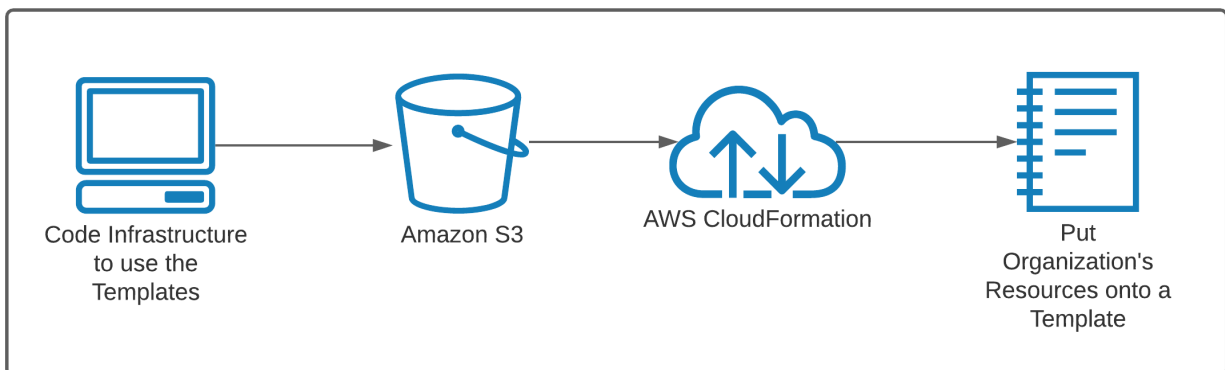
## AWS CloudFormation

CloudFormation is a means to template known good configurations of an organization's services. For example, if an organization has a common application that requires the configuration of several servers with specific patches, a CloudFormation template can make sure all new servers are properly configured. CloudFormation therefore helps you provision applications in a safe and repeatable manner. CloudFormation templates can be made with simple text files or via supported programming languages. AWS Cloud Formation templates are available through a multitude of options. CloudFront can deploy your templates across your infrastructure by rebuilding applications or building new ones.<sup>90</sup>

## How It Works

1. Develop the code for the organization's infrastructure.
2. The code can be made from a template or from scratch in either JSON or YAML format.
3. Store your code either locally or on S3.
4. Use CloudFormation with the customized code either with the CloudFormation console, CLI, or API.
5. CloudFront will then provision your systems.

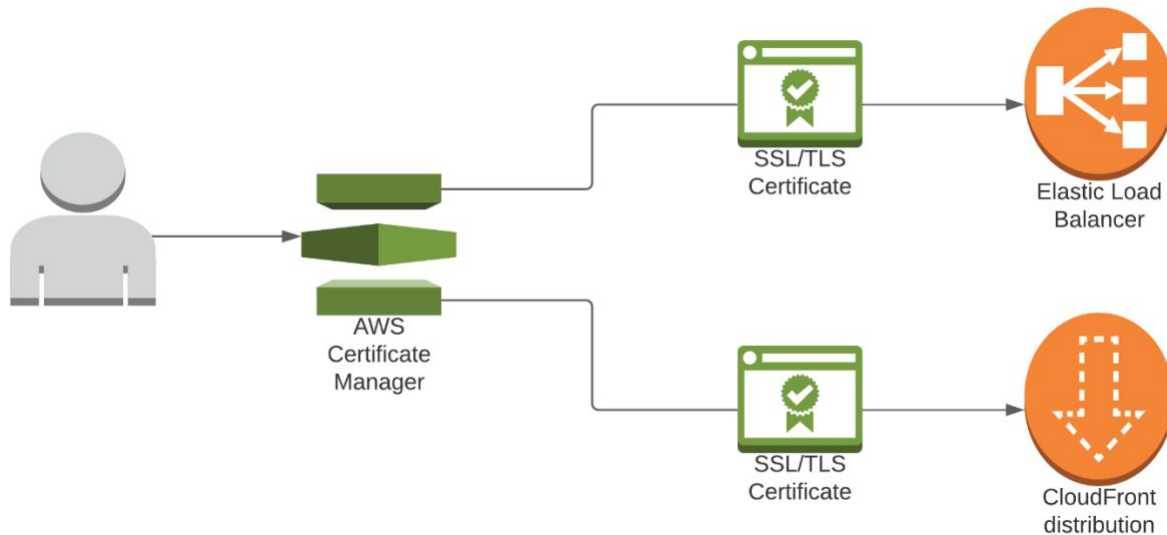
The diagram below shows an example of CloudFormation used to template an organization's services.



## AWS Certificate Manager (ACM)

AWS Certificate Manager is a service for SSL/TLS certificates. AWS Certificate Manager makes it easy to provision, manage, and deploy certificates either publicly or privately. It allows users to deploy certificates on AWS resources quickly and effectively. It provides free public and private certificates to AWS services such as ELBs and API Gateways. AWS Certificate Manager provides a means to obtain certificates to websites, promoting safe and secure connections. AWS Certificate Manager is a platform to manage all of your certificates in the AWS cloud centrally.<sup>91,92</sup> There are two options when deploying certificates: the default certificate manager and ACM private CA.

The diagram below shows how the certificate manager can be used to obtain SSL/TLS certificates.



### Default AWS Certificate Manager (ACM)

AWS Certificate Manager is for customers who want encryption and security using TLS. The certificates are deployed through ELBs, CloudFront, and API Gateways in order to make communication secure.

### ACM Private CA

Private CA is for communication within the organization. Private CA can issue certificates for users, computers, applications, services, servers, and more devices throughout the organization. Private CA certificates are for use internally and not on the internet. Private certificates also come at an additional cost.

## Chapter 9

### Cost Optimization

#### Financial Differences Between Traditional Data Centers and Cloud Computing

Migrating from the data center to the cloud can have a profound effect on an organization's technology costs. In most scenarios a move to the cloud will have lower total cost of ownership than with traditional data centers. This is because traditional data centers have large costs to purchase equipment, build data centers, and staff management of the data center and IT systems. These are heavy capital expenditures (CAPEX) with a moderate degree of operational expenses (OPEX). The list below shows the typical capital and operational expenses with traditional data centers. With a traditional data center, the organization purchases the following equipment (CAPEX):

- Physical servers
- Routers
- Switches
- Firewalls
- Load balancers
- Racks
- Power distribution units
- Generators
- UPS
- Data-center cooling

Additionally, there are moderate OPEX costs associated with traditional data centers. The primary OPEX costs are:

- Large IT staff
- Electric bills
- WAN connections
- Internet connections

#### Optimizing Technology Costs on the AWS Cloud

Moving to the cloud changes the cost structure completely. In the cloud computing environment, there is really minimal to purchase, so CAPEX is very low. However, the organization pays every time cloud services are used, and usage can get quite expensive. Therefore, with cloud computing, while CAPEX is low, OPEX is high. Generally speaking, moving to the cloud will have a lower total cost of ownership than with traditional data centers.

Furthermore, the better the cloud architecture is designed, the lower the total costs for cloud computing. There are five steps to lowering the cost of cloud computing:

### **Step One**

- Provision only the resources that need, as you pay for all resources used.
- Monitor your systems so you can get insights into the proper size of compute and network resources.

### **Step Two**

- Properly size resources.
- Plan and size equipment based upon average use and not peak usage. Cloud computing allows autoscaling, so you don't need to overprovision in advance as in a traditional data center.
- Leverage means to decouple systems in the architecture when possible. For example, an SQS queue can dramatically decrease the spikes in the system, allowing for less expensive resources to be used in the architecture.

### **Step Three**

Purchase the right computing platform. Know when it's best to use On-Demand Instances, Reserved Instances, and Spot Instances, as they can have a dramatic effect on costs. These options are discussed below.

#### **On-Demand Instances**

- Are the most expensive at a pure pricing level.
- Provide instant access to computing power.
- Ideal when you don't know the exact amount of computing power required but need flexible options.
- Highly reliable, in that on-demand instances won't be terminated like a spot instance when AWS pricing changes.
- Promote scalability by facilitating autoscaling.
- Ideal for situations when you have a temporary application or when you don't know how long the application will be used.

#### **Spot Instances**

- Are the lowest cost option for computing power within the AWS platform.
- Let the customer bid for unused computing power in AWS.
- Pricing changes constantly based upon AWS capacity and current bids by other organizations using the AWS platform.

- Are not for critical workloads, as they can be shut down by AWS if the price for spot instances changes.
- Are optimal for batch jobs that are not critical and do have a means to restart the processing if the system is shut down.

### **Reserved Instances**

- Offer discounted service when an organization makes a guaranteed purchase of computing capacity for a period time.
- Provide pricing based on a contract—with the longer the contract, the greater the discount.
- Are ideal for an application with a known capacity and a known duration for which the organization will use the computing platform.

There are three types of reserved instances.

#### **Standard Reserved Instances**

- Are the lowest cost option.
- Are optimal for a long-running application.

#### **Convertible Reserved Instances**

- Convertible reserved instances are reserved instances with flexibility to change the size of computing instances.
- With convertible reserved instances, an organization purchases a computing platform based upon need. If the organization needs to resize its computing instances, it has the flexibility to change.
- Convertible reserved instances offer flexibility but with higher costs than standard reserved instances.

#### **Scheduled Reserved Instances**

- Scheduled reserved instances are reserved instances for computing platforms that are used by organizations with frequent and periodic needs for computing power.
- Scheduled reserved instances are optimal for critical workloads that happen periodically. For example, a batch job that needs to run uninterrupted every weekend for forty-eight hours straight.
- Scheduled reserved instances cost more than standard reserved instances, but they enable discounted pricing for periodic workloads.

To optimize costs for computing instances, purchase the computing platform that will be most cost effective based upon the organization's needs. Costs can be best optimized by using a combination of on-demand, reserved, and spot instances based upon an organization's needs.

#### **Step Four**

- Leverage managed services and serverless options to minimize time and costs spent managing computing instances.

#### **Step Five**

Step five is about managing data transfer costs.

- AWS charges for data sent between regions. Being mindful of cross-region data charges can make a big difference in an organization's costs. S3 cross-region replication can assist with data transfer costs when there is a large amount of data being requested across regions.
- Leverage CloudFront to reduce data transfer costs between regions, as content will be cached and served locally.
- Use the right connection to AWS. Data in and out of a VPC over a direct connection can be lower cost than VPN if large amounts of data are being transferred.

#### **AWS Budgets**

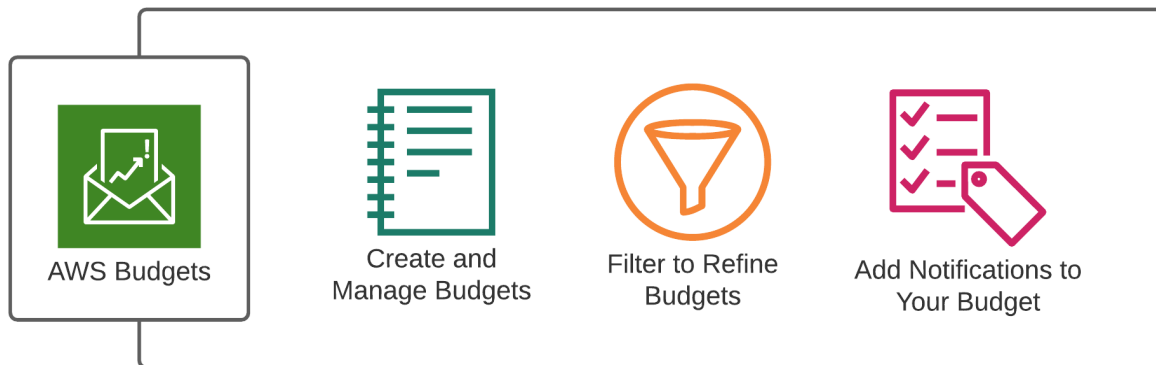
Another means to control costs is to create a budget and stick to that budget. AWS budgets help the organization stick to a budget with budget notifications.<sup>93</sup>

#### **How AWS Budgets Work**

- The organization creates a budget and sets custom alerts.
- The budget is created in the AWS Management Console or within the AWS Billing Console.
- When an organization gets close to exceeding its budget, an alert is sent.

Budgets can ensure that an organization adheres to its budget. Additionally, the budget alarms can help an organization plan for future optimizations of their network. For example, an organization may find that due to use, reserved instances may enable large cost savings. This enables the organization to optimize cloud computing expenses.

The diagram below shows how AWS budgets can be used to help an organization manage its AWS cloud computing expenses.



## AWS Trusted Advisor

AWS Trusted Advisor is an online tool to help an organization optimize its spending on the AWS platform.<sup>94</sup>

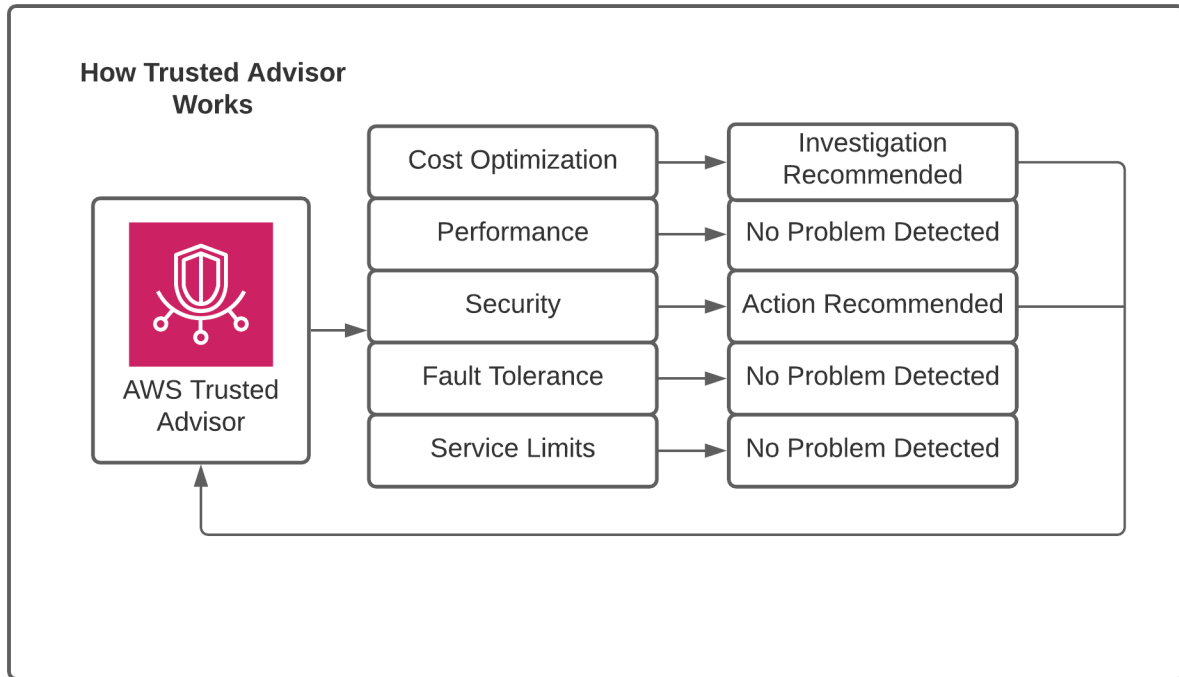
### How AWS Trusted Advisor Works

1. AWS Trusted Advisor scans and evaluates an organization's infrastructure.
2. AWS Trusted Advisor compares an organization's infrastructure to AWS best practices and provides recommendations. These recommendations can improve performance, security, availability, and system costs.
3. The organization evaluates the Trusted Advisor recommendations.
4. The organization implements the appropriate recommendations from AWS Trusted Advisor.
5. This should reduce costs and or improve system performance.

There are two versions of Trusted Advisor for clients of the Developer Support Plans and Business Support Plans. Organizations with AWS Basic and Developer Support Plans have access to six security checks and fifty service limit checks with Trusted Advisor. Organizations with Business Support Plans or AWS Enterprise Support Plans receive fifteen Trusted Advisor checks (fourteen cost optimization, seventeen security, twenty-four fault tolerance, ten performance, and fifty service limits).



The diagram below shows how AWS Trusted Advisor is used to help an organization optimize its AWS infrastructure.



## Chapter 10 Building High Availability Architectures

### What Is Availability?

Availability refers to the service being available for use when you need it. A high-availability infrastructure is highly likely to be ready when needed. Designing for high availability can become extremely expensive, depending upon the availability required.<sup>95</sup>

In general, there are four levels of availability:

- 99.0 percent
- 99.9 percent
- 99.99 percent
- 99.999 percent

Most people would consider levels of availability of 99.9 percent or greater to be a highly available network. Many organizations require much higher levels of availability. For example, service providers, banks, health care organizations, and other organizations that are completely dependent upon technology—with serious consequences if the systems are not operational—require 99.999 percent or greater availability.

The diagram below shows the typical availability metrics and associated downtime per year.

Availability Level	Maximum Downtime Per Year
99.000%	3.65 days
99.900%	8.76 hours
99.990%	52.6 minutes
99.999%	5.25 minutes

### Building a High Availability Network

Building a high-availability network is based upon the key tenant of no single points of failure. This means complete redundancy is required in all aspects of the computing environment. Necessary redundant services are:

- Redundant power
- Redundant cooling for servers
- Redundant network connections
- Redundant service providers
- Redundant routers
- Redundant switches

- Redundant servers
- Redundant load balancers
- Redundant DNS
- Redundant storage
- Redundant locations
- Redundant applications, i.e., databases

## **Change Management**

A strong change management program is required for high-availability systems. Configuration changes can have a major impact on system performance, especially if configuration mistakes occur. Therefore, prior to making any changes across an organization's systems, all stakeholders need to be notified of prospective changes. All stakeholders need to evaluate that any changes made will not affect the systems they manage. Additionally, all stakeholders need to agree on a time for configuration changes. Configuration changes should be made at a time when the system is minimally used—ideally a time when user access and system utilization are at their lowest levels.

## **High Availability in the Cloud**

Building a high-availability system in the cloud is much simpler than with the traditional data center. This is because AWS maintains the key elements of high availability in the architecture, including:

- Redundant power
- Redundant cooling
- Redundant connections to the internet and across the backbone
- Redundant routers and switches

Since AWS natively performs many of the parts of a high-availability architecture, only a subset of redundancy is required, and those elements can be seen below. Whenever possible, place the platform in multiple availability zones. These elements include:

- EC2 compute instances
- Databases
- Elastic load balancers
- DNS or Route 53
- NAT gateways
- Storage

Another key component of high-availability design are the connections from the organization to the AWS VPC. For most clients this will include a direct connection to AWS and a VPN backup. Organizations requiring even higher levels of availability and performance might have a primary

direct connection, a backup direction connection, and a VPN backup to the direct connections. Ideally all connections to AWS are across multiple service providers, so that if a service provider were to have an outage, the backup connections will remain intact. Furthermore, on the customer end, redundant connections should be placed on redundant routers.

### **High Availability Requires High Security**

High-availability environments require a strong security posture. A strong security posture is required because if security is compromised, it can have a major impact on system performance and availability. Key components of a high security for high-availability architecture are below.

#### **Principle of Least Privilege**

- Allow users access to only the systems they need to perform their job.
- Use strong IAM policies to limit the users to the minimum services necessary.

#### **Containing Problems**

- Limit blast radius of an application by using AWS Organizations.

#### **Keep Unwanted Traffic Out**

- Keep unwanted network traffic out using network ACLs.
- Configure services with security groups so that only desired traffic is allowed.
- Use Amazon WAF for firewall, AWS Shield for DDoS, and IDS/IPS for intrusion protection.

#### **Physical Security**

- Equipment accessing the AWS network should be secured to prevent unauthorized access.

#### **Passwords and Authentication**

- Allow only strong passwords to be used, and rotate them frequently.
- Use multifactor authentication.
- Use temporary passwords or tokens whenever possible.

#### **Data Privacy**

- Encrypt data to ensure its privacy from unauthorized users.
- Encrypt data both at rest and in transit.

## **Logging and Monitoring**

There should be constant monitoring of the systems for:

- System alerts
- Security breaches
- Usage
- Performance

## **Other Essential Security Components**

- Disable all unused and unnecessary services on systems.
- Allow only approved services to be used.
- Template known configurations with CloudFormation to be sure new systems meet security requirements.

## **Additional Means to Increase Availability**

There are some additional methods to increasing availability. Using systems and services that are designed to enhance availability is one method. DNS and load balancers with health checks are purposely designed to enhance availability and performance. Elasticity and autoscaling features help to make sure that applications are always usable and don't become unstable with high demand.

Availability can also be increased by decoupling applications using services like SQS. Decoupling components of the architecture can promote system integrity and availability by enabling functionality when one or more parts of the system are unavailable. For example, adding an SQS queue in front of a database can keep the system functioning and not lose messages during a database outage. Additionally, use services designed to lower overall system load so the systems will be available and not busy when needed. An example is using redundant caching in front of web servers.

## Chapter 11 Passing the AWS Exam

### The AWS Certified Solutions Architect Exams

AWS exams can be quite challenging, as there is a lot of material covered in the exams. It is our experience that these exams can cover an incredibly wide range of topics. Therefore, we advise strong preparation so you are in the best position to answer challenging questions. There are some key elements of AWS questions that make them challenging to answer, including:

- Questions can be extremely wordy and challenging to understand. Don't be surprised if you need to read the questions two or three times each prior to understanding the question.
- There may not be a correct answer among the options presented in the question. Choose what feels like the best answer.
- The questions frequently do not provide sufficient information to answer them without a lot of guessing and interpretation.
- Don't overthink the questions. If you have an extensive IT background, when you read the questions you will see multiple options that could be right because there are many ways to accomplish the same goals. For the exam, forget past experiences and remember the AWS way. It is an AWS test, so think in terms of AWS.
- Sometimes the answers on the questions differ by only one word, so read very carefully.
- Sometimes AWS questions have a *NOT* in them. For example, *Which one of these options is not required under these circumstances*. Read carefully so you don't miss the *NOT*.
- Don't spend too much time on each question, as you can go back to them and may find the answer elsewhere on the exam.

### Recommendations

We have several recommendations to help you pass the exam.

Read this book in its entirety. We spent a lot of time putting the materials in a short and easy-to-read format. Since we wrote for readability, even small sections of the book may contain a lot of information.

Look very carefully at the diagrams contained in this book. We designed those diagrams to help explain AWS concepts.

Read the AWS white papers. In our experience a lot of AWS questions are taken from information contained in AWS white papers.

Use practice tests. We feel that knowledge is only part of passing the AWS exams. A lot to passing the AWS exam is learning to read and understand the complexity of AWS questions.

Practice tests are a great way to assess your knowledge, reinforce tricky concepts, and get you used to the way AWS asks questions. When you can score a 95 percent or better on a practice test, we recommend scheduling the exam.

Avoid brain dumps. We strongly advise against using services that claim to have actual exam questions with answers. First, it is cheating and unethical. Secondly, when we find AWS questions on the internet, often the answers provided are incorrect. Additionally, if AWS suspects someone of cheating, AWS can pull any certifications they receive.

Don't cram the day of the exam. If you have prepared properly, you will have the tools necessary to pass the exam. We believe that for many, the biggest challenge is actually reading and understanding the questions. If you are tired from cramming, the questions may be challenging if not impossible to understand.

### **The Day Before the Exam**

The day before the exam make every effort to get a good night's sleep so you will have the energy and concentration for a three-hour exam. Eat healthy foods the day before and the day of the exam so you will be at your best. Avoid alcohol or any substance that can affect thinking or judgment, unless prescribed by your physician for a health condition.

### **The Day of the Exam**

We recommend taking it easy on the day of the exam. As we have previously stated, the AWS questions are very challenging to understand, so it's best to arrive feeling refreshed and not fatigued. Arrive early for the exam, whether its online or in person. Online exams can take time to set up, as there are photos to be taken and many other setup components. This can take thirty minutes or more in certain situations. When taking an in-person exam, there can be traffic problems, parking problems, or tech problems, so be early so you don't lose valuable exam time. Remember to have a valid photo ID when you take the exam.

### **Thank You**

Thank you for reading this book. We are excited for your journey into the world of cloud computing. We are always excited when our students pass a new certification exam. Please let us know about your success by sending an email to [elitetechcareers@gmail.com](mailto:elitetechcareers@gmail.com).

## Practice Exam

Below is a sample practice exam. Please note that not all questions are grammatically perfect. This is intentional to make the questions feel more like the actual exam. As stated in the book, we strongly recommend purchasing additional practice exams and scheduling your exam when you can consistently score above 95%. This test is a cross between a certified solutions architect associate and professional exam. We want our students to be optimally prepared for their careers and their exams so we were sure to include extra content to help you be maximally prepared.

1. An organization has an application in their on-premises datacenter that stores multiple 5GB files per day in S3. Recently many of these uploads have been failing. The customers datacenter is geographically close to the S3 region where they store their data. What can the organization do to increase the reliability of data transfers to AWS without incurring substantial costs?
  - A) Upload data to S3 using transfer acceleration
  - B) Upload data as part of a multipart upload
  - C) Upload data to glacier and then copy to S3
  - D) Upgrade to a faster connection to the internet
  
2. An organization has users who upload a large number of files (each file is about 30MB) each day to S3. Recently many of these uploads have been very slow. The organizations employees are spread throughout the world. What can the organization do to increase the performance of these transfers to S3.
  - A) Upload data to S3 using transfer acceleration
  - B) Upload data as part of a multipart upload
  - C) Upload data to glacier and then copy to S3
  - D) Upgrade to a faster connection to the internet



3. You have deployed a three-tier architecture in a VPC with a CIDR block of 172.16.1.0/28. The initial deployment has two web servers, two application servers, two database servers and a custom server deployed on an EC2 instance. All web, application servers and database servers are spread across two availability zones. Additionally, there is an ELB and DNS using route 53. Demand for the application grows and autoscaling is not able to keep up with demand, as autoscaling stops after adding two additional servers.

Why did autoscaling stop adding instances? Choose two

- A) AWS reserves the first four and last IP address, so there are not enough addresses to launch additional instances
  - B) There should be 15 usable addresses in a /28 subnet so there must be a configuration error
  - C) Autoscaling is configured improperly
  - D) The customer needs a larger subnet i.e /27 instead of a /28
4. When using IAM a group is regarded as a:
- A) Collection of AWS accounts
  - B) Collection of AWS users
  - C) Collection of computing instances
  - D) Link between a database and a compute instance
5. You have set up an autoscaling policy to scale in and out. You would like to control which instances are stopped first. How would you configure this?
- A) IAM Role
  - B) A termination policy
  - C) Route 53
  - D) DynamoDB

6. What are characteristics of VPC subnets? Choose 3

- A) Each subnet maps to a single availability zone
  - B) Subnets are spread across availability zones
  - C) Instances in a private subnet can access the internet if they have an elastic IP
  - D) The smallest subnet on AWS is a /28
  - E) With the default configuration all subnets can route between each other in a VPC
7. In a CloudFormation template, each identified resource includes the following:
- A) An operating system and AMI
  - B) A dedicated host and hypervisor
  - C) Logical ID, resource type and resource properties
  - D) Physical ID, resource type and resource properties
8. Every time you attempt to delete an SSL certificate from the IAM certificate store you keep getting the error "Certificate: <certificate-id> is being used by CloudFront". What is the most likely reason for this error?
- A) SSL certificates cannot be deleted
  - B) You do not have sufficient IAM permissions
  - C) CloudFront is not set up properly
  - D) Prior to deleting the SSL certificate, its necessary to rotate SSL certificates or revert to the default CloudFront certificate
9. You plan on launching a new product. There is tremendous buzz and enthusiasm around the product launch, but you don't know exactly the demand. Orders will be sent to the database so it's critical that writes to the database will not be lost. What is the best way to be sure orders are not lost when being written to the database?
- A) Use a Microsoft SQL server cluster

- B) Use DynamoDB with the max write capacity
  - C) Use an Amazon Simple Queue Service (SQS) to store orders until written to the database
  - D) Add additional read replicas
10. An organization wants autoscaling to scale out at 65% CPU utilization and scale in at 35 percent. How can the organization make sure this occurs?
- A) Use auto-scaling with the default policy
  - B) Use autoscaling with a policy
  - C) Use CloudWatch alarms to send an SNS message to auto scale
  - D) It is not possible to scale at these CPU levels
11. Which of the following EBS volume types is ideal for applications with light or burst I/O requirements?
- A) Provisioned IOPS
  - B) EBS General Purpose SSD (gp2)
  - C) EBS Throughput Optimized HDD (st1)
  - D) EBS Cold HDD (sc1)
12. Which of the following EBS volume types is ideal for applications requiring the lowest latency possible?
- A) Provisioned IOPS
  - B) EBS General Purpose SSD (gp2)
  - C) EBS Throughput Optimized HDD (st1)
  - D) EBS Cold HDD (sc1)

13. Your company is getting ready to do a major public announcement about a highly anticipated new product. The website is running on EC2 instances deployed across multiple Availability Zones with a Multi-AZ RDS MySQL Extra Large DB Instance. There are a large number of read and writes on the database. After examination you discover that there is read contention on RDS MySQL. How can you best scale in this environment?

- A) Deploy ElastiCache in-memory cache running in each availability zone
- B) Add an SQS queue in front of the RDS MySQL database
- C) Increase the RDS MySQL instance size and Implement provisioned IOPS
- D) Add an RDS MySQL read replica in each availability zone

14. An organization has a requirement for the highest throughput and lowest latency storage option. The organization is willing to trade redundancy for performance. What is the best RAID option for this situation?

- A) Raid 0
- B) Raid 1
- C) Raid 5
- D) Raid 10

15. An organization has a requirement for the highest throughput and lowest latency storage option with complete redundancy. What is the best RAID option for this situation?

- A) Raid 0
- B) Raid 1
- C) Raid 5
- D) Raid 10

16. An organization requires a solution that provides complete redundancy. Speed is not a concern. What is the best RAID option?

- A) Raid 0
- B) Raid 1
- C) Raid 5
- D) Raid 10

17. Your company is developing a next generation wearable device that collects health information to assist individuals with adopting healthy lifestyles. The sensor will push 25kb of health data in JSON format every 2 seconds. The data should be processed and analyzed, and information should be sent to the individuals primary care provider.

The application must provide the ability for real-time analytics of the inbound health data. The health data must be highly durable. The results of the analytic processing should persist for data mining.

Which architecture outlined below will meet the initial requirements for the collection platform?

- A) Use S3 to collect the inbound sensor data analyze the data with amazon Athena
- B) Use Amazon Kinesis to collect the inbound sensor data, analyze the data with Kinesis clients and save the results to a Redshift cluster using EMR
- C) Send data to SNS to collect the inbound sensor data and save the results to AWS RDS Multi-AZ
- D) Send the data to SQS which then sends to DynamoDB

18. A new reality gameshow is being created. During the show users will vote for their favorite contestant. It is expected that millions of users will be voting. The votes must be collected into a durable, scalable, and highly available location. Which service should you use?

- A) Amazon DynamoDB
- B) Amazon Redshift

- C) Microsoft SQL Server
- D) AWS S3

19. You are tasked with creating a solution to analyze a customer's clickstream data on a website to analyze user behavior. The analysis must provide the sequence of pages that are clicked by websites users. This data will be used in real time to optimize the websites performance in terms of page stickiness and advertising click-through rates. Which is the best option to capture and analyze user behavior in real time?

- A) Send web clicks data to Amazon S3, and then analyze and analyze behavior using Amazon Athena
- B) Push web clicks data to Amazon Kinesis and analyze behavior using Kinesis workers
- C) Write web clicks directly to DynamoDB
- D) Write web clicks directly to Amazon RDS for Oracle

20. An application provides data transformation services. Data to be transformed is uploaded to Amazon S3 and then transformed by a fleet of spot EC2 instances. VIP customers should have their files transformed before other customers. How should you implement a system that services VIP customers first?

- A) This cannot be performed as the apposition process messages in the order they are received
- B) Use an ELB to distribute VIP traffic first and then generic traffic to the spot fleet of transformation instances
- C) Set-up two SQS queues. A priority queue for VIP customers and a second queue with default priority for everyone else. Have the transformation instances first poll the high priority queue; if there is no message, then poll the default priority queue.
- D) Use SNS to send a message to administrators to manually send VIP customers data for immediate transformation

21. An organization is planning on setting up a bastion host to help manage systems on their VPC. The bastion host must be reachable from all internet addresses. The bastion host must also be able to access the internal network and should only be open to SSH traffic from a small CIDR range of addresses. How can the bastion host be configured for this purpose?

- A) This cannot be performed as the host is on a public subnet
- B) Create two network interfaces on two different subnets. Assign security groups to allow external traffic on the public interface and SSH traffic on the internal network interface
- C) Create two network interfaces with the same subnets. Assign security groups to allow external traffic on the public interface and SSH traffic on the internal network interface
- D) Separate the services. Put the web server on an EC2 instance and set up a second server for SSH traffic

22. \_\_\_\_\_ pricing offers a significant discount over on demand pricing. This pricing approach works well for mission critical applications with known capacity utilization and know duration of use.

- A) Discount Voucher
- B) Reserved Instance
- C) AWS coupon code
- D) Spot instance

23. An organization's security policy requires encryption of sensitive data at rest. The data is stored on an EBS volume which is attached to an EC2 instance. Which options would facilitate to encrypting your data at rest? (Choose 3)

- A) Leverage third party volume encryption tools
- B) Move data from EBS to S3
- C) Encrypt data prior to storing on EBS
- D) Encrypt data using native data encryption drivers at the file system level

E) Unnecessary as all data on AWS is encrypted

24. What does the PollForTask action perform when it's called by a task runner in AWS Data Pipeline?

A) It retrieves the pipeline definition

B) It sends an SNS message to AWS administrators

C) It sends the data to the next application in the task

D) It performs the next task to perform from AWS Data Pipeline

25. Which of the following are customer responsibilities under the shared security model? Choose 3.

A) Security groups

B) ACLs

C) Patch management of the serverless operating system

D) IAM credentials

E) Managing the underlying hardware of an EC2 instance

26. An organization has three separate divisions (VPCs) and they are main, autos and auto parts. The main organization needs access to auto and auto parts. How can the main organization access the VPCs of autos and auto parts?

A) Set up VPC peering between main and autos and auto parts

B) Open NACLs to allow for full communication

C) Make sure the security groups allow for the CIDR ranges of all VPCs

D) This is not possible as VPCs cannot communicate with each other



27. A company has 500 TB of business-critical data. The company had a fire at their facility and is in immediate need to move their datacenter to the AWS cloud. The company needs to perform this within 7 business days. The company has a 1 Gbps direct connection to AWS which is running at near full capacity. How can you get the system fully operational within the short timeframe?
- A) Request multiple snowball devices from AWS. Load data on the Snowball device. Have AWS download data to an S3 bucket.
  - B) Order a 10Gbps direct connection and send over that link.
  - C) Upload to S3 over the existing 1Gbps internet connection with multipart uploads.
  - D) Use the AWS import/export service. Load data on the hard drives. Have AWS download data to an S3 bucket.
28. An organization is using ElastiCache in front of Amazon RDS database which has four read replicas deployed. The database CPU is at 65 percent with the ElastiCache and cannot meet current capacity if the ElastiCache fails. The server has very limited write use and is mostly limited by read contention. What is a solution to mitigate the impact of an ElastiCache failure?
- A) Spread memory and capacity over a smaller number of larger cache nodes
  - B) Spread memory and capacity over a larger number of smaller cache nodes
  - C) Implement an SQS queue to assist with write capacity
  - a. Use AWS SNS messenger to alert team of cache failures
29. What indicates that an object is successfully stored when put in S3?
- A) An HTTP 404 code is received
  - B) An HTTP 300 code is received
  - C) Cloud watch logs show put was successful
  - D) An HTTP 200 code is received, along with an MD5 hash
30. What is the maximum number of VPCs per region?

- A) 10
- B) 50
- C) 100
- D) 5

31. S3 bucket policies are written in what language?

- A) JavaScript
- B) C++
- C) JSON
- D) Python

32. A global organization is hosting a website on S3. The company is experiencing large data charges for cross-region sharing from the S3 bucket. What changes can be made to reduce costs.

- A) VPC peering
- B) S3 cross region replication
- C) Move the website off S3 and onto an EC2 instance
- D) CloudHub

33. An organization has been storing their data on instance storage. The server was patched for security vulnerability and when it was rebooted all data stored was gone. Why did this happen?

- A) Malware infection
- B) Not enough information is provided to troubleshoot
- C) Instance storage is deleted upon termination or reboot

D) None of the above

34. An organization is using an AWS RDS database. The database is currently running on EBS general purpose storage. At times read and write latency is too high for the organization's needs. How can this be easily remedied?

- A) Upgrade the EBS volume to provisioned IOPS
- B) Change storage type to EBS throughput optimized
- C) Change storage location to an EFS volume
- D) Change to high speed instance storage

35. Relational databases follow the BASE model (Basically Available, Soft State and Eventually Consistent)

- A) True
- B) False

36. An organization has noticed the CPU on their RDS database is consistently at 85%. When looking at the database there is heavy read activity from frequent SQL queries from the finance department. What can the organization do to improve the performance and scalability of the database? Choose 2

- A) Add a read replica and point the finance department's SQL queries to the read replica
- B) Add an ElastiCache to reduce read contention for frequently accessed information
- C) Add an SQS queue to reduce read contention
- D) Set up Multi AZ for the RDS database

37. An organization has set up a high-availability database architecture using a Multi-AZ environment. If the primary database fails which of the following will cause the database to failover to the backup database? Choose 4

- A) The primary database instance fails
  - B) There is an outage in an availability zone
  - C) The database instance type is changed
  - D) The primary database is under maintenance (i.e., patching an operating system)
  - E) The database is busy with a CPU utilization of 90%
38. An organization has a web server in a private subnet that is connected to the internet with a NAT gateway. External users cannot access the web server. What changes can the organization make to have this server reachable from the internet? Choose 3
- A) Put the webserver on a public subnet
  - B) Use a NAT instance with an internet gateway
  - C) Put an ELB in a public subnet and keep the webserver in a private subnet
  - D) There must be a configuration error as the server can ping addresses on the public internet
39. An organization has noticed that when users connect to S3 their traffic is traversing the public internet. The organization is experiencing low performance and high internet costs. What can the organization do to increase the performance, privacy and the scalability of the solution?
- A) Create an endpoint for S3 and connect to the endpoint
  - B) S3 always uses the internet so increase the internet connection speed and encrypt with a VPN
  - C) Set a routing policy to have the organization connect to S3 via the AWS network
  - D) Set up VPC peering to S3
40. An organization has three VPCs. VPC A, VPC B, and VPC C. VPC A is peered with VPC B and VPC C. VPC A can reach VPC B and C. But VPC B and VPC C cannot communicate with each other. Why can't these VPCs communicate with each other.

- A) A firewall is blocking connectivity
- B) There is a misconfigured ACL policy
- C) A configuration error occurred
- D) VPC peering is not transitive and this is normal

41. An organization has set up a NACL to increase the security of their VPC. The organization wants to allow web traffic, TCP Port 80 into the subnet where the web servers reside. They apply the NACL, but no web requests are making it to the web server. Why could this be happening? The ACL can be seen below:

Rule 110 – Deny all traffic

Rule 110 – Inbound Allow TCP Port 80 Source 192.168.1.1

Rule 120 – Outbound Allow TCP Port 80 Source 192.168.1.1

- A) The ACL is properly configured there must be another problem
- B) The order is incorrect, as all traffic is denied prior to being permitted by rule 110 and 120
- C) It is not possible to have a deny statement in an ACL
- D) The security group on the web server is improperly configured

42. With NACL both an inbound and outbound policy is necessary. Why are security groups only configured for an inbound policy?

- A) Security groups are stateful, so they allow outbound return traffic
- B) Security groups require both an inbound and outbound policy
- C) It's an inconsistency in the AWS Cloud
- D) Security groups don't allow outbound traffic

43. If an organization is looking to maximize performance for specific applications but doesn't need high availability for this application. What would be the best option in terms of placement groups?
- A) Spread placement group
  - B) Partition placement group
  - C) Cluster placement group
  - D) Placement groups don't affect performance
44. Your organization requires encryption of all data at rest. What can you do to encrypt and protect data on the EBS volume that is mounted by an EC2 instance? Which of these options would allow you to encrypt your data at rest? Choose 2
- A) Leverage third party volume encryption tools
  - B) Use SSH to encrypt data
  - C) Encrypt data prior to storing it on EBS
  - D) Use an EFS instance instead of EBS
45. You are designing Internet connectivity for your organizations VPC. The organization has web servers with private addresses that must be reachable from the internet. The web servers must be highly available. What can you use to ensure the web servers are highly available and reachable from the internet? Choose 2
- A) Configure a NAT instance in your VPC. Place web servers behind the NAT instance
  - B) Configure CloudFront and place it in front of your web servers. Put CloudFront on a public subnet
  - C) Assign EIPs to all web servers. Configure a Route53 failover policy attached to the EIPs. Use route 53 with health checks
  - D) Put web servers in multiple availability zones. Create a DNS A record for all EIPs

- E) Put all web servers behind and ELB. Configure a Route53 CNAME record that point to the ELB DNS name
46. An organization is looking to implement an intrusion detection and prevention system into their VPC. This platform should have the ability to scale to meet the needs of a global enterprise organization. How should they design their VPC to achieve scalable IDS/IPS?
- A) Call AWS support and ask them to put a switch port in SPAN mode and attach a packet sniffer to the SPAN
  - B) Set up a proxy server and send all internet requests through the proxy server
  - C) Deploy IDS/IPS onto the organizations firewall
  - D) Put an agent on all servers, that sends network traffic to the IDS/IPS for inspection and management
47. An organization has been using a domain name for their business. The company hired a marketing firm that suggested they change their domain name to something more indicative of their brand. What can the organization do to use a new domain name, while not losing its current customers using the old domain name.
- A) Migrate to the new domain. Set up a DNS CNAME record that redirects current users to the new URL.
  - B) Set up a new website with the domain name and keep the old website operational
  - C) Migrate to the new domain. Set up a DNS A record that redirects current users to the new URL
  - D) Migrate to the new domain. Set up a DNS NS record that redirects current users to the new URL.
48. An organization is looking to use a load balancer to increase performance and availability of their website. The organization has tens of millions of customers and is looking for the highest speed load balancer available on the AWS platform. Which is the best option when speed is of utmost importance.
- A) Elastic Load Balancer – Application

- B) Elastic Load Balancer – Network
  - C) Classic Load Balancer – Application
  - D) Route 53 with policy routing instead of a load balancer
49. An organization has configured an IAM role for a user. The user cannot seem to access any information on the VPC. What could be wrong in this situation?
- A) The IAM role was configured in a manner that blocks access to all resources
  - B) IAM roles are not for users, there are created for systems to access other systems in a VPC
  - C) The IAM role was not properly applied to the user
  - D) There is not enough information provided to answer this question
50. Using the IAM concept of AAA (Authentication, Authorization and Accounting) which component determines whether a user is allowed access to a resource?
- A) Authentication
  - B) Authorization
  - C) Accounting
51. An organization desires to connect their on-premise Microsoft Active Directory services with AWS for easier IAM management. How can the organization federate to the organizations Microsoft AD servers?
- A) This is not possible
  - B) Use the AWS AD migration tool
  - C) Build a connection using SAML 2.0
  - D) Leverage AWS directory services and have them peer with on-premise AD servers



52. An organization wants to give several developers access to all AWS resources except IAM. What access should the developers have to support their jobs?

- A) Administrator
- B) Developer
- C) Power User
- D) Sys Admin

53. An organization is looking for a means to store critical information such as passwords and software licenses. Which is the best and most secure option.

- A) Store passwords on an encrypted database
- B) Store passwords in a hidden folder in the root account of an EC2 instance
- C) Store passwords and licenses in the Systems Manager Parameter Store
- D) Store passwords in a spreadsheet, which is inside an encrypted folder on the CEO's computer

54. An organization is setting up a platform for managing and analyzing extremely large amounts of data. The organization is looking to use a serverless environment. What would be the easiest option to deploy and manage a big data framework for this customer?

- A) Set up an EC2 instance and install Apache Hadoop
- B) Use AWS EMR
- C) Build a custom big data management platform and place on a Fargate container
- D) Use DynamoDB

55. An organization wants to use CloudWatch to monitor memory utilization in an application with large memory demands. The organization would like updates every 1 minute. How should the organization set this up?

- A) No set up is required, CloudWatch performs this service automatically
- B) Leverage CloudWatch detailed monitoring and set a CloudWatch custom metric
- C) Leverage CloudTrail and not CloudWatch for this purpose
- D) Set up a lambda function that will use SNS to notify systems administrators when memory utilization goes over 80 percent

56. Lambda functions can be set up in which of the following programming languages.  
Choose 3

- A) Node.js
- B) Visual basic
- C) Python
- D) C#
- E) C++
- F) Pascal

57. An organization wants to restrict access to information stored on S3. Which of the below options can be used to for access control?

- A) ACLs
- B) IAM policies
- C) Bucket Policies
- D) All of the above

58. Your organization stores a substantial amount of data on S3. The files stored on S3 are very large, most over 1 GB. You are traveling and in a part of the world with slow internet access. You only need some of the data in the file but not the entire file. What options, if any, can you use to work more efficiently?

- A) Find a location with better internet access

- B) Use a range get
  - C) There is nothing you can do so be patient
  - D) Use a python script to download files overnight
59. In order to maintain the integrity of autoscaling, autoscaling requests are signed with a hash based upon the users private key. What hashing algorithm is used to calculate the hash?
- A) SHA-256
  - B) HMAC-SHA1
  - C) X11
  - D) X11Gost
60. AWS uses the term elastic for many of their services. What does the Amazon definition of elastic mean?
- A) Bursting capabilities
  - B) The ability to create instances easily
  - C) The ability to scale resources up and down with minimal challenges
  - D) The speed of deployment compared to a traditional datacenter
61. When restoring a database from a DB snapshot which of the following occurs? Choose 2
- A) A new instance is created
  - B) It restores the original instance
  - C) All information including the operating system, database and all data is restored to the new instance
  - D) All information including the operating system, database and all data is restored to the old instance

62. An organization is creating a machine learning application on the AWS platform. Which is the best type of instance for this application?
- A) C5
  - B) T3
  - C) M5
  - D) G3
63. When connecting to AWS via a direct connection what routing protocol is used to exchange routing information for network layer reachability?
- A) OSPF
  - B) EIGRP
  - C) BGP
  - D) RIP V2
64. An organization is using an application that requires physical access to the underlying hardware of the server. Which is the best type of tenancy on the AWS platform?
- A) Shared tenancy
  - B) Dedicated instance
  - C) Dedicated host
  - D) Placement group
65. An organization that creates video games is launching a new game. They expect this application to have then of millions of global users. What is the best database option to store game state?
- A) Microsoft SQL

- B) MySQL
- C) DynamoDB
- D) Amazon Aurora

66. An organization is moving to the cloud. The organization is looking for a way to be more efficient and is willing to modify its current processes. The organization needs a relational database and a data warehouse. The organization has currently been using a custom developed and problematic ETL tool to exchange information between databases and storage. What can the organization do to replace the current ETL tool?

- A) AWS Glue
- B) DynamoDB
- C) ElastiCache
- D) VPC endpoint

67. An organization is looking to set up a mail server on an EC2 instance. What type of DNS record should be set up in route 53?

- A) A record
- B) CNAME record
- C) MX record
- D) NS record

68. An organization wants to set up a scalable IAM solution for mobile phones. The organization wants to authenticate via Facebook and other identity providers. What solution is recommended in this use case?

- A) Set up IPsec between the VPC and Facebook
- B) Set up VPC peering with Facebook
- C) Use Amazon Cognito

D) Use the AWS Directory service

69. When setting up an IAM Policy which statement are not required? Choose 2

A) Action

B) Resource

C) Condition

D) Shield

70. It's often necessary to fan out messages to multiple systems for distributed workflows. Which AWS services is designed for this purpose?

A) EMR

B) ECS

C) SNS

D) EKS

71. An organization has been recently attacked by a hacker. The organization is looking for a means to find systems that do not comply with the organizations policies (operating system, patch level, security groups etc). What is the simplest method for the organization to find systems that do not meet organizational standards?

A) AWS CloudWatch

B) AWS CloudTrail

C) AWS Config

D) AWS CloudFront

72. An organization uses a substantial number of videos in its digital marketing campaigns. The organization would like to ensure that there is no content that could be potentially offensive to some customers. What would be the simplest means to achieve this on the AWS platform?

- A) Set up an EC2 G3 instance, with a python script using a machine algorithm to identify suspect content
  - B) Use AWS Rekognition to identify suspect content
  - C) Use AWS Mechanical Turk to identify suspect content
  - D) Use the AWS Content Manager to identify suspect content
73. A migration from a traditional data center to the cloud can have a profound effect on the organization's technology expenses. A CFO is asking what type of effect a migration to the cloud would have on their technology costs. How would you describe the financial impact to the CFO?
- A) Cloud computing has lower capital expenses (CAPEX) and higher Operational Expenses (OPEX)
  - B) Cloud computing has lower capital expenses (OPEX) and higher Operational Expenses (CAPEX)
  - C) The total cost of ownership is likely more expensive, but there is increased business agility
  - D) The total cost of ownership is likely less expensive and there is increased business agility
74. Creating a full security posture involves the following? Choose all that apply
- A) NACLs
  - B) IAM
  - C) IDS/IPS
  - D) DDos prevention
  - E) Firewalls
  - F) Linux only
75. An organization is looking for a means to sequence multiple lambda functions. What is the simplest way to do this on the AWS platform?

- A) Lambda@Edge
- B) Python script
- C) Step Functions
- D) Glue



## Answer Key

1. B
2. A
3. A, D
4. B
5. B
6. A, D, E
7. C
8. D
9. C
10. B
11. D
12. A
13. D
14. A
15. D
16. B
17. B
18. A
19. B
20. C
21. B
22. B
23. A, C, E
24. A
25. A, B, D
26. A
27. A
28. B
29. D
30. D
31. C
32. B
33. C
34. A
35. B
36. A, B
37. A, B, C, D
38. A, B, C
39. A
40. D
41. B
42. A

- 43. C
- 44. A, C
- 45. C, E
- 46. D
- 47. A
- 48. B
- 49. B
- 50. B
- 51. C
- 52. C
- 53. C
- 54. B
- 55. B
- 56. A, C, D,
- 57. D
- 58. B
- 59. B
- 60. C
- 61. A, C
- 62. D
- 63. C
- 64. C
- 65. C
- 66. A
- 67. B
- 68. C
- 69. C, D
- 70. C
- 71. C
- 72. B
- 73. A, D
- 74. A, B, D, E
- 75. C

## References:

1. <https://aws.amazon.com/s3/>
2. <https://docs.aws.amazon.com/AmazonS3/latest/dev/using-iam-policies.html>
3. <https://learning.oreilly.com/library/view/aws-certified-solutions/9781119138556/c02.xhtml>
4. <https://docs.aws.amazon.com/kms/latest/developerguide/services-s3.html>
5. <https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>
6. <https://aws.amazon.com/ebs/?ebs-whats-new.sort-by=item.additionalFields.postDateTime&ebs-whats-new.sort-order=desc>
7. <https://aws.amazon.com/ebs/volume-types/>
8. <https://www.enterprisestorageforum.com/storage-management/raid-levels.html>
9. <https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html>
10. <https://aws.amazon.com/workdocs/?amazon-workdocs-whats-new.sort-by=item.additionalFields.postDateTime&amazon-workdocs-whats-new.sort-order=desc>
11. <https://aws.amazon.com/fsx/windows/?nc=sn&loc=2>
12. <https://learning.oreilly.com/library/view/aws-certified-solutions/9781119138556/c03.xhtml>
13. <https://www.oracle.com/database/what-is-a-relational-database/>
14. <https://www.ibm.com/cloud/learn/nosql-databases>
15. <https://www.ibm.com/cloud/learn/nosql-databases>
16. <https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/>

17. <https://aws.amazon.com/rds/>
18. <https://aws.amazon.com/dynamodb/>
19. <https://aws.amazon.com/nosql/>
20. <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-indexes-general.html>
21. <https://www.dummies.com/programming/big-data/hadoop/acid-versus-base-data-stores/>
22. <https://aws.amazon.com/redshift/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
23. [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_WorkingWithAutomatedBackups.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html)
24. <https://aws.amazon.com/sqs/>
25. <https://aws.amazon.com/glue/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
26. <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>
27. <https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html>
28. <https://tools.ietf.org/html/rfc1918>
29. <https://tools.ietf.org/html/rfc4291>
30. <https://blog.apnic.net/2020/01/14/bgp-in-2019-the-bgp-table/>
31. [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_Internet\\_Gateway.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html)
32. <https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html>

33. [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_NAT\\_Instance.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html)
34. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>
35. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>
36. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints.html>
37. <https://docs.aws.amazon.com/vpc/latest/userguide/vpce-interface.html>
38. <https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html>
39. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-peering.html>
40. <https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html>
41. <https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-vpn-cloudhub.html>
42. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>
43. [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html)
44. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>
45. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-cluster>
46. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-partition>
47. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-spread>
48. <https://aws.amazon.com/route53/>
49. <https://www.f5.com/services/resources/glossary/load-balancer>
50. <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.htm>

51. <https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>
52. <https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html>
53. <https://aws.amazon.com/compliance/shared-responsibility-model/>
54. <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege>
55. <https://aws.amazon.com/compliance/programs/>
56. <https://aws.amazon.com/iam/>
57. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>
58. [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_common-scenarios\\_aws\\_accounts.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_aws_accounts.html)
59. <https://aws.amazon.com/blogs/security/how-to-audit-cross-account-roles-using-aws-cloudtrail-and-amazon-cloudwatch-events/>
60. <https://aws.amazon.com/identity/federation/>
61. <https://aws.amazon.com/single-sign-on/>
62. <https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>
63. <https://aws.amazon.com/directoryservice/>
64. [https://docs.aws.amazon.com/IAM/latest/UserGuide/access\\_policies\\_managed-vs-inline.html#customer-managed-policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#customer-managed-policies)
65. <https://awspolicygen.s3.amazonaws.com/policygen.html>
66. <https://aws.amazon.com/organizations/getting-started/best-practices/>

67. <https://aws.amazon.com/waf/>
68. <https://aws.amazon.com/blogs/aws/aws-shield-protect-your-applications-from-ddos-attacks/>
69. <https://aws.amazon.com/servicecatalog/?aws-service-catalog.sort-by=item.additionalFields.createdDate&aws-service-catalog.sort-order=desc>
70. <https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html>
71. <https://docs.aws.amazon.com/systems-manager/latest/userguide/what-is-systems-manager.html>
72. <https://aws.amazon.com/sqs/>
73. <https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
74. <https://aws.amazon.com/swf/>
75. <https://aws.amazon.com/emr/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
76. <https://aws.amazon.com/kinesis/>
77. <https://www.docker.com/resources/what-container>
78. <https://aws.amazon.com/ecs/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&ecs-blogs.sort-by=item.additionalFields.createdDate&ecs-blogs.sort-order=desc>

79. <https://aws.amazon.com/eks/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&eks-blogs.sort-by=item.additionalFields.createdDate&eks-blogs.sort-order=desc>
80. <https://aws.amazon.com/elasticbeanstalk/>
81. [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing\\_metrics\\_with\\_cloud\\_watch.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloud_watch.html)
82. <https://aws.amazon.com/cloudwatch/>
83. <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html>
84. <https://aws.amazon.com/blogs/networking-and-content-delivery/dynamic-whole-site-delivery-with-amazon-cloudfront/>
85. <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Invalidation.html>
86. <https://learning.oreilly.com/library/view/aws-certified-solutions/9781119138556/c11.xhtml>
87. <https://aws.amazon.com/lambda/>
88. <https://aws.amazon.com/step-functions/>
89. <https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc>
90. <https://docs.aws.amazon.com/cloudformation/>
91. <https://docs.aws.amazon.com/acm/latest/userguide/acm-overview.html>
92. <https://aws.amazon.com/certificate-manager/?nc=sn&loc=1>



93. <https://aws.amazon.com/blogs/aws-cost-management/getting-started-with-aws-budgets/>
94. <https://aws.amazon.com/premiumsupport/technology/trusted-advisor/>
95. <https://phoenixnap.com/blog/what-is-high-availability>