AWS Cloud Practitioner Essentials

Contents

Introduction to Amazon Web Services 7
What is a Client-Server Model? 7
Deployment Models for Cloud Computing 7
Cloud-based Deployment 7
On-premises Deployment 8
Hybrid Deployment 8
Benefits of Cloud Computing 8
Trade upfront expense for variable expense 8
Stop spending money to run and maintain data centers 8
Stop guessing capacity 9
Benefit from massive economies of scale 9
Increase speed and agility 9
Go global in minutes 9
Resources 9
Compute in the Cloud 10
Amazon Elastic Compute Cloud (Amazon EC2) 10
How Amazon EC2 works 10
Launch 10
Connect 11
Use 11
Amazon EC2 instance types 11
General purpose instances 11
Compute optimized instances 11
Memory optimized instances 12
Accelerated computing instances 12
Storage optimized instances 12
Amazon EC2 Pricing 12
On-Demand 13
Amazon EC2 Savings Plans 13
Reserved Instances 13
Spot Instances 13
Dedicated Hosts 14
Scaling Amazon EC2 (Part 1) 14
Scalability 14
Amazon EC2 Auto Scaling 14
Scaling Amazon EC2 (Part 2) 15
Elastic Load Balancer 17
Example: Elastic Load Balancing 17
Low-demand period 17
High-demand period 18
Monolithic Applications and Microservices 19
Monolithic Applications 19
Microservices 20

Amazon Simple Notification Service (Amazon SNS) 20
Publishing updates from a single topic 21
Publishing updates from multiple topics 21
Amazon Simple Queue Service (Amazon SQS) 22
Example: Fulfilling an order 22
Example: Orders in a queue 23
Serverless Computing 23
AWS Lambda 24
How AWS Lambda works 24
Containers 25
Examples: 25
One host with multiple containers 25
Tens of hosts with hundreds of containers 26
Amazon Elastic Container Service (Amazon ECS) 27
Resources 27
AWS Global Infrastructure & Reliability 28
Selecting a Region 28
Compliance with data governance and legal requirements 28
Proximity to your customers 28
Available services within a Region 28
Pricing 28
Availability Zones 28
Running Amazon EC2 instances in multiple Availability Zones 29
Amazon EC2 instance in a single Availability Zone 29
Amazon EC2 instances in multiple Availability Zones 30
Availability Zone failure 31
Edge Locations 32
Edge Location 33
Customer 33
Ways to Interact with AWS Services 33
AWS Management Console 33
AWS Command Line Interface 34
Software development kits (SDKs) 35
AWS Elastic Beanstalk and AWS CloudFormation 35
AWS Elastic Beanstalk 35
AWS CloudFormation 35
Resources 36
Networking 36
Connectivity to AWS 36
Amazon Virtual Private Cloud (Amazon VPC) 36
Internet gateway 36
What if you have a VPC that includes only private resources? 37
Virtual private gateway 37
AWS Direct Connect 38
Subnets and Network Access Control Lists 38
Subnets 39
Network Traffic in a VPC 40
Network Access Control Lists (ACLs) 41
Stateless Packet Filtering 42

Security Groups 42
Stateful Packet Filtering 43
Global Networking 44
Domain Name System (DNS) 44
Amazon Route 53 45
Example: How Amazon Route 53 and Amazon CloudFront deliver content 45
Resources 46
Storage and Database 46
Instance Stores and Amazon Elastic Block Store (Amazon EBS) 46
Instance stores 46
An Amazon EC2 instance with an attached instance store is running. 47
The instance is stopped or terminated. 47
All data on the attached instance store is deleted. 48
Amazon Elastic Block Storage (Amazon EBS) 48
Amazon EBS Snapshots 49
Amazon Simple Storage Service (Amazon S3) 50
Object Storage 50
Amazon Simple Storage Service (Amazon S3) 50
Amazon S3 Storage Classes 50
Amazon S3 Standard 51
Amazon S3 Standard-Infrequent Access (S3 Standard-IA) 51
Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) 51
Amazon S3 Intelligent-Tiering 51
Amazon S3 Glacier Instant Retrieval 51
Amazon S3 Glacier Flexible Retrieval 51
Amazon S3 Glacier Deep Archive 52
Amazon S3 Outposts 52
Amazon Elastic File System (Amazon EFS) 52
File Storage 52
Comparing Amazon EBS and Amazon EFS 52
Amazon Relational Database Service (Amazon RDS) 53
Relational databases 53
Amazon Relational Database Service 53
Amazon RDS database engines 53
Amazon Aurora 54
Amazon DynamoDB 54
Nonrelational Databases 54
Amazon DynamoDB 55
Amazon Redshift 55
AWS Database Migration Service (AWS DMS) 55
Other use cases for AWS DMS 55
Additional Database Services 56
Amazon DocumentDB 56
Amazon Neptune 56
Amazon Quantum Ledger Database (Amazon QLDB) 56
Amazon Managed Blockchain 56
Amazon ElastiCache 56
Amazon DynamoDB Accelerator 56
Resources 56

Security 57
Shared Responsibility Model 57
Customers: Security in the Cloud 57
AWS: Security of the Cloud 58
User Permission and Access 58
AWS Identity and Access Management (IAM) 58
AWS Account Root User 58
Best practice: 59
IAM users 59
Best practice: 59
IAM policies 60
Best practice: 60
IAM Groups 61
IAM Roles 62
Best practice: 62
Example: IAM Roles 62
AWS Organizations 64
AWS Organizations 64
Organizational Units 64
Example: AWS Organizations 64
Compliance 66
AWS Artifact 66
AWS Artifact Agreements 66
AWS Artifact Reports 66
Denial-of-Service Attacks 67
Distributed Denial-of-Service Attacks 68
AWS Shield 69
AWS Shield Standard 69
AWS Shield Advanced 69
Additional Security Services 70
AWS Key Management Service (AWS KMS) 70
AWS WAF 70
Amazon Inspector 72
Amazon GuardDuty 73
Resources 73
Monitoring and Analytics 73
Amazon CloudWatch 73
Amazon CloudWatch 73
CloudWatch Alarms 74
CloudWatch Dashboard 74
AWS CloudTrail 74
AWS CloudTrail 74
Example: AWS CloudTrail Event 75
CloudTrail Insights 75
AWS Trusted Advisor 75
AWS Trusted Advisor 75
AWS Trusted Advisor Dashboard 76
Resources 76
Pricing and Support 76

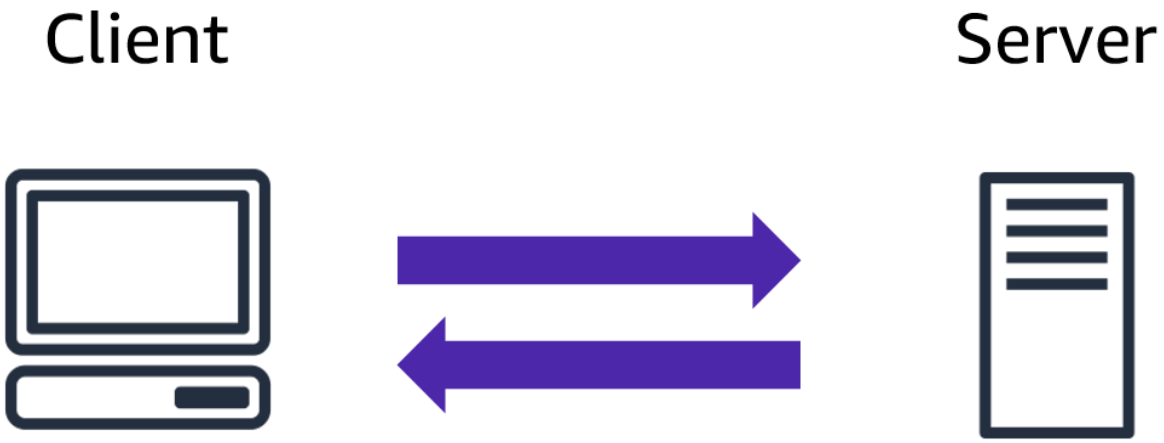
AWS Free Tier 76
AWS Free Tier 76
Always Free 77
12 Months Free 77
Trials 77
AWS Pricing Concepts 77
How AWS Pricing Works 77
Pay for what you use. 77
Pay less when you reserve. 77
Pay less with volume-based discounts when you use more. 78
AWS Pricing Calculator 78
AWS Pricing Examples 78
AWS Lambda Pricing 78
AWS Lambda Pricing Example 79
Amazon EC2 Pricing 79
Amazon EC2 Pricing Example 79
Amazon S3 Pricing 80
Amazon S3 Pricing Example 80
Billing Dashboard 81
Consolidated Billing 81
Example: Consolidated Billing 81
AWS Budgets 83
AWS Budgets 83
Example: AWS Budgets 84
AWS Cost Explorer 84
AWS Cost Explorer 84
Example: AWS Cost Explorer 85
AWS Support Plans 85
AWS Support 85
Basic Support 86
Developer, Business, Enterprise On-Ramp, and Enterprise Support 86
Developer Support 86
Business Support 86
Enterprise On-Ramp Support 87
Enterprise Support 87
Technical Account Manager (TAM) 87
AWS Marketplace 88
AWS Marketplace 88
AWS Marketplace Categories 88
Resources 88
Migration and Innovation 89
AWS Cloud Adoption Framework (AWS CAF) 89
Six core perspectives of the Cloud Adoption Framework 89
Business Perspective 89
People Perspective 89
Governance Perspective 90
Platform Perspective 90
Security Perspective 90
Operations Perspective 91

Migration Strategies ... 91
6 Strategies for Migration ... 91
Rehosting ... 91
Replatforming ... 91
Refactoring/re-architecting ... 91
Repurchasing ... 92
Retaining ... 92
Retiring ... 92
AWS Snow Family ... 92
AWS Snow Family Members ... 92
AWS Snowcone ... 92
AWS Snowball ... 93
AWS Snowmobile ... 93
Innovate with AWS ... 93
Serverless Applications ... 93
Artificial Intelligence ... 93
Machine Learning ... 94
Resources ... 94
The Cloud Journey ... 94
The AWS Well-Architected Framework ... 94
Operational Excellence ... 95
Security ... 95
Reliability ... 95
Performance Efficiency ... 96
Cost Optimization ... 96
Sustainability ... 96
Benefits of the AWS Cloud ... 96
Trade upfront expense for variable expense. ... 96
Benefit from massive economies of scale. ... 97
Stop guessing capacity. ... 97
Increase speed and agility. ... 97
Stop spending money running and maintaining data centers. ... 97
Go global in minutes. ... 97
Resources ... 97
Additional resources ... 98
AWS Certified Cloud Practitioner Basics ... 98
Exam Details ... 98
Exam Domains ... 98
Recommended Experience ... 99
Exam Details ... 99
Whitepapers and Resources ... 99

Introduction to Amazon Web Services

What is a Client-Server Model?

You just learned more about AWS and how almost all of modern computing uses a basic client-server model. Let's recap what a client-server model is.



In computing, a **client** can be a web browser or desktop application that a person interacts with to make requests to computer servers. A **server** can be services such as Amazon Elastic Compute Cloud (Amazon EC2), a type of virtual server.

For example, suppose that a client makes a request for a news article, the score in an online game, or a funny video. The server evaluates the details of this request and fulfills it by returning the information to the client.

Deployment Models for Cloud Computing

When selecting a cloud strategy, a company must consider factors such as required cloud application components, preferred resource management tools, and any legacy IT infrastructure requirements.

The three cloud computing deployment models are cloud-based, on-premises, and hybrid.

Cloud-based Deployment

- Run all parts of the application in the cloud.
- Migrate existing applications to the cloud.
- Design and build new applications in the cloud.

In a **cloud-based deployment** model, you can migrate existing applications to the cloud, or you can design and build new applications in the cloud. You can build those applications on low-level infrastructure that requires your IT staff to manage them. Alternatively, you can build them using

higher-level services that reduce the management, architecting, and scaling requirements of the core infrastructure.

For example, a company might create an application consisting of virtual servers, databases, and networking components that are fully based in the cloud.

On-premises Deployment

- Deploy resources by using virtualization and resource management tools.
- Increase resource utilization by using application management and virtualization technologies.

On-premises deployment is also known as a *private cloud* deployment. In this model, resources are deployed on premises by using virtualization and resource management tools. For example, you might have applications that run on technology that is fully kept in your on-premises data center. Though this model is much like legacy IT infrastructure, its incorporation of application management and virtualization technologies helps to increase resource utilization.

Hybrid Deployment

- Connect cloud-based resources to on-premises infrastructure.
- Integrate cloud-based resources with legacy IT applications.

In a **hybrid deployment**, cloud-based resources are connected to on-premises infrastructure. You might want to use this approach in a number of situations. For example, you have legacy applications that are better maintained on premises, or government regulations require your business to keep certain records on premises.

For example, suppose that a company wants to use cloud services that can automate batch data processing and analytics. However, the company has several legacy applications that are more suitable on premises and will not be migrated to the cloud. With a hybrid deployment, the company would be able to keep the legacy applications on premises while benefiting from the data and analytics services that run in the cloud.

Benefits of Cloud Computing

Consider why a company might choose to take a particular cloud computing approach when addressing business needs.

Trade upfront expense for variable expense

Upfront expense refers to data centers, physical servers, and other resources that you would need to invest in before using them. Variable expense means you only pay for computing resources you consume instead of investing heavily in data centers and servers before you know how you're going to use them.

By taking a cloud computing approach that offers the benefit of variable expense, companies can implement innovative solutions while saving on costs.

Stop spending money to run and maintain data centers

Computing in data centers often requires you to spend more money and time managing infrastructure and servers. A benefit of cloud computing is the ability to focus less on these tasks and more on your applications and customers.

Stop guessing capacity

With cloud computing, you don’t have to predict how much infrastructure capacity you will need before deploying an application. For example, you can launch Amazon EC2 instances when needed, and pay only for the compute time you use. Instead of paying for unused resources or having to deal with limited capacity, you can access only the capacity that you need. You can also scale in or scale out in response to demand.

Benefit from massive economies of scale

By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers can aggregate in the cloud, providers, such as AWS, can achieve higher economies of scale. The economy of scale translates into lower pay-as-you-go prices.

Increase speed and agility

The flexibility of cloud computing makes it easier for you to develop and deploy applications.

This flexibility provides you with more time to experiment and innovate. When computing in data centers, it may take weeks to obtain new resources that you need. By comparison, cloud computing enables you to access new resources within minutes.

Go global in minutes

The global footprint of the AWS Cloud enables you to deploy applications to customers around the world quickly, while providing them with low latency. This means that even if you are located in a different part of the world than your customers, customers are able to access your applications with minimal delays.

Later in this course, you will explore the AWS global infrastructure in greater detail. You will examine some of the services that you can use to deliver content to customers around the world.

Resources

To learn more about the concepts that were explored in Module 1, review these resources.

- AWS glossary
- Whitepaper: Overview of Amazon Web Services
- AWS Fundamentals: Overview
- What is cloud computing?
- Types of cloud computing
- Cloud computing with AWS

Compute in the Cloud

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2) provides secure, resizable compute capacity in the cloud as Amazon EC2 instances.

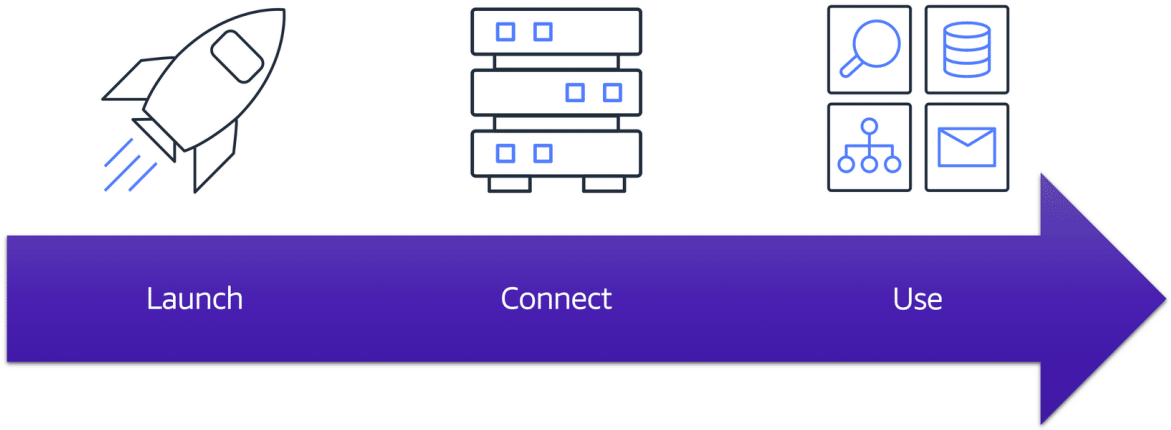
Imagine you are responsible for the architecture of your company's resources and need to support new websites. With traditional on-premises resources, you have to do the following:

- Spend money upfront to purchase hardware.
- Wait for the servers to be delivered to you.
- Install the servers in your physical data center.
- Make all the necessary configurations.

By comparison, with an Amazon EC2 instance you can use a virtual server to run applications in the AWS Cloud.

- You can provision and launch an Amazon EC2 instance within minutes.
- You can stop using it when you have finished running a workload.
- You pay only for the compute time you use when an instance is running, not when it is stopped or terminated.
- You can save costs by paying only for server capacity that you need or want.

How Amazon EC2 works



Launch

First, you launch an instance. Begin by selecting a template with basic configurations for your instance. These configurations include the operating system, application server, or applications. You also select the instance type, which is the specific hardware configuration of your instance.

As you are preparing to launch an instance, you specify security settings to control the network traffic that can flow into and out of your instance. Later in this course, we will explore Amazon EC2 security features in greater detail.

Connect

Next, connect to the instance. You can connect to the instance in several ways. Your programs and applications have multiple different methods to connect directly to the instance and exchange data. Users can also connect to the instance by logging in and accessing the computer desktop.

Use

After you have connected to the instance, you can begin using it. You can run commands to install software, add storage, copy and organize files, and more.

Amazon EC2 instance types

Amazon EC2 instance types are optimized for different tasks. When selecting an instance type, consider the specific needs of your workloads and applications. This might include requirements for compute, memory, or storage capabilities.

General purpose instances

General purpose instances provide a balance of compute, memory, and networking resources. You can use them for a variety of workloads, such as:

- application servers
- gaming servers
- backend servers for enterprise applications
- small and medium databases

Suppose that you have an application in which the resource needs for compute, memory, and networking are roughly equivalent. You might consider running it on a general purpose instance because the application does not require optimization in any single resource area.

Compute optimized instances

Compute optimized instances are ideal for compute-bound applications that benefit from high-performance processors. Like general purpose instances, you can use compute optimized instances for workloads such as web, application, and gaming servers.

However, the difference is compute optimized applications are ideal for high-performance web servers, compute-intensive applications servers, and dedicated gaming servers. You can also use compute optimized instances for batch processing workloads that require processing many transactions in a single group.

Memory optimized instances

Memory optimized instances are designed to deliver fast performance for workloads that process large datasets in memory. In computing, memory is a temporary storage area. It holds all the data and instructions that a central processing unit (CPU) needs to be able to complete actions. Before a computer program or application is able to run, it is loaded from storage into memory. This preloading process gives the CPU direct access to the computer program.

Suppose that you have a workload that requires large amounts of data to be preloaded before running an application. This scenario might be a high-performance database or a workload that involves performing real-time processing of a large amount of unstructured data. In these types of use cases, consider using a memory optimized instance. Memory optimized instances enable you to run workloads with high memory needs and receive great performance.

Accelerated computing instances

Accelerated computing instances use hardware accelerators, or coprocessors, to perform some functions more efficiently than is possible in software running on CPUs. Examples of these functions include floating-point number calculations, graphics processing, and data pattern matching.

In computing, a hardware accelerator is a component that can expedite data processing. Accelerated computing instances are ideal for workloads such as graphics applications, game streaming, and application streaming.

Storage optimized instances

Storage optimized instances are designed for workloads that require high, sequential read and write access to large datasets on local storage. Examples of workloads suitable for storage optimized instances include distributed file systems, data warehousing applications, and high-frequency online transaction processing (OLTP) systems.

In computing, the term input/output operations per second (IOPS) is a metric that measures the performance of a storage device. It indicates how many different input or output operations a device can perform in one second. Storage optimized instances are designed to deliver tens of thousands of low-latency, random IOPS to applications.

You can think of input operations as data put into a system, such as records entered into a database. An output operation is data generated by a server. An example of output might be the analytics performed on the records in a database. If you have an application that has a high IOPS requirement, a storage optimized instance can provide better performance over other instance types not optimized for this kind of use case.

Amazon EC2 Pricing

With Amazon EC2, you pay only for the compute time that you use. Amazon EC2 offers a variety of pricing options for different use cases. For example, if your use case can withstand interruptions, you can save with Spot Instances. You can also save by committing early and locking in a minimum level of use with Reserved Instances.

On-Demand

On-Demand Instances are ideal for short-term, irregular workloads that cannot be interrupted. No upfront costs or minimum contracts apply. The instances run continuously until you stop them, and you pay for only the compute time you use. Sample use cases for On-Demand Instances include developing and testing applications and running applications that have unpredictable usage patterns. On-Demand Instances are not recommended for workloads that last a year or longer because these workloads can experience greater cost savings using Reserved Instances.

Amazon EC2 Savings Plans

AWS offers Savings Plans for several compute services, including Amazon EC2. **Amazon EC2 Savings Plans** enable you to reduce your compute costs by committing to a consistent amount of compute usage for a 1-year or 3-year term. This term commitment results in savings of up to 72% over On-Demand costs.

Any usage up to the commitment is charged at the discounted Savings Plan rate (for example, $10 an hour). Any usage beyond the commitment is charged at regular On-Demand rates.

Later in this course, you will review AWS Cost Explorer, a tool that enables you to visualize, understand, and manage your AWS costs and usage over time. If you are considering your options for Savings Plans, AWS Cost Explorer can analyze your Amazon EC2 usage over the past 7, 30, or 60 days. AWS Cost Explorer also provides customized recommendations for Savings Plans. These recommendations estimate how much you could save on your monthly Amazon EC2 costs, based on previous Amazon EC2 usage and the hourly commitment amount in a 1-year or 3-year Savings Plan.

Reserved Instances

Reserved Instances are a billing discount applied to the use of On-Demand Instances in your account. You can purchase Standard Reserved and Convertible Reserved Instances for a 1-year or 3-year term, and Scheduled Reserved Instances for a 1-year term. You realize greater cost savings with the 3-year option.

At the end of a Reserved Instance term, you can continue using the Amazon EC2 instance without interruption. However, you are charged On-Demand rates until you do one of the following:

- Terminate the instance.
- Purchase a new Reserved Instance that matches the instance attributes (instance type, Region, tenancy, and platform).

Spot Instances

Spot Instances are ideal for workloads with flexible start and end times, or that can withstand interruptions. Spot Instances use unused Amazon EC2 computing capacity and offer you cost savings at up to 90% off of On-Demand prices. Suppose that you have a background processing job that can start and stop as needed (such as the data processing job for a customer survey). You want to start and stop the processing job without affecting the overall operations of your business. If you make a Spot request and Amazon EC2 capacity is available, your Spot Instance launches. However, if you make a Spot request and Amazon EC2 capacity is unavailable, the request is not successful until capacity becomes available. The unavailable capacity might delay the launch of your background processing job. After you have launched a Spot Instance, if

capacity is no longer available or demand for Spot Instances increases, your instance may be interrupted. This might not pose any issues for your background processing job. However, in the earlier example of developing and testing applications, you would most likely want to avoid unexpected interruptions. Therefore, choose a different EC2 instance type that is ideal for those tasks.

Dedicated Hosts

Dedicated Hosts are physical servers with Amazon EC2 instance capacity that is fully dedicated to your use.

You can use your existing per-socket, per-core, or per-VM software licenses to help maintain license compliance. You can purchase On-Demand Dedicated Hosts and Dedicated Hosts Reservations. Of all the Amazon EC2 options that were covered, Dedicated Hosts are the most expensive.

Scaling Amazon EC2 (Part 1)

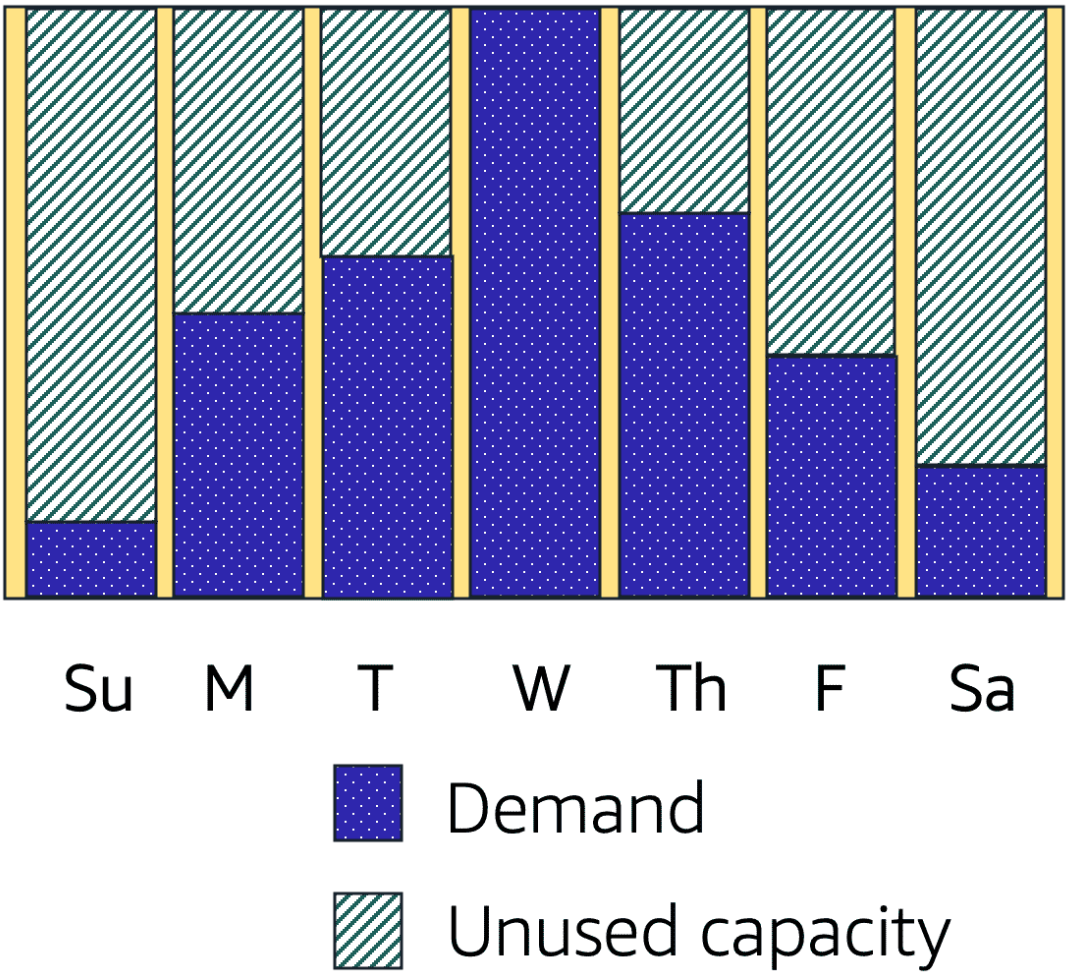
Scalability

Scalability involves beginning with only the resources you need and designing your architecture to automatically respond to changing demand by scaling out or in. As a result, you pay for only the resources you use. You don't have to worry about a lack of computing capacity to meet your customers' needs.

If you wanted the scaling process to happen automatically, which AWS service would you use? The AWS service that provides this functionality for Amazon EC2 instances is **Amazon EC2 Auto Scaling**.

Amazon EC2 Auto Scaling

If you've tried to access a website that wouldn't load and frequently timed out, the website might have received more requests than it was able to handle. This situation is similar to waiting in a long line at a coffee shop, when there is only one barista present to take orders from customers.



Amazon EC2 Auto Scaling enables you to automatically add or remove Amazon EC2 instances in response to changing application demand. By automatically scaling your instances in and out as needed, you are able to maintain a greater sense of application availability.

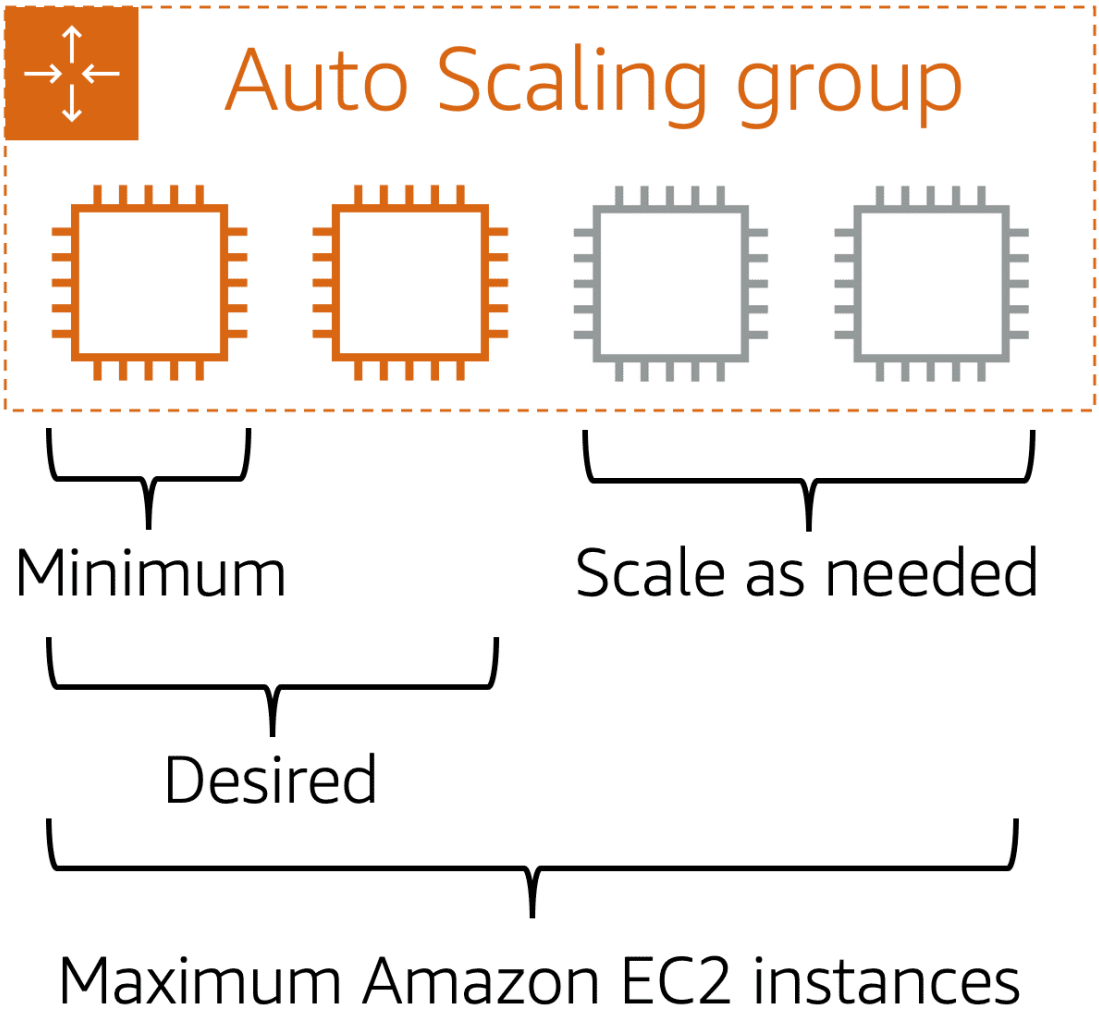
Within Amazon EC2 Auto Scaling, you can use two approaches: dynamic scaling and predictive scaling.

- *Dynamic scaling* responds to changing demand.
- *Predictive scaling* automatically schedules the right number of Amazon EC2 instances based on predicted demand.

Scaling Amazon EC2 (Part 2)

In the cloud, computing power is a programmatic resource, so you can take a more flexible approach to the issue of scaling. By adding Amazon EC2 Auto Scaling to an application, you can add new instances to the application when necessary and terminate them when no longer needed.

Suppose that you are preparing to launch an application on Amazon EC2 instances. When configuring the size of your Auto Scaling group, you might set the minimum number of Amazon EC2 instances at one. This means that at all times, there must be at least one Amazon EC2 instance running.



When you create an Auto Scaling group, you can set the minimum number of Amazon EC2 instances. The **minimum capacity** is the number of Amazon EC2 instances that launch immediately after you have created the Auto Scaling group. In this example, the Auto Scaling group has a minimum capacity of one Amazon EC2 instance.

Next, you can set the **desired capacity** at two Amazon EC2 instances even though your application needs a minimum of a single Amazon EC2 instance to run.

Note: If you do not specify the desired number of Amazon EC2 instances in an Auto Scaling group, the desired capacity defaults to your minimum capacity.

The third configuration that you can set in an Auto Scaling group is the **maximum capacity**. For example, you might configure the Auto Scaling group to scale out in response to increased demand, but only to a maximum of four Amazon EC2 instances.

Because Amazon EC2 Auto Scaling uses Amazon EC2 instances, you pay for only the instances you use, when you use them. You now have a cost-effective architecture that provides the best customer experience while reducing expenses.

Elastic Load Balancer

Elastic Load Balancing is the AWS service that automatically distributes incoming application traffic across multiple resources, such as Amazon EC2 instances.

A load balancer acts as a single point of contact for all incoming web traffic to your Auto Scaling group. This means that as you add or remove Amazon EC2 instances in response to the amount of incoming traffic, these requests route to the load balancer first. Then, the requests spread across multiple resources that will handle them. For example, if you have multiple Amazon EC2 instances, Elastic Load Balancing distributes the workload across the multiple instances so that no single instance has to carry the bulk of it.

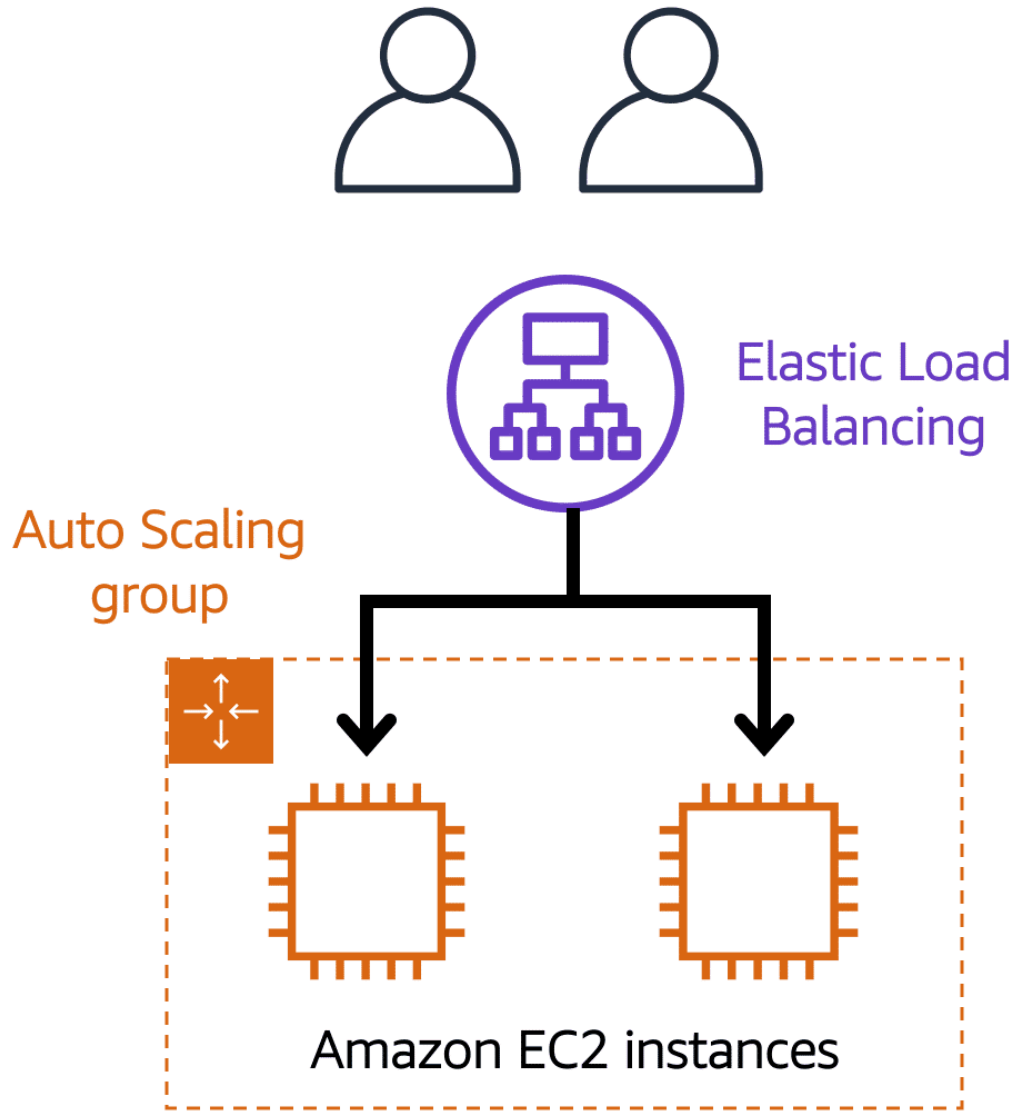
Although Elastic Load Balancing and Amazon EC2 Auto Scaling are separate services, they work together to help ensure that applications running in Amazon EC2 can provide high performance and availability.

Example: Elastic Load Balancing

Low-demand period

Here’s an example of how Elastic Load Balancing works. Suppose that a few customers have come to the coffee shop and are ready to place their orders.

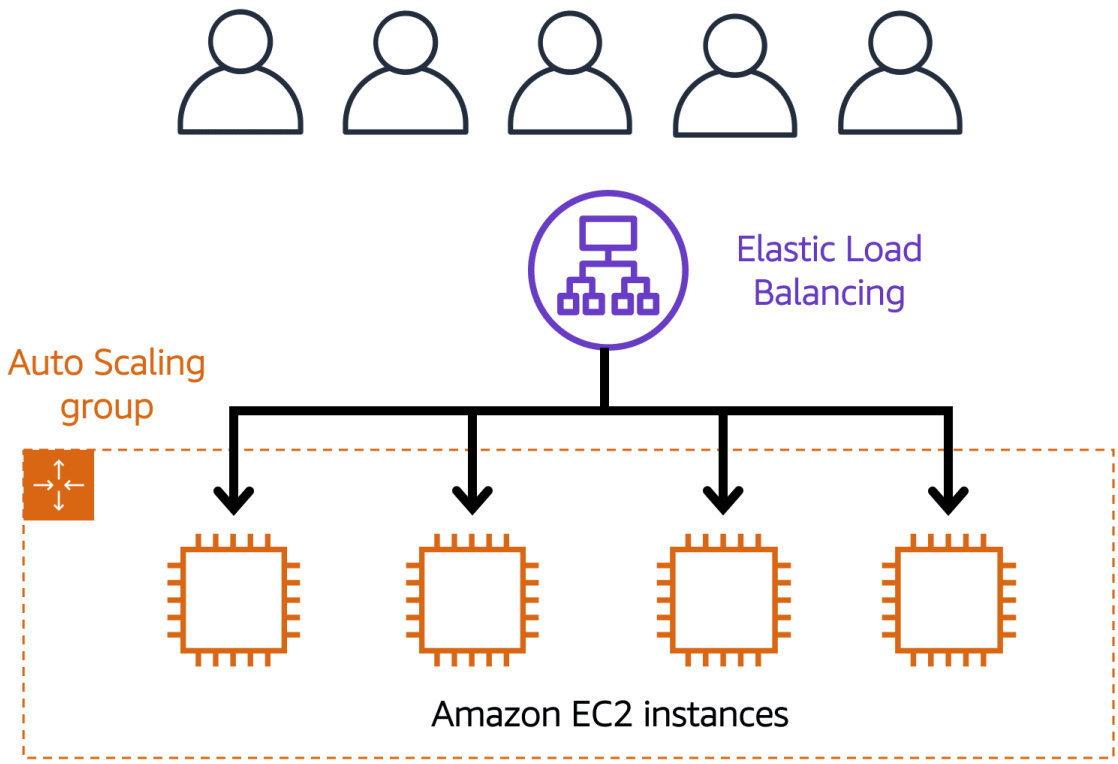
If only a few registers are open, this matches the demand of customers who need service. The coffee shop is less likely to have open registers with no customers. In this example, you can think of the registers as Amazon EC2 instances.



High-demand period

Throughout the day, as the number of customers increases, the coffee shop opens more registers to accommodate them. In the diagram, the Auto Scaling group represents this.

Additionally, a coffee shop employee directs customers to the most appropriate register so that the number of requests can evenly distribute across the open registers. You can think of this coffee shop employee as a load balancer.



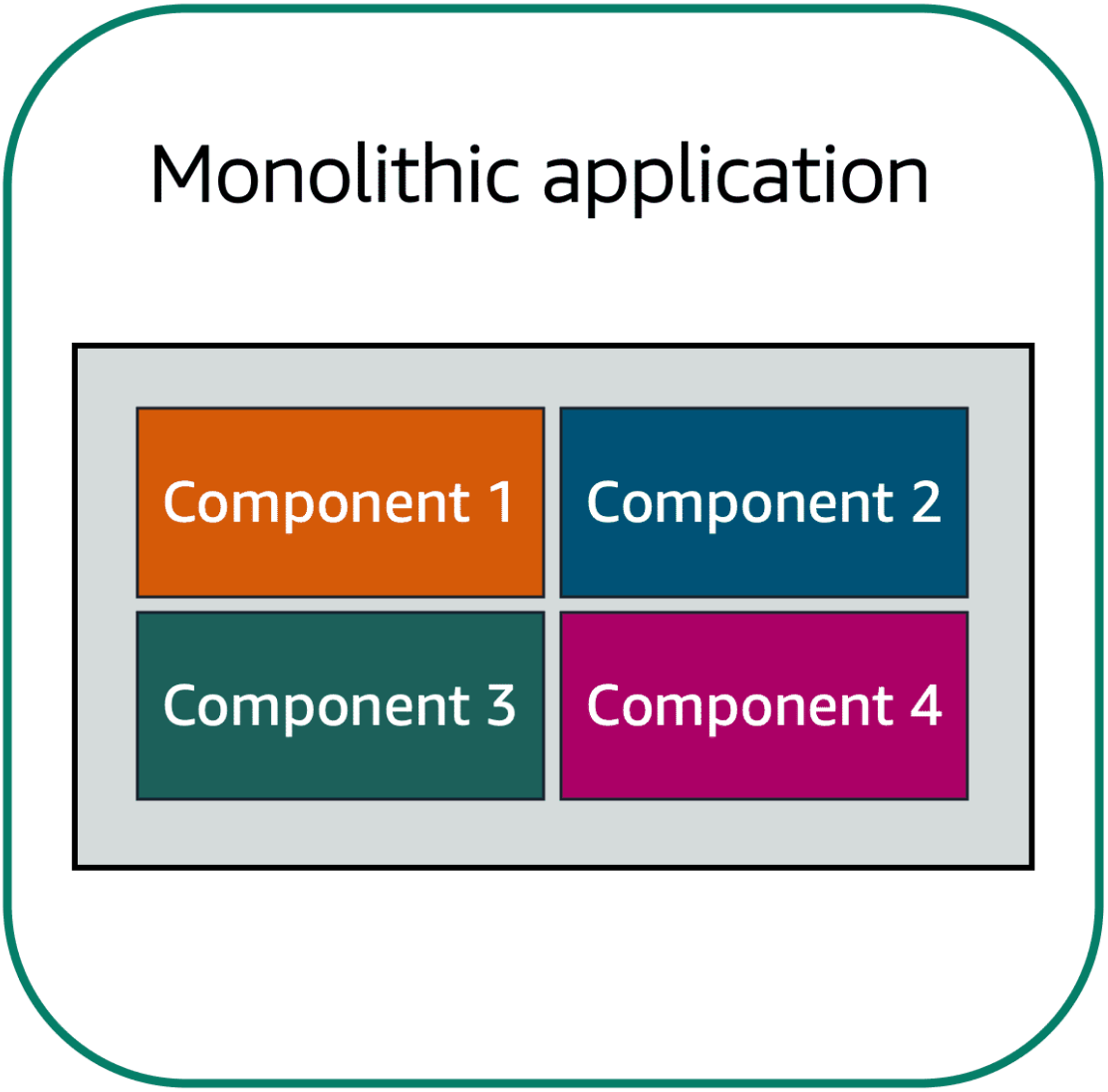
Monolithic Applications and Microservices

Monolithic Applications

Applications are made of multiple components. The components communicate with each other to transmit data, fulfill requests, and keep the application running.

Suppose that you have an application with tightly coupled components. These components might include databases, servers, the user interface, business logic, and so on. This type of architecture can be considered a **monolithic application**.

In this approach to application architecture, if a single component fails, other components fail, and possibly the entire application fails.



Microservices

To help maintain application availability when a single component fails, you can design your application through a **microservices** approach.

In a microservices approach, application components are loosely coupled. In this case, if a single component fails, the other components continue to work because they are communicating with each other. The loose coupling prevents the entire application from failing.

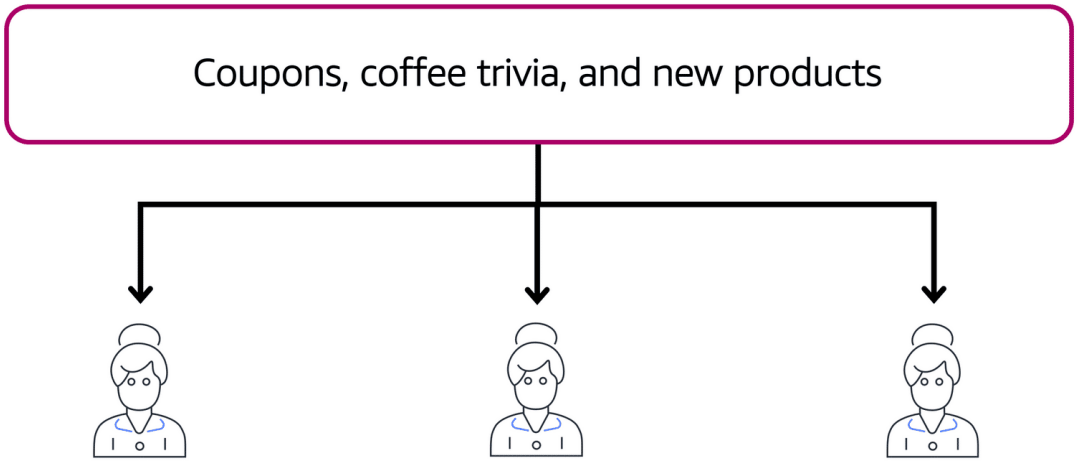
When designing applications on AWS, you can take a microservices approach with services and components that fulfill different functions. Two services facilitate application integration: Amazon Simple Notification Service (Amazon SNS) and Amazon Simple Queue Service (Amazon SQS).

Amazon Simple Notification Service (Amazon SNS)

Amazon Simple Notification Service (Amazon SNS) is a publish/subscribe service. Using Amazon SNS topics, a publisher publishes messages to subscribers. This is similar to the coffee shop; the cashier provides coffee orders to the barista who makes the drinks.

In Amazon SNS, subscribers can be web servers, email addresses, AWS Lambda functions, or several other options.

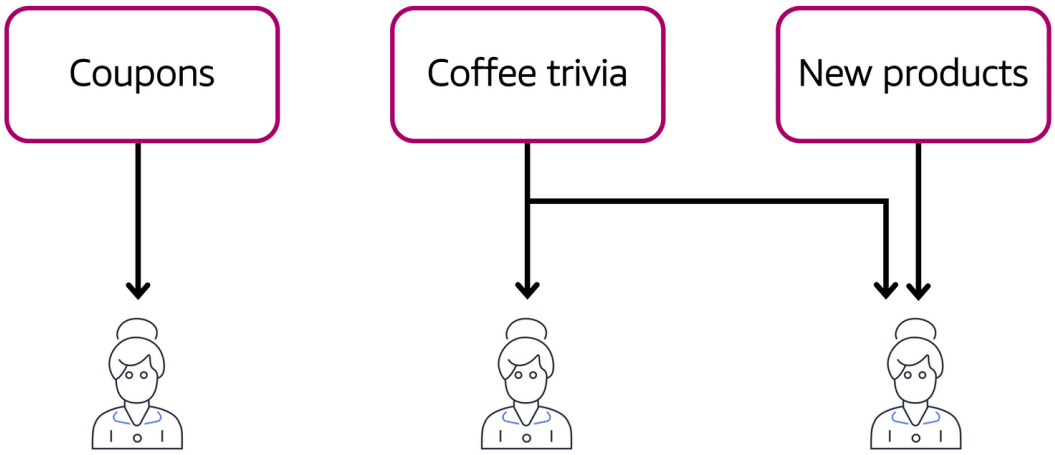
Publishing updates from a single topic



Suppose that the coffee shop has a single newsletter that includes updates from all areas of its business. It includes topics such as coupons, coffee trivia, and new products. All of these topics are grouped because this is a single newsletter. All customers who subscribe to the newsletter receive updates about coupons, coffee trivia, and new products.

After a while, some customers express that they would prefer to receive separate newsletters for only the specific topics that interest them. The coffee shop owners decide to try this approach.

Publishing updates from multiple topics



Now, instead of having a single newsletter for all topics, the coffee shop has broken it up into three separate newsletters. Each newsletter is devoted to a specific topic: coupons, coffee trivia, and new products.

Subscribers will now receive updates immediately for only the specific topics to which they have subscribed.

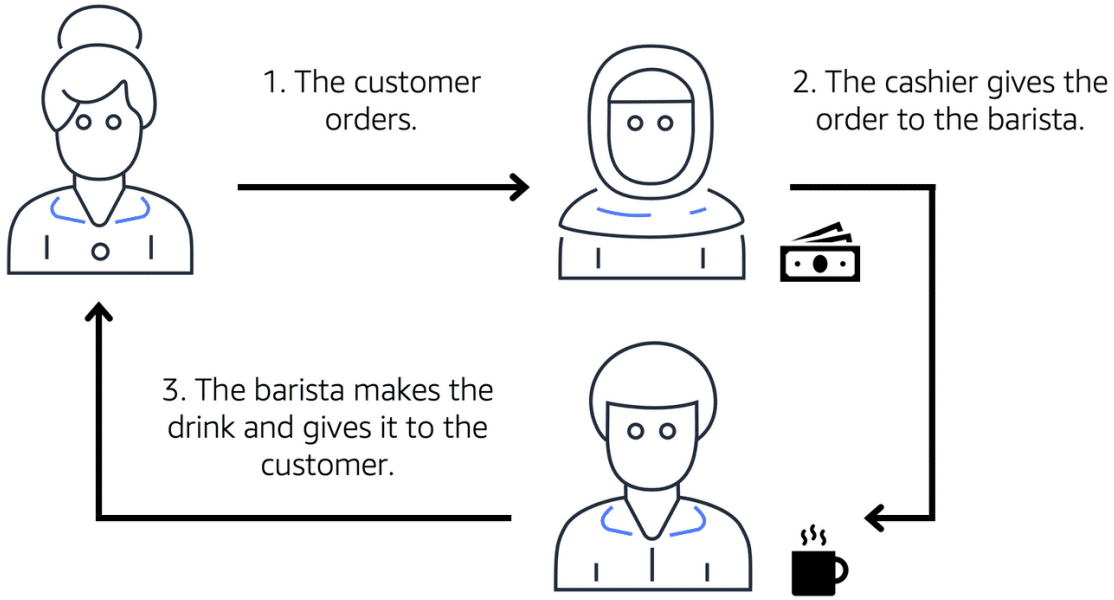
It is possible for subscribers to subscribe to a single topic or to multiple topics. For example, the first customer subscribes to only the coupons topic, and the second subscriber subscribes to only the coffee trivia topic. The third customer subscribes to both the coffee trivia and new products topics.

Amazon Simple Queue Service (Amazon SQS)

Amazon Simple Queue Service (Amazon SQS) is a message queuing service.

Using Amazon SQS, you can send, store, and receive messages between software components, without losing messages or requiring other services to be available. In Amazon SQS, an application sends messages into a queue. A user or service retrieves a message from the queue, processes it, and then deletes it from the queue.

Example: Fulfilling an order



Suppose that the coffee shop has an ordering process in which a cashier takes orders, and a barista makes the orders. Think of the cashier and the barista as two separate components of an application.

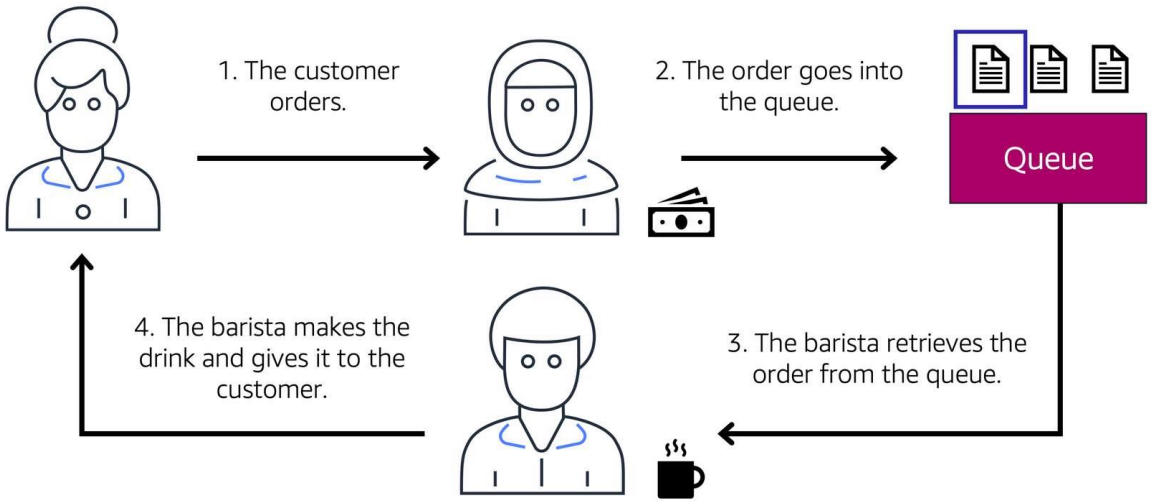
First, the cashier takes an order and writes it down on a piece of paper. Next, the cashier delivers the paper to the barista. Finally, the barista makes the drink and gives it to the customer.

When the next order comes in, the process repeats. This process runs smoothly as long as both the cashier and the barista are coordinated.

What might happen if the cashier took an order and went to deliver it to the barista, but the barista was out on a break or busy with another order? The cashier would need to wait until the barista is ready to accept the order. This would cause delays in the ordering process and require customers to wait longer to receive their orders.

As the coffee shop has become more popular and the ordering line is moving more slowly, the owners notice that the current ordering process is time consuming and inefficient. They decide to try a different approach that uses a queue.

Example: Orders in a queue



Recall that the cashier and the barista are two separate components of an application. A message queuing service such as Amazon SQS enables messages between decoupled application components.

In this example, the first step in the process remains the same as before: a customer places an order with the cashier.

The cashier puts the order into a queue. You can think of this as an order board that serves as a buffer between the cashier and the barista. Even if the barista is out on a break or busy with another order, the cashier can continue placing new orders into the queue.

Next, the barista checks the queue and retrieves the order.

The barista prepares the drink and gives it to the customer.

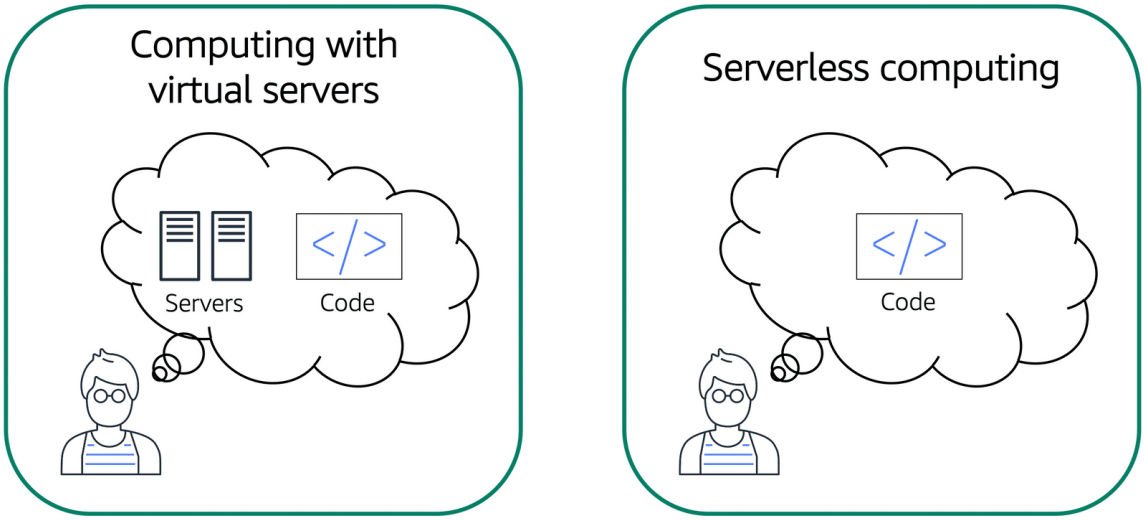
The barista then removes the completed order from the queue.

While the barista is preparing the drink, the cashier is able to continue taking new orders and add them to the queue.

Serverless Computing

Earlier in this module, you learned about Amazon EC2, a service that lets you run virtual servers in the cloud. If you have applications that you want to run in Amazon EC2, you must do the following:

1. Provision instances (virtual servers).
2. Upload your code.
3. Continue to manage the instances while your application is running.



The term “serverless” means that your code runs on servers, but you do not need to provision or manage these servers. With serverless computing, you can focus more on innovating new products and features instead of maintaining servers.

Another benefit of serverless computing is the flexibility to scale serverless applications automatically. Serverless computing can adjust the applications' capacity by modifying the units of consumptions, such as throughput and memory.

An AWS service for serverless computing is **AWS Lambda**.

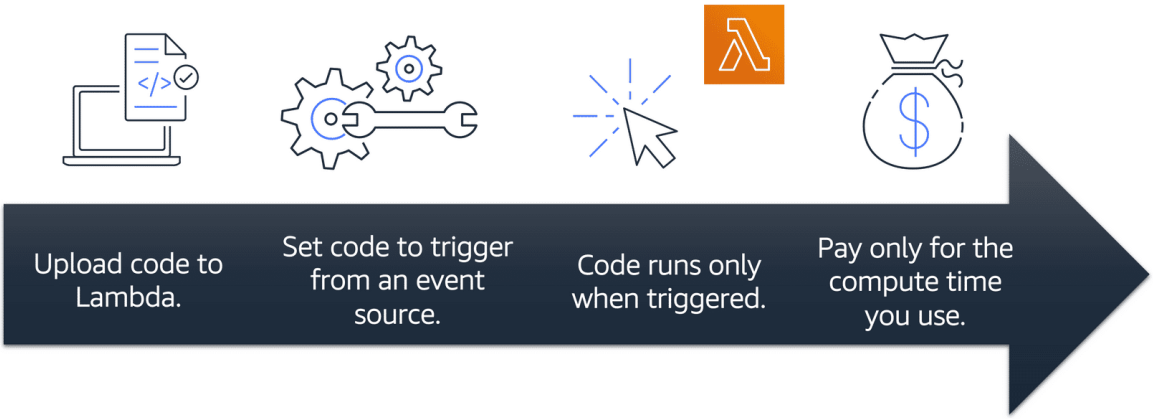
AWS Lambda

AWS Lambda is a service that lets you run code without needing to provision or manage servers.

While using AWS Lambda, you pay only for the compute time that you consume. Charges apply only when your code is running. You can also run code for virtually any type of application or backend service, all with zero administration.

For example, a simple Lambda function might involve automatically resizing uploaded images to the AWS Cloud. In this case, the function triggers when uploading a new image.

How AWS Lambda works



1. You upload your code to Lambda.
2. You set your code to trigger from an event source, such as AWS services, mobile applications, or HTTP endpoints.
3. Lambda runs your code only when triggered.
4. You pay only for the compute time that you use. In the previous example of resizing images, you would pay only for the compute time that you use when uploading new images. Uploading the images triggers Lambda to run code for the image resizing function.

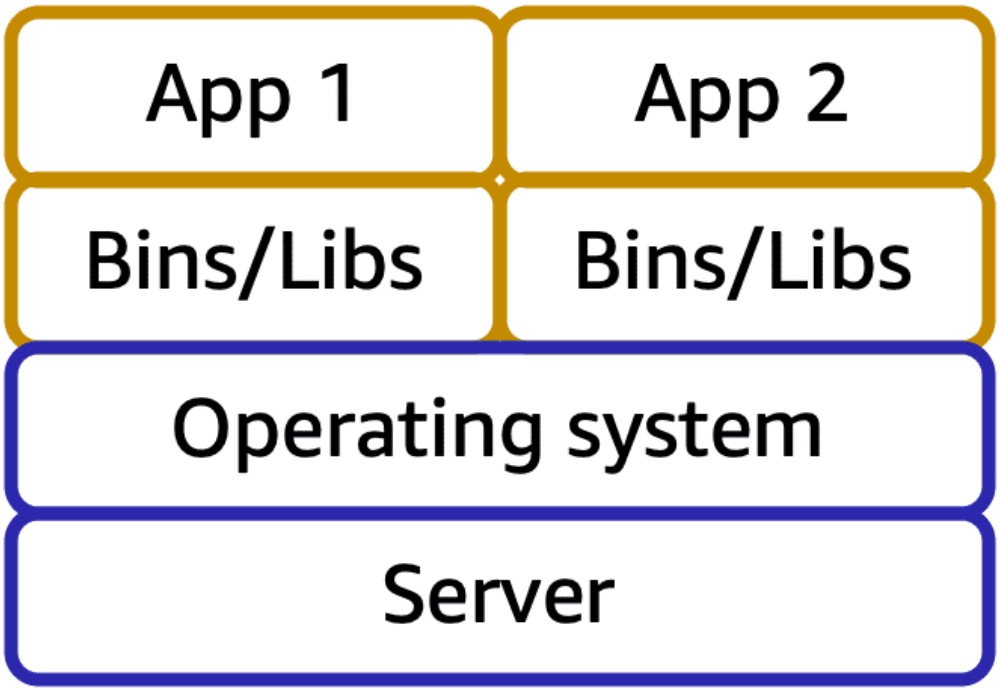
Containers

In AWS, you can also build and run **containerized** applications.

Containers provide you with a standard way to package your application's code and dependencies into a single object. You can also use containers for processes and workflows in which there are essential requirements for security, reliability, and scalability.

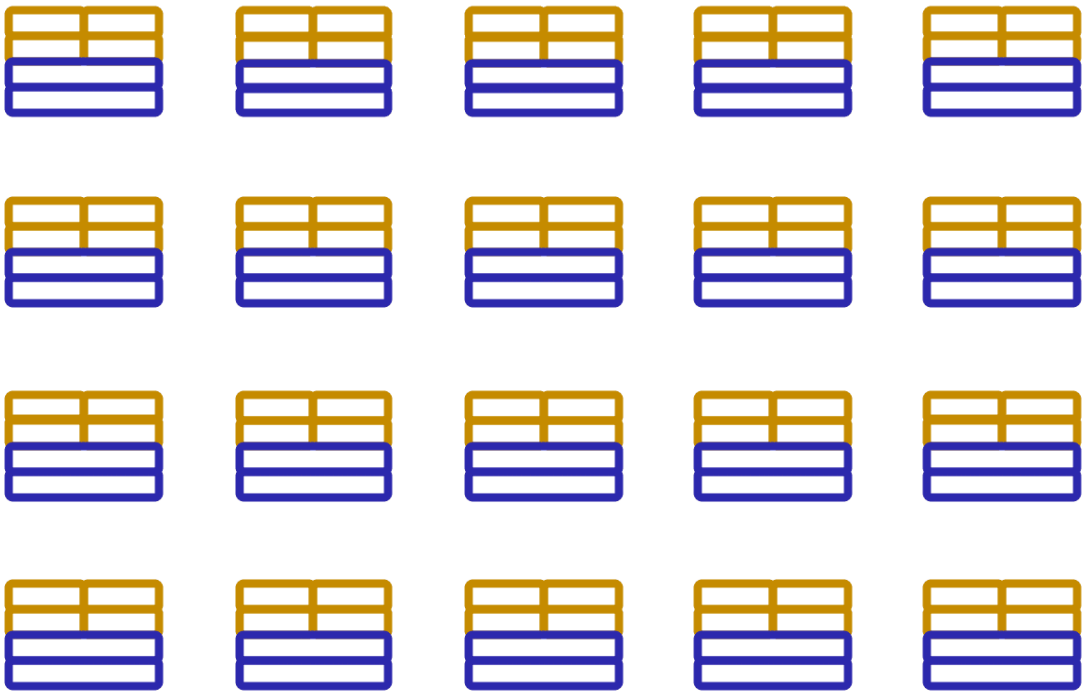
Examples:

One host with multiple containers



Suppose that a company's application developer has an environment on their computer that is different from the environment on the computers used by the IT operations staff. The developer wants to ensure that the application's environment remains consistent regardless of deployment, so they use a containerized approach. This helps to reduce time spent debugging applications and diagnosing differences in computing environments.

Tens of hosts with hundreds of containers



When running containerized applications, it's important to consider scalability. Suppose that instead of a single host with multiple containers, you have to manage tens of hosts with hundreds of containers. Alternatively, you have to manage possibly hundreds of hosts with thousands of containers. At a large scale, imagine how much time it might take for you to monitor memory usage, security, logging, and so on.

Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container management system that enables you to run and scale containerized applications on AWS.

Amazon ECS supports Docker containers. Docker is a software platform that enables you to build, test, and deploy applications quickly. AWS supports the use of open-source Docker Community Edition and subscription-based Docker Enterprise Edition. With Amazon ECS, you can use API calls to launch and stop Docker-enabled applications.

Resources

To learn more about the concepts that were explored in Module 2, review these resources.

- Compute on AWS
- AWS Compute Blog
- AWS Compute Services
- Hands-On Tutorials: Compute
- Category Deep Dive: Serverless
- AWS Customer Stories: Serverless

AWS Global Infrastructure & Reliability

Selecting a Region

When determining the right Region for your services, data, and applications, consider the following four business factors.

Compliance with data governance and legal requirements

Depending on your company and location, you might need to run your data out of specific areas. For example, if your company requires all of its data to reside within the boundaries of the UK, you would choose the London Region.

Not all companies have location-specific data regulations, so you might need to focus more on the other three factors.

Proximity to your customers

Selecting a Region that is close to your customers will help you to get content to them faster. For example, your company is based in Washington, DC, and many of your customers live in Singapore. You might consider running your infrastructure in the Northern Virginia Region to be close to company headquarters, and run your applications from the Singapore Region.

Available services within a Region

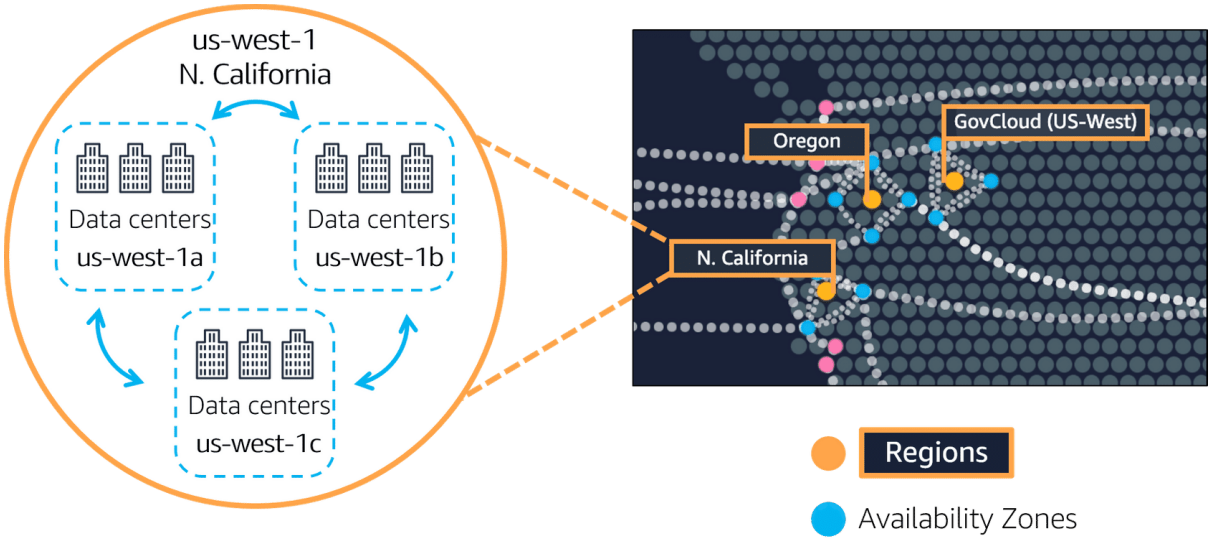
Sometimes, the closest Region might not have all the features that you want to offer to customers. AWS is frequently innovating by creating new services and expanding on features within existing services. However, making new services available around the world sometimes requires AWS to build out physical hardware one Region at a time.

Suppose that your developers want to build an application that uses Amazon Braket (AWS quantum computing platform). As of this course, Amazon Braket is not yet available in every AWS Region around the world, so your developers would have to run it in one of the Regions that already offers it.

Pricing

Suppose that you are considering running applications in both the United States and Brazil. The way Brazil's tax structure is set up, it might cost 50% more to run the same workload out of the São Paulo Region compared to the Oregon Region. You will learn in more detail that several factors determine pricing, but for now know that the cost of services can vary from Region to Region.

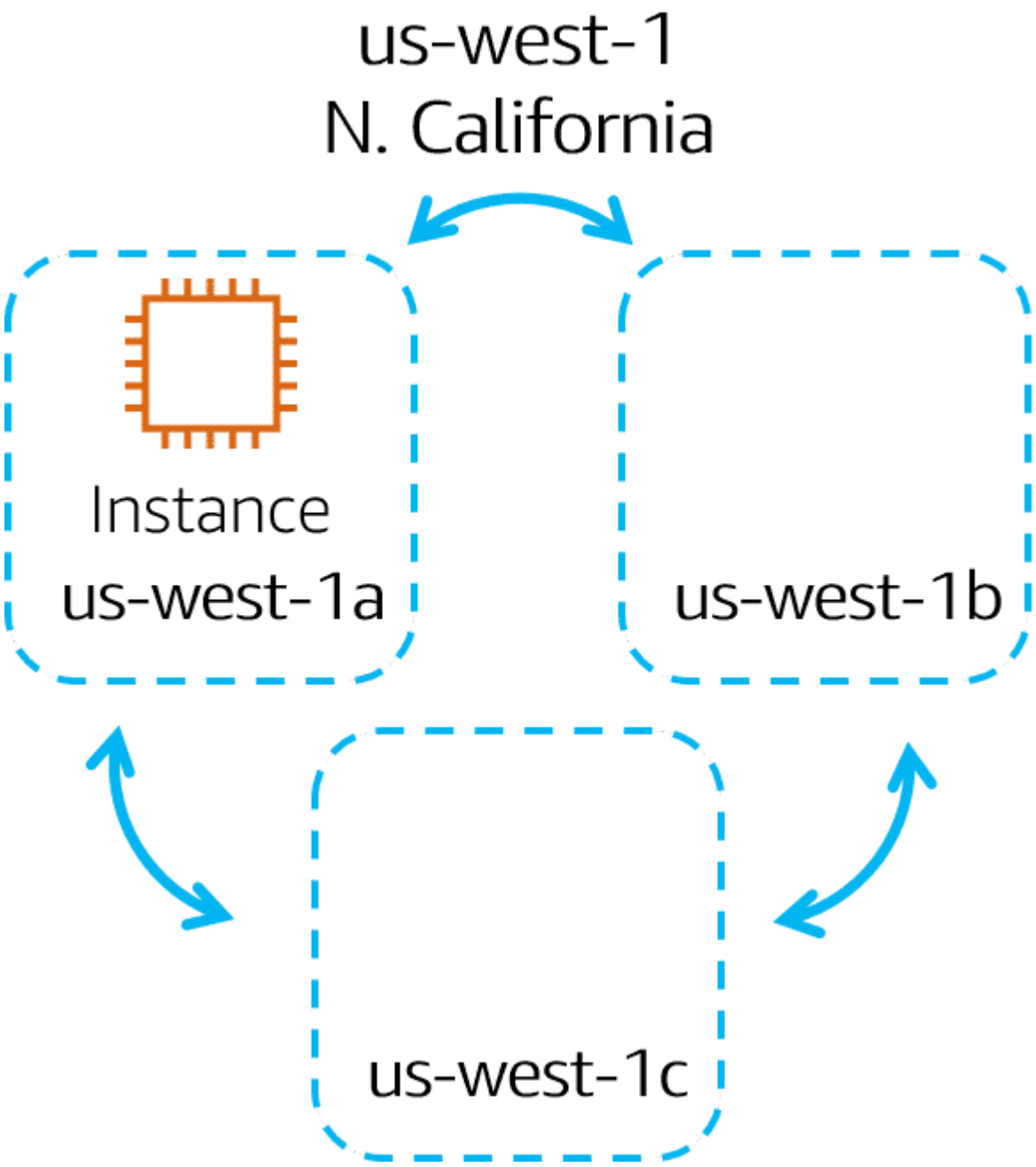
Availability Zones



An **Availability Zone** is a single data center or a group of data centers within a Region. Availability Zones are located tens of miles apart from each other. This is close enough to have low latency (the time between when content requested and received) between Availability Zones. However, if a disaster occurs in one part of the Region, they are distant enough to reduce the chance that multiple Availability Zones are affected.

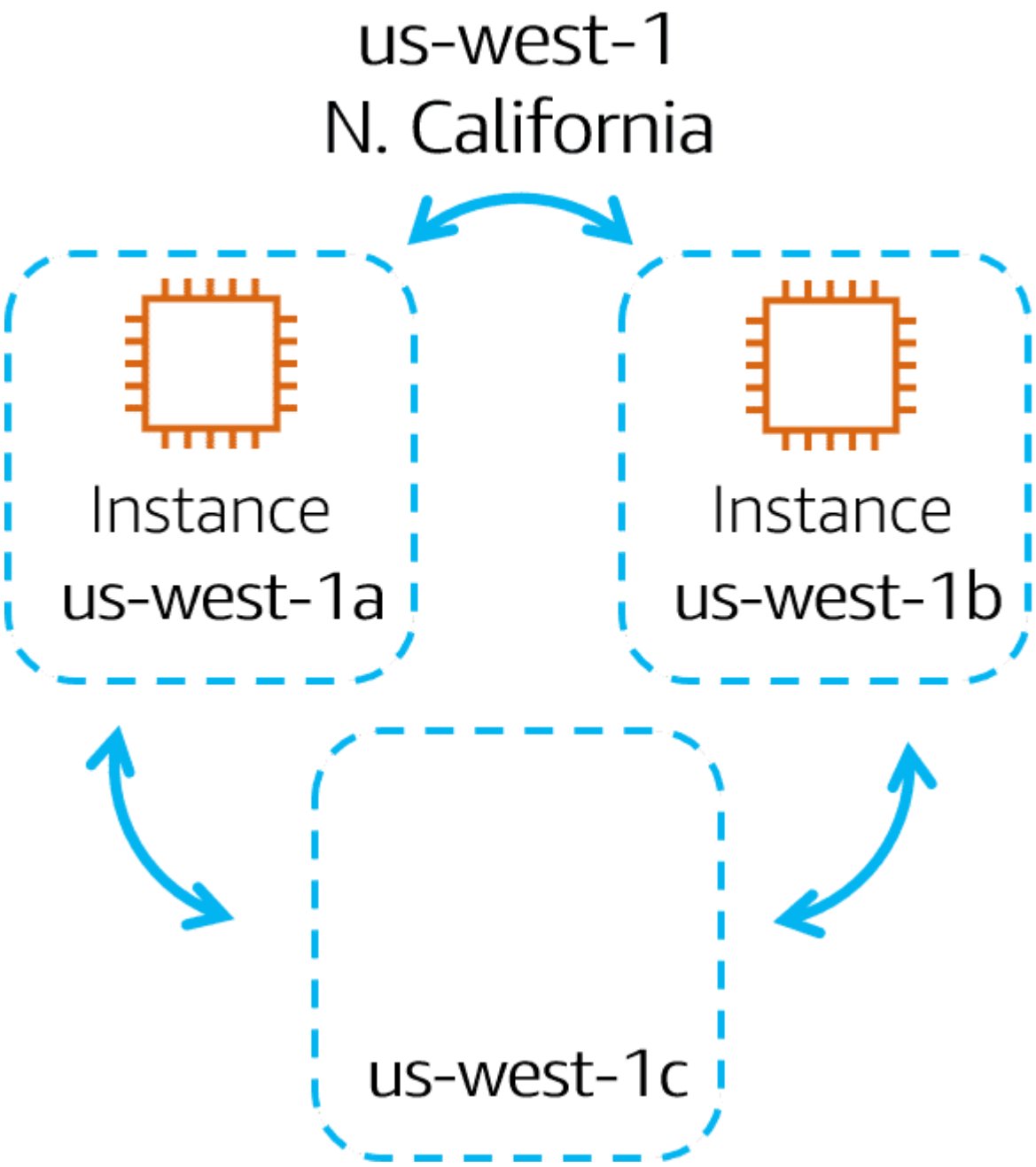
Running Amazon EC2 instances in multiple Availability Zones

Amazon EC2 instance in a single Availability Zone



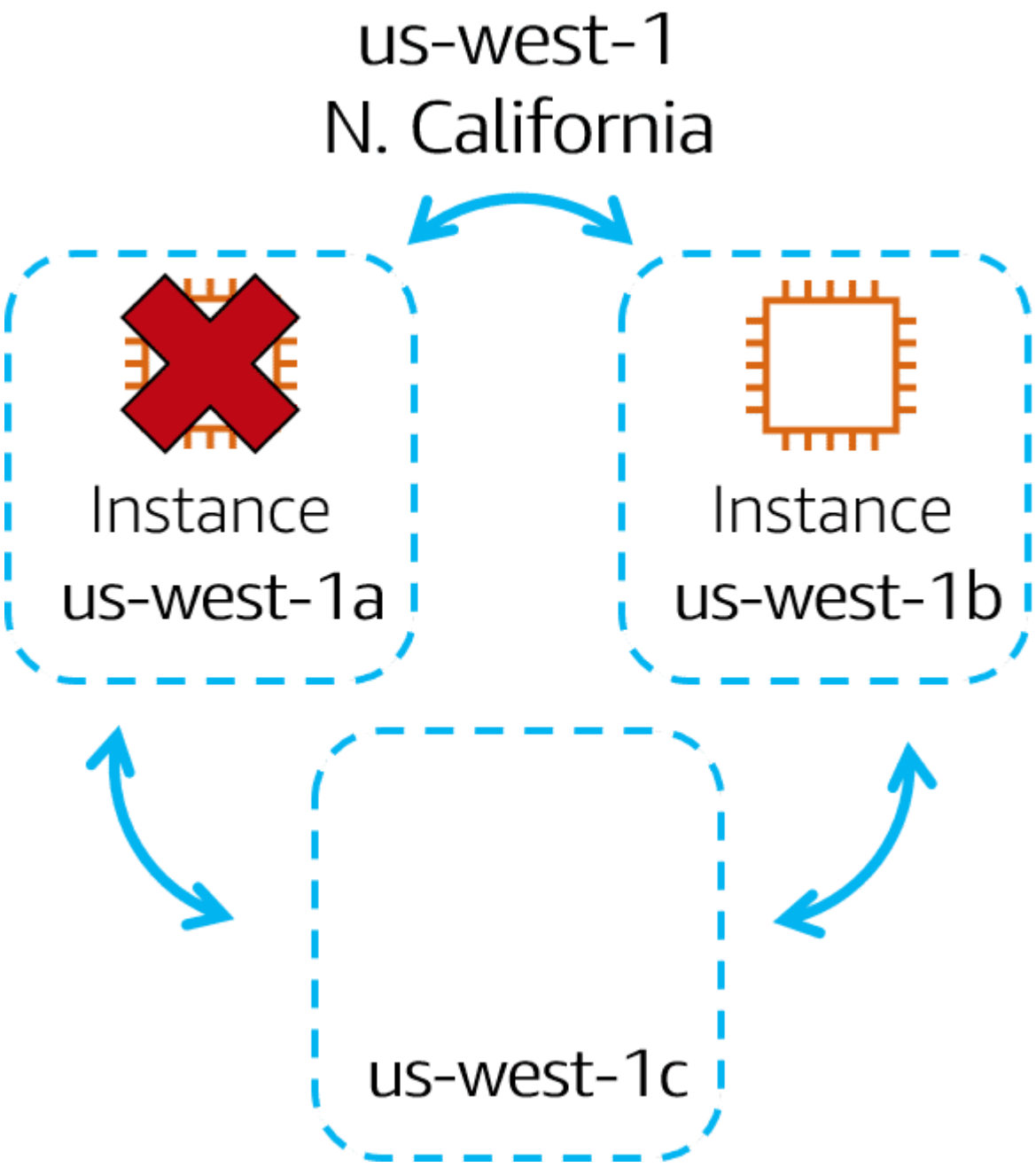
Suppose that you're running an application on a single Amazon EC2 instance in the Northern California Region. The instance is running in the us-west-1a Availability Zone. If us-west-1a were to fail, you would lose your instance.

Amazon EC2 instances in multiple Availability Zones



A best practice is to run applications across at least two Availability Zones in a Region. In this example, you might choose to run a second Amazon EC2 instance in us-west-1b.

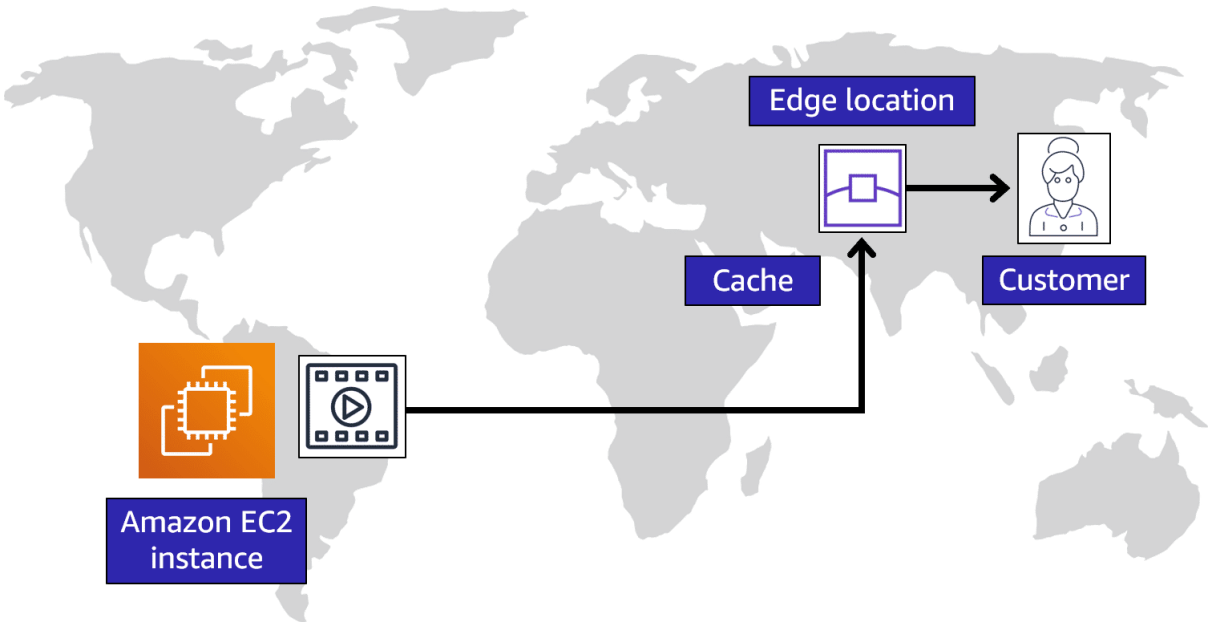
Availability Zone failure



If us-west-1a were to fail, your application would still be running in us-west-1b.

Edge Locations

An **edge location** is a site that Amazon CloudFront uses to store cached copies of your content closer to your customers for faster delivery.



Origin Suppose that your company's data is stored in Brazil, and you have customers who live in China. To provide content to these customers, you don't need to move all the content to one of the Chinese Regions.

Edge Location

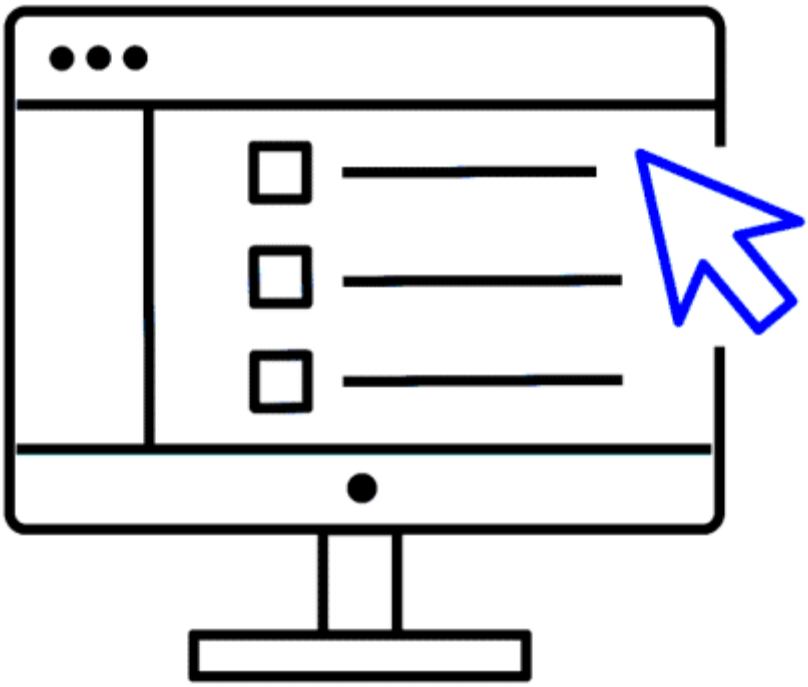
Instead of requiring your customers to get their data from Brazil, you can cache a copy locally at an edge location that is close to your customers in China.

Customer

When a customer in China requests one of your files, Amazon CloudFront retrieves the file from the cache in the edge location and delivers the file to the customer. The file is delivered to the customer faster because it came from the edge location near China instead of the original source in Brazil.

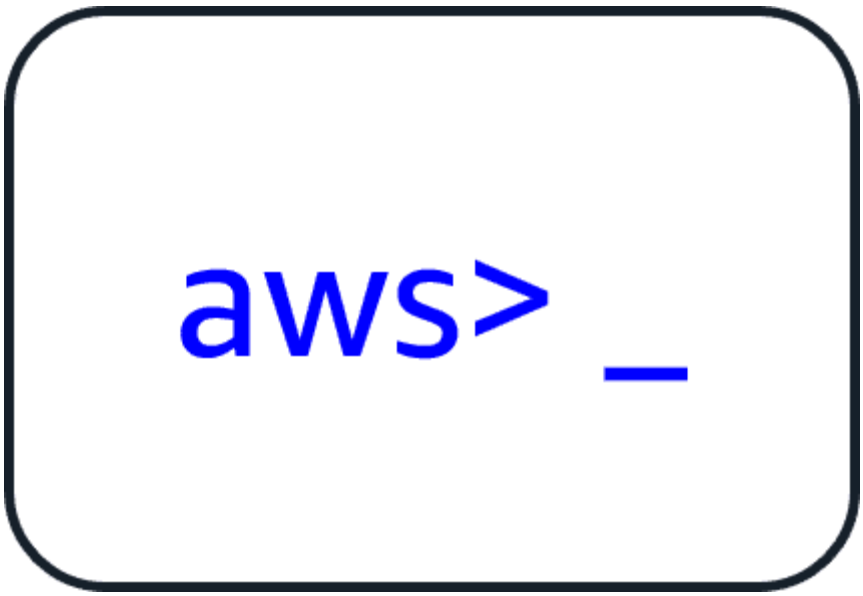
Ways to Interact with AWS Services

AWS Management Console



The **AWS Management Console** is a web-based interface for accessing and managing AWS services. You can quickly access recently used services and search for other services by name, keyword, or acronym. The console includes wizards and automated workflows that can simplify the process of completing tasks. You can also use the AWS Console mobile application to perform tasks such as monitoring resources, viewing alarms, and accessing billing information. Multiple identities can stay logged into the AWS Console mobile app at the same time.

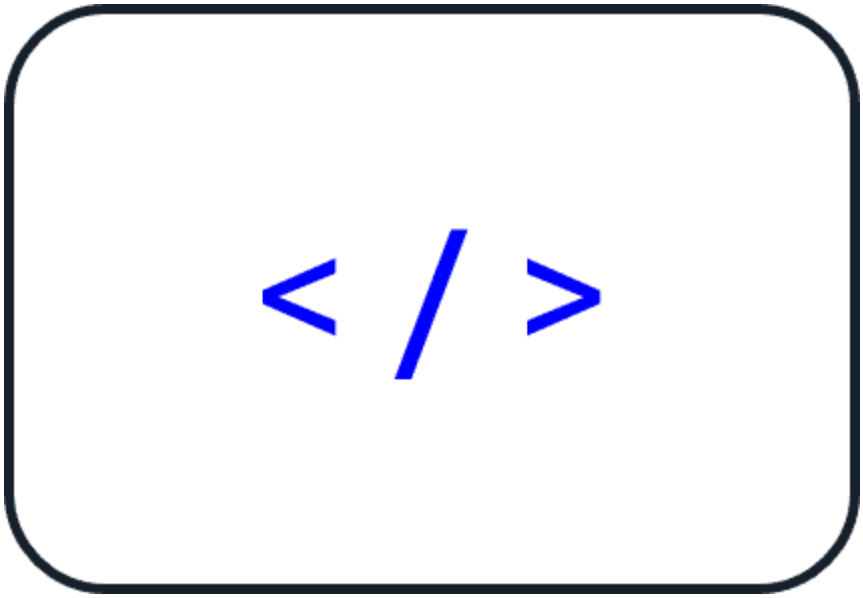
AWS Command Line Interface



To save time when making API requests, you can use the **AWS Command Line Interface (AWS CLI)**. AWS CLI enables you to control multiple AWS services directly from the command line within one tool. AWS CLI is available for users on Windows, macOS, and Linux.

By using AWS CLI, you can automate the actions that your services and applications perform through scripts. For example, you can use commands to launch an Amazon EC2 instance, connect an Amazon EC2 instance to a specific Auto Scaling group, and more.

Software development kits (SDKs)



Another option for accessing and managing AWS services is the **software development kits (SDKs)**. SDKs make it easier for you to use AWS services through an API designed for your programming language or platform. SDKs enable you to use AWS services with your existing applications or create entirely new applications that will run on AWS.

To help you get started with using SDKs, AWS provides documentation and sample code for each supported programming language. Supported programming languages include C++, Java, .NET, and more.

AWS Elastic Beanstalk and AWS CloudFormation

AWS Elastic Beanstalk

With **AWS Elastic Beanstalk**, you provide code and configuration settings, and Elastic Beanstalk deploys the resources necessary to perform the following tasks:

- Adjust capacity
- Load balancing
- Automatic scaling
- Application health monitoring

AWS CloudFormation

With **AWS CloudFormation**, you can treat your infrastructure as code. This means that you can build an environment by writing lines of code instead of using the AWS Management Console to individually provision resources.

AWS CloudFormation provisions your resources in a safe, repeatable manner, enabling you to frequently build your infrastructure and applications without having to perform manual actions or write custom scripts. It determines the right operations to perform when managing your stack and rolls back changes automatically if it detects errors.

Resources

Review these resources to learn more about the concepts that were explored in Module 3.

- Global Infrastructure
- Interactive map of the AWS Global Infrastructure
- Regions and Availability Zones
- AWS Networking and Content Delivery Blog
- Tools to Build on AWS
- AWS Customer Stories: Content Delivery

Networking

Connectivity to AWS

Amazon Virtual Private Cloud (Amazon VPC)

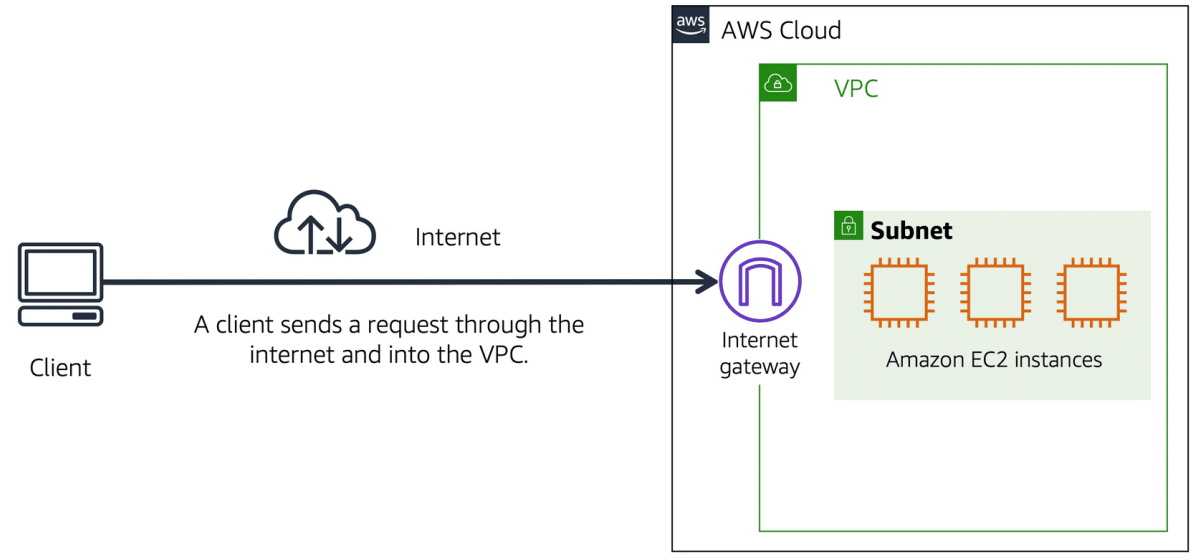
Imagine the millions of customers who use AWS services. Also, imagine the millions of resources that these customers have created, such as Amazon EC2 instances. Without boundaries around all of these resources, network traffic would be able to flow between them unrestricted.

A networking service that you can use to establish boundaries around your AWS resources is **Amazon Virtual Private Cloud (Amazon VPC)**.

Amazon VPC enables you to provision an isolated section of the AWS Cloud. In this isolated section, you can launch resources in a virtual network that you define. Within a virtual private cloud (VPC), you can organize your resources into subnets. A **subnet** is a section of a VPC that can contain resources such as Amazon EC2 instances.

Internet gateway

To allow public traffic from the internet to access your VPC, you attach an **internet gateway** to the VPC.



An internet gateway is a connection between a VPC and the internet. You can think of an internet gateway as being similar to a doorway that customers use to enter the coffee shop. Without an internet gateway, no one can access the resources within your VPC.

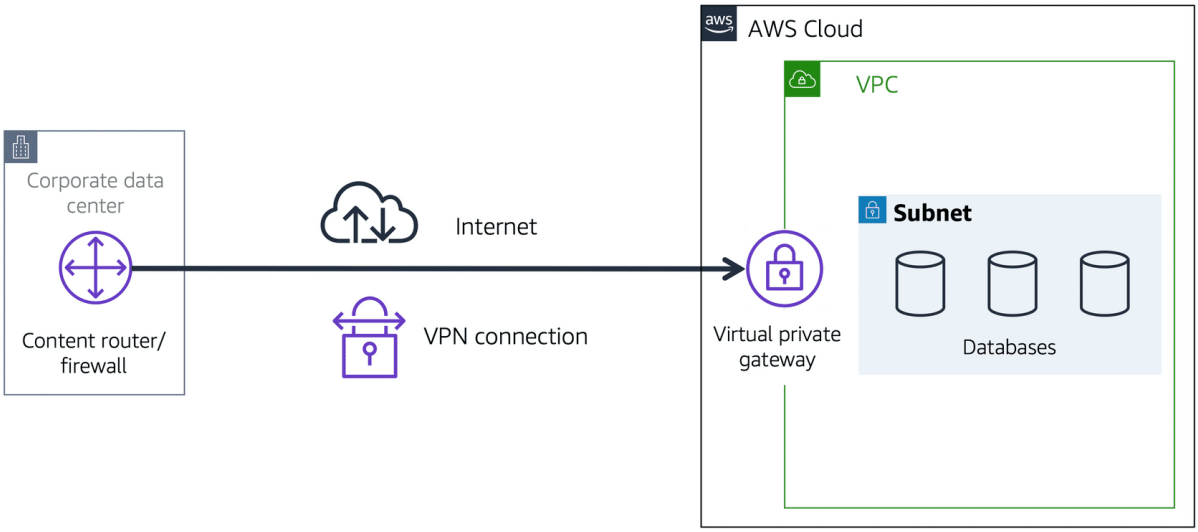
What if you have a VPC that includes only private resources?

Virtual private gateway

Here's an example of how a virtual private gateway works. You can think of the internet as the road between your home and the coffee shop. Suppose that you are traveling on this road with a bodyguard to protect you. You are still using the same road as other customers, but with an extra layer of protection.

The bodyguard is like a virtual private network (VPN) connection that encrypts (or protects) your internet traffic from all the other requests around it.

The virtual private gateway is the component that allows protected internet traffic to enter into the VPC. Even though your connection to the coffee shop has extra protection, traffic jams are possible because you're using the same road as other customers.



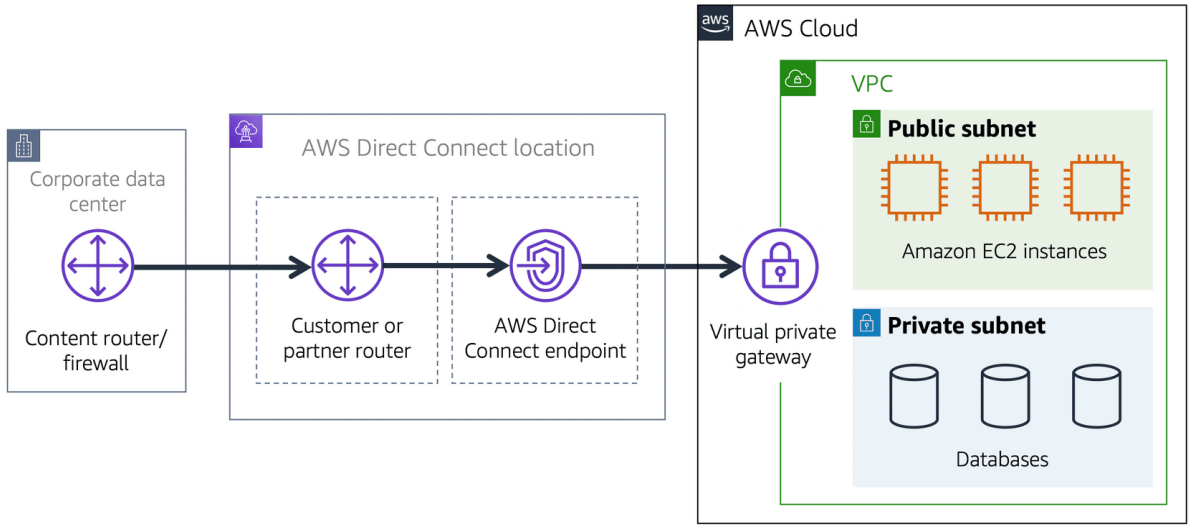
A virtual private gateway enables you to establish a virtual private network (VPN) connection between your VPC and a private network, such as an on-premises data center or internal corporate network. A virtual private gateway allows traffic into the VPC only if it is coming from an approved network.

AWS Direct Connect

AWS Direct Connect is a service that enables you to establish a dedicated private connection between your data center and a VPC.

Suppose that there is an apartment building with a hallway directly linking the building to the coffee shop. Only the residents of the apartment building can travel through this hallway.

This private hallway provides the same type of dedicated connection as AWS Direct Connect. Residents are able to get into the coffee shop without needing to use the public road shared with other customers.



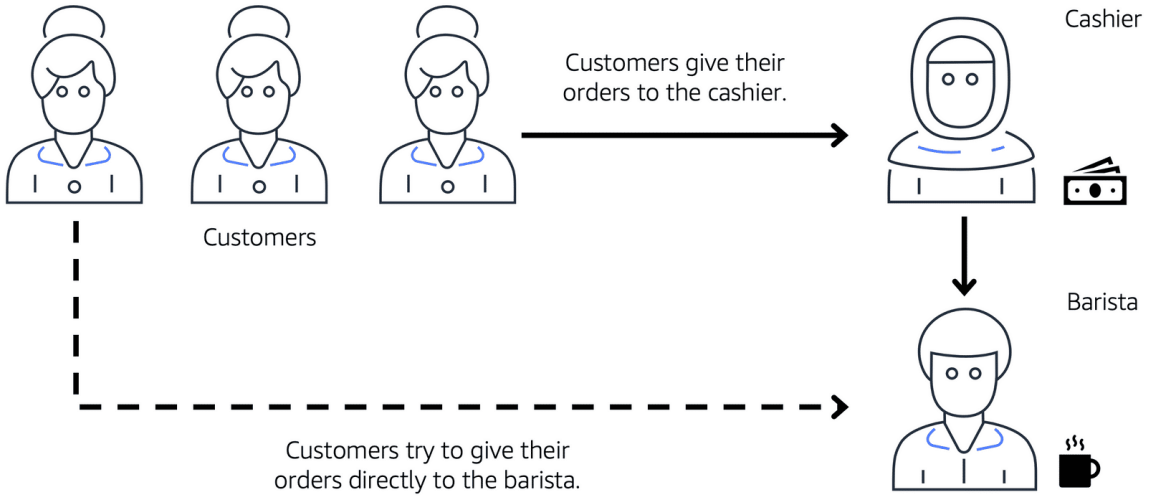
The private connection that AWS Direct Connect provides helps you to reduce network costs and increase the amount of bandwidth that can travel through your network.

Subnets and Network Access Control Lists

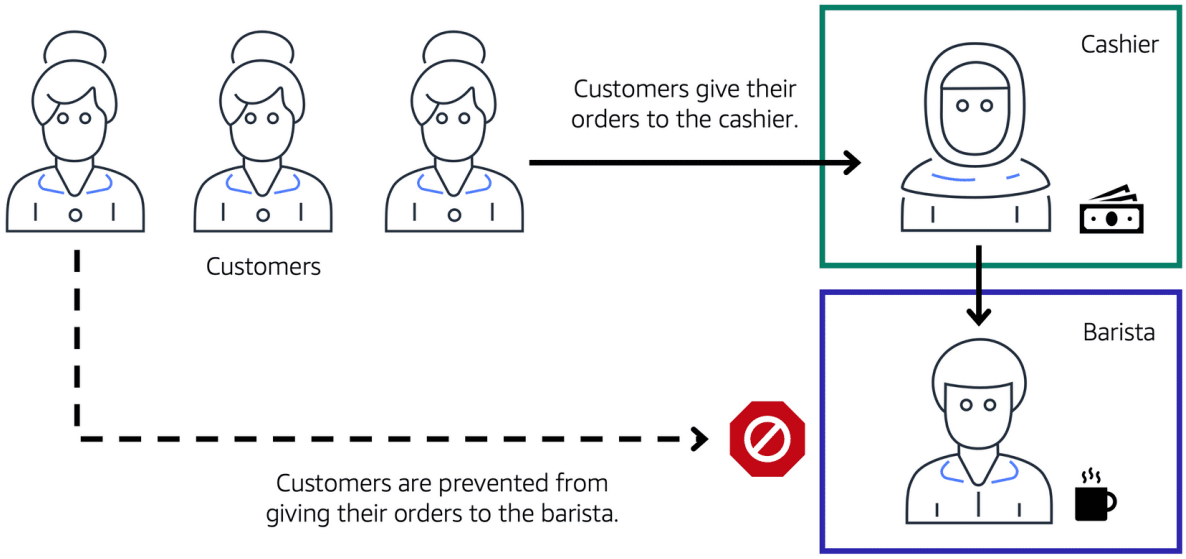
To learn more about the role of subnets within a VPC, review the following example from the coffee shop.

First, customers give their orders to the cashier. The cashier then passes the orders to the barista. This process allows the line to keep running smoothly as more customers come in.

Suppose that some customers try to skip the cashier line and give their orders directly to the barista. This disrupts the flow of traffic and results in customers accessing a part of the coffee shop that is restricted to them.



To fix this, the owners of the coffee shop divide the counter area by placing the cashier and the barista in separate workstations. The cashier's workstation is public facing and designed to receive customers. The barista's area is private. The barista can still receive orders from the cashier but not directly from customers.

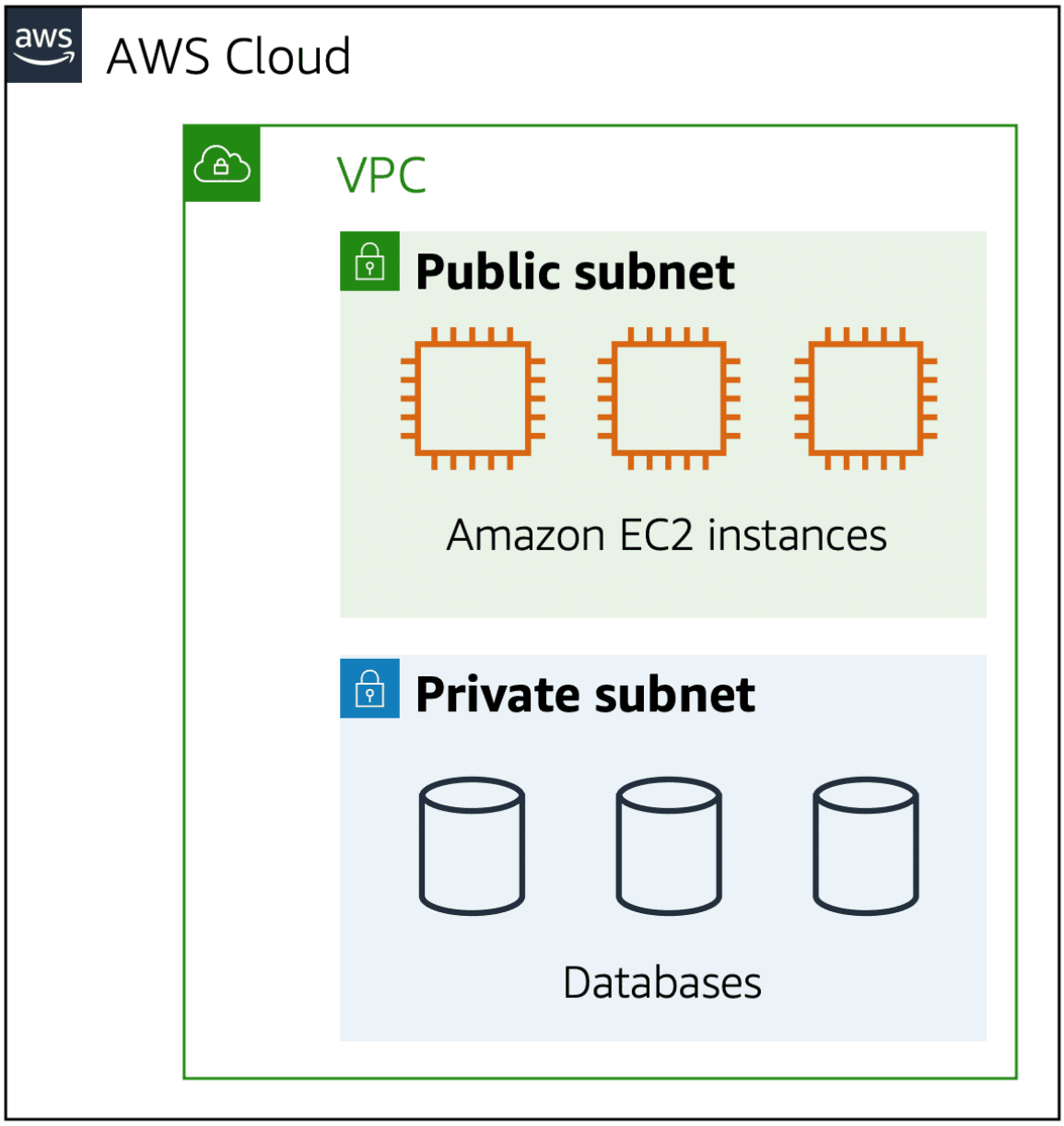


This is similar to how you can use AWS networking services to isolate resources and determine exactly how network traffic flows.

In the coffee shop, you can think of the counter area as a VPC. The counter area divides into two separate areas for the cashier's workstation and the barista's workstation. In a VPC, **subnets** are separate areas that are used to group together resources.

Subnets

A subnet is a section of a VPC in which you can group resources based on security or operational needs. Subnets can be public or private.



Public subnets contain resources that need to be accessible by the public, such as an online store's website.

Private subnets contain resources that should be accessible only through your private network, such as a database that contains customers' personal information and order histories.

In a VPC, subnets can communicate with each other. For example, you might have an application that involves Amazon EC2 instances in a public subnet communicating with databases that are located in a private subnet.

Network Traffic in a VPC

When a customer requests data from an application hosted in the AWS Cloud, this request is sent as a packet. A **packet** is a unit of data sent over the internet or a network.

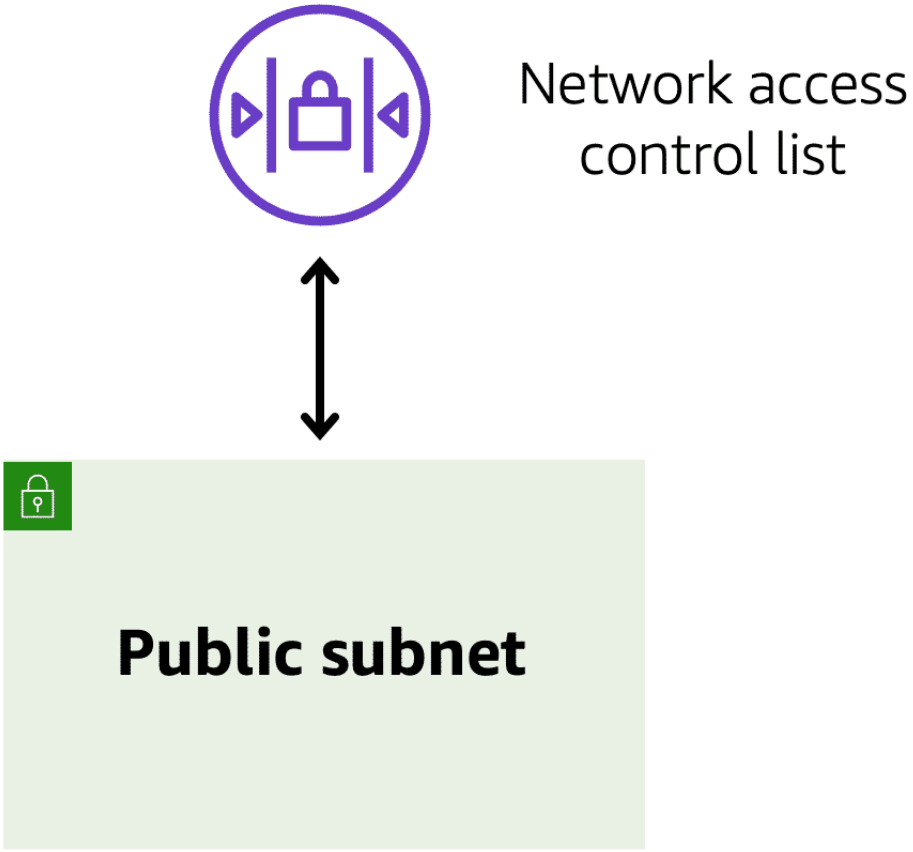
It enters into a VPC through an internet gateway. Before a packet can enter into a subnet or exit from a subnet, it checks for permissions. These permissions indicate who sent the packet and how the packet is trying to communicate with the resources in a subnet.

The VPC component that checks packet permissions for subnets is a **network access control list (ACL)**.

Network Access Control Lists (ACLs)

A network access control list (ACL) is a virtual firewall that controls inbound and outbound traffic at the subnet level.

For example, step outside of the coffee shop and imagine that you are in an airport. In the airport, travelers are trying to enter into a different country. You can think of the travelers as packets and the passport control officer as a network ACL. The passport control officer checks travelers' credentials when they are both entering and exiting out of the country. If a traveler is on an approved list, they are able to get through. However, if they are not on the approved list or are explicitly on a list of banned travelers, they cannot come in.



Each AWS account includes a default network ACL. When configuring your VPC, you can use your account's default network ACL or create custom network ACLs.

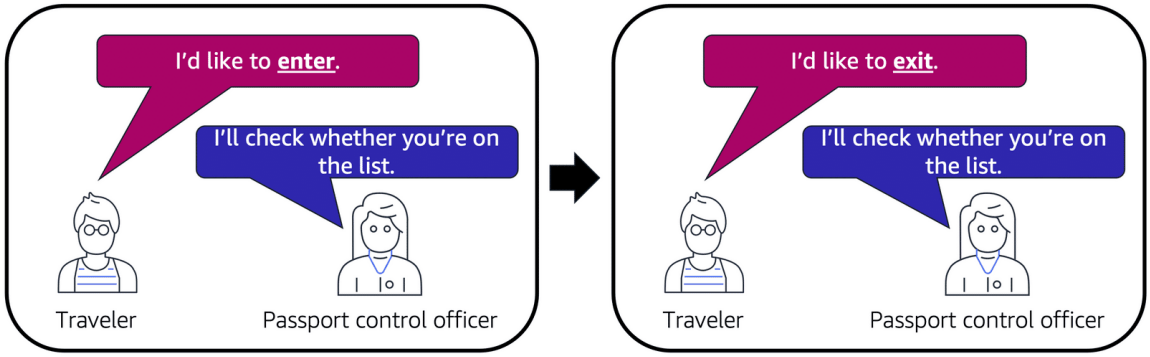
By default, your account's default network ACL allows all inbound and outbound traffic, but you can modify it by adding your own rules. For custom network ACLs, all inbound and outbound traffic is denied until you add rules to specify which traffic to allow. Additionally, all network ACLs have an explicit deny rule. This rule ensures that if a packet doesn't match any of the other rules on the list, the packet is denied.

Stateless Packet Filtering

Network ACLs perform **stateless** packet filtering. They remember nothing and check packets that cross the subnet border each way: inbound and outbound.

Recall the previous example of a traveler who wants to enter into a different country. This is similar to sending a request out from an Amazon EC2 instance and to the internet.

When a packet response for that request comes back to the subnet, the network ACL does not remember your previous request. The network ACL checks the packet response against its list of rules to determine whether to allow or deny.

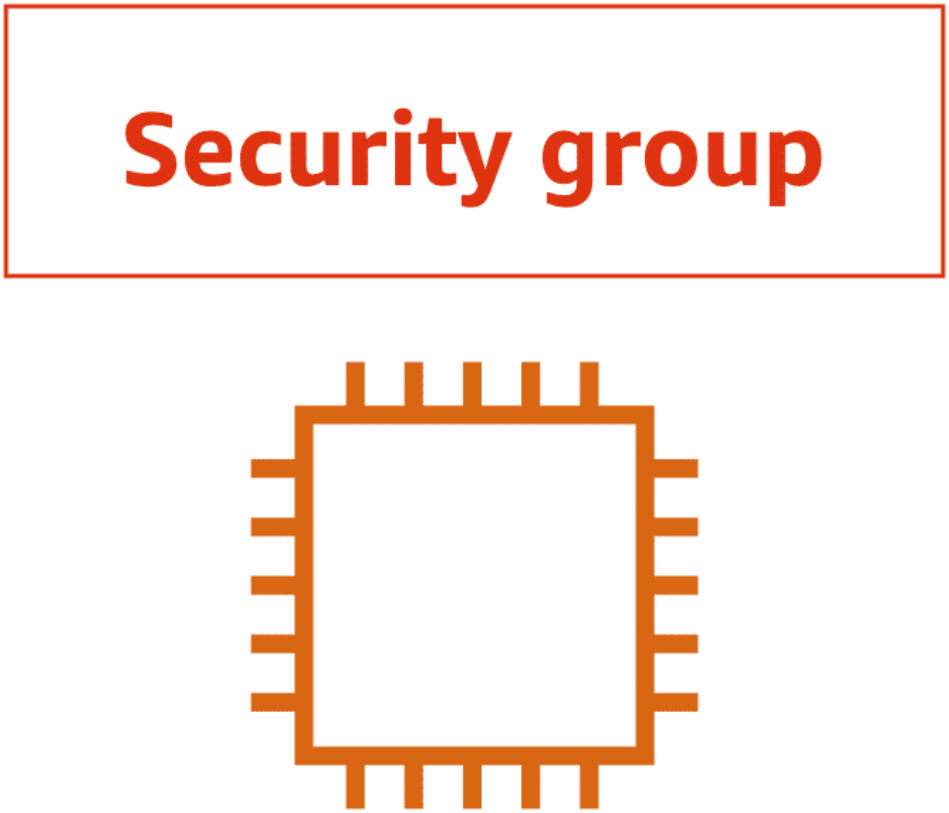


After a packet has entered a subnet, it must have its permissions evaluated for resources within the subnet, such as Amazon EC2 instances.

The VPC component that checks packet permissions for an Amazon EC2 instance is a **security group**.

Security Groups

A security group is a virtual firewall that controls inbound and outbound traffic for an Amazon EC2 instance.



By default, a security group denies all inbound traffic and allows all outbound traffic. You can add custom rules to configure which traffic to allow or deny.

For this example, suppose that you are in an apartment building with a door attendant who greets guests in the lobby. You can think of the guests as packets and the door attendant as a security group. As guests arrive, the door attendant checks a list to ensure they can enter the building. However, the door attendant does not check the list again when guests are exiting the building

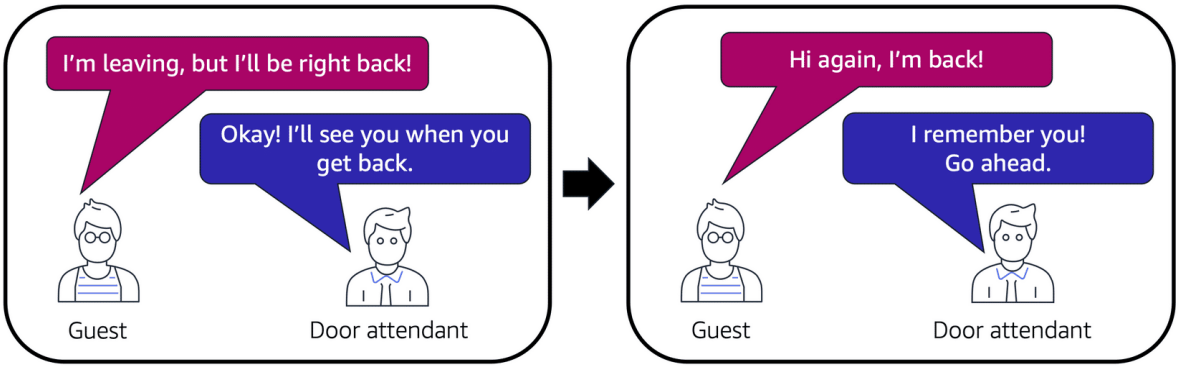
If you have multiple Amazon EC2 instances within a subnet, you can associate them with the same security group or use different security groups for each instance.

Stateful Packet Filtering

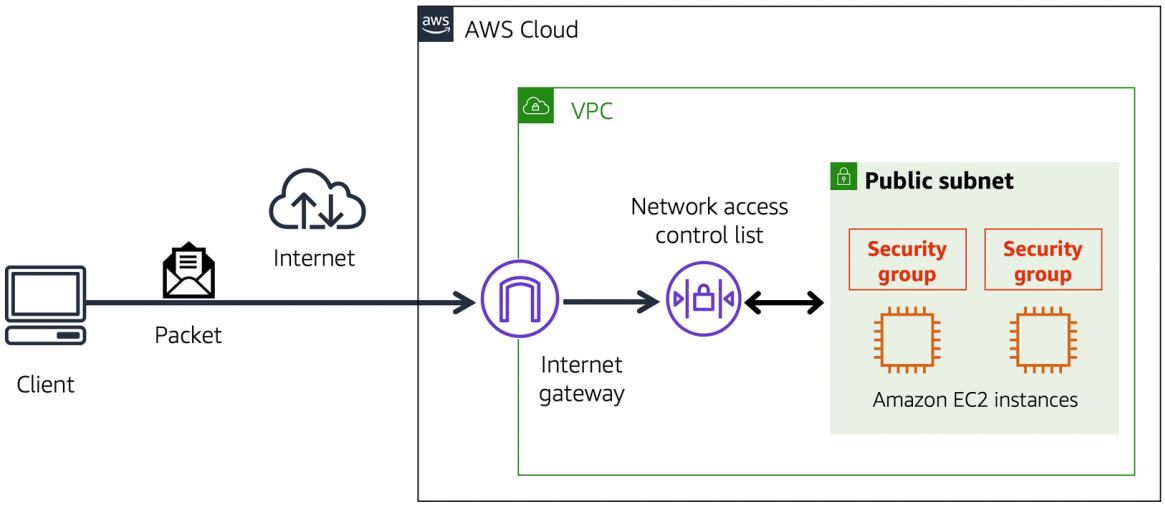
Security groups perform **stateful** packet filtering. They remember previous decisions made for incoming packets.

Consider the same example of sending a request out from an Amazon EC2 instance to the internet.

When a packet response for that request returns to the instance, the security group remembers your previous request. The security group allows the response to proceed, regardless of inbound security group rules.



Both network ACLs and security groups enable you to configure custom rules for the traffic in your VPC. As you continue to learn more about AWS security and networking, make sure to understand the differences between network ACLs and security groups.

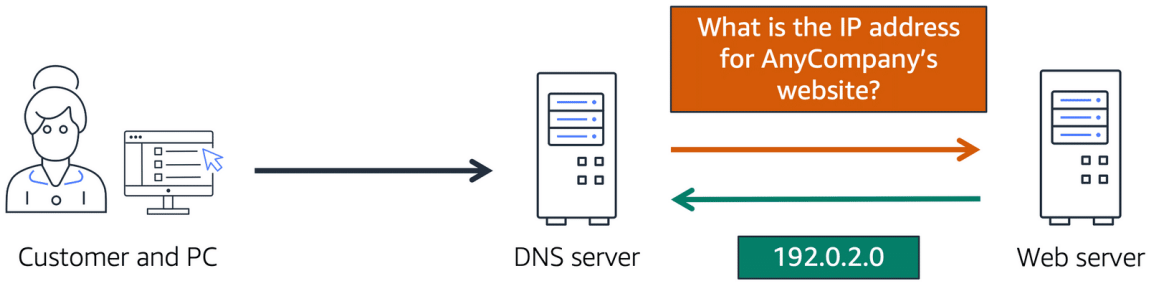


Global Networking

Domain Name System (DNS)

Suppose that AnyCompany has a website hosted in the AWS Cloud. Customers enter the web address into their browser, and they are able to access the website. This happens because of **Domain Name System (DNS)** resolution. DNS resolution involves a DNS server communicating with a web server.

You can think of DNS as being the phone book of the internet. DNS resolution is the process of translating a domain name to an IP address.



For example, suppose that you want to visit AnyCompany's website.

1. When you enter the domain name into your browser, this request is sent to a DNS server.
2. The DNS server asks the web server for the IP address that corresponds to AnyCompany's website.
3. The web server responds by providing the IP address for AnyCompany's website, 192.0.2.0.

Amazon Route 53

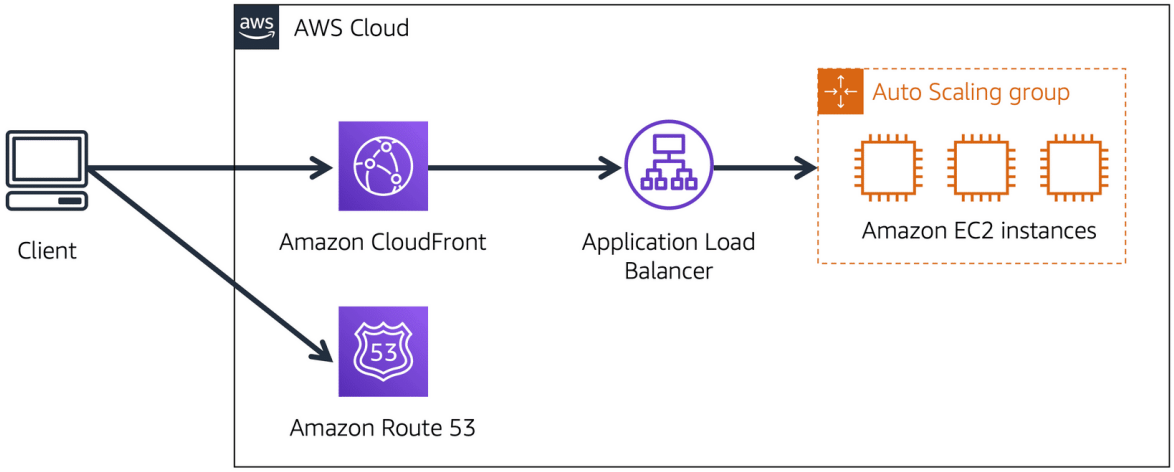
Amazon Route 53 is a DNS web service. It gives developers and businesses a reliable way to route end users to internet applications hosted in AWS.

Amazon Route 53 connects user requests to infrastructure running in AWS (such as Amazon EC2 instances and load balancers). It can route users to infrastructure outside of AWS.

Another feature of Route 53 is the ability to manage the DNS records for domain names. You can register new domain names directly in Route 53. You can also transfer DNS records for existing domain names managed by other domain registrars. This enables you to manage all of your domain names within a single location.

In the previous module, you learned about Amazon CloudFront, a content delivery service. The following example describes how Route 53 and Amazon CloudFront work together to deliver content to customers.

Example: How Amazon Route 53 and Amazon CloudFront deliver content



Suppose that AnyCompany's application is running on several Amazon EC2 instances. These instances are in an Auto Scaling group that attaches to an Application Load Balancer.

1. A customer requests data from the application by going to AnyCompany's website.
2. Amazon Route 53 uses DNS resolution to identify AnyCompany.com's corresponding IP address, 192.0.2.0. This information is sent back to the customer.
3. The customer's request is sent to the nearest edge location through Amazon CloudFront.
4. Amazon CloudFront connects to the Application Load Balancer, which sends the incoming packet to an Amazon EC2 instance.

Resources

To learn more about the concepts that were explored in Module 4, review these resources.

- Networking and Content Delivery on AWS
- AWS Networking & Content Delivery Blog
- Amazon Virtual Private Cloud
- What is Amazon VPC?
- How Amazon VPC works

Storage and Database

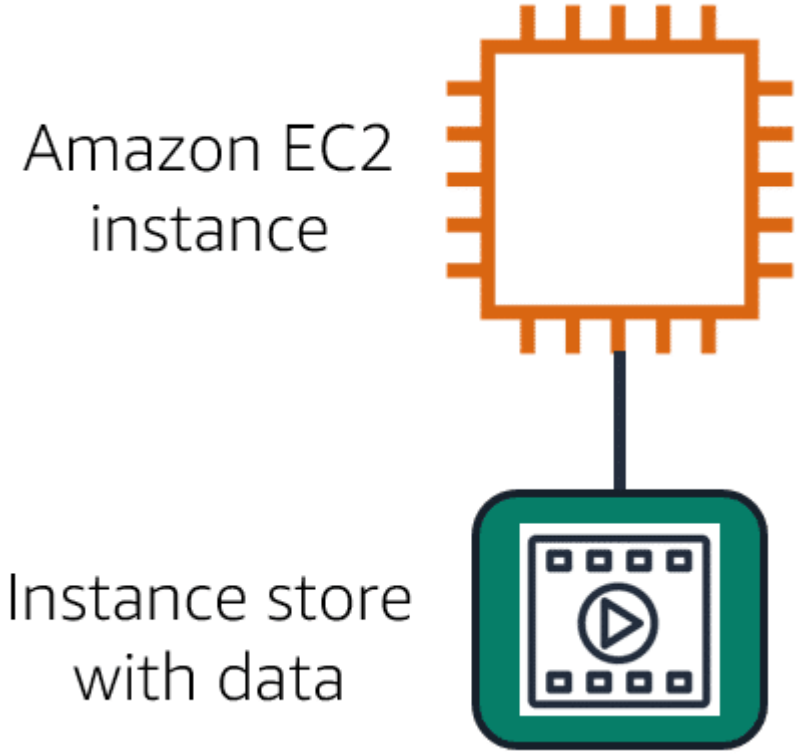
Instance Stores and Amazon Elastic Block Store (Amazon EBS)

Instance stores

Block-level storage volumes behave like physical hard drives.

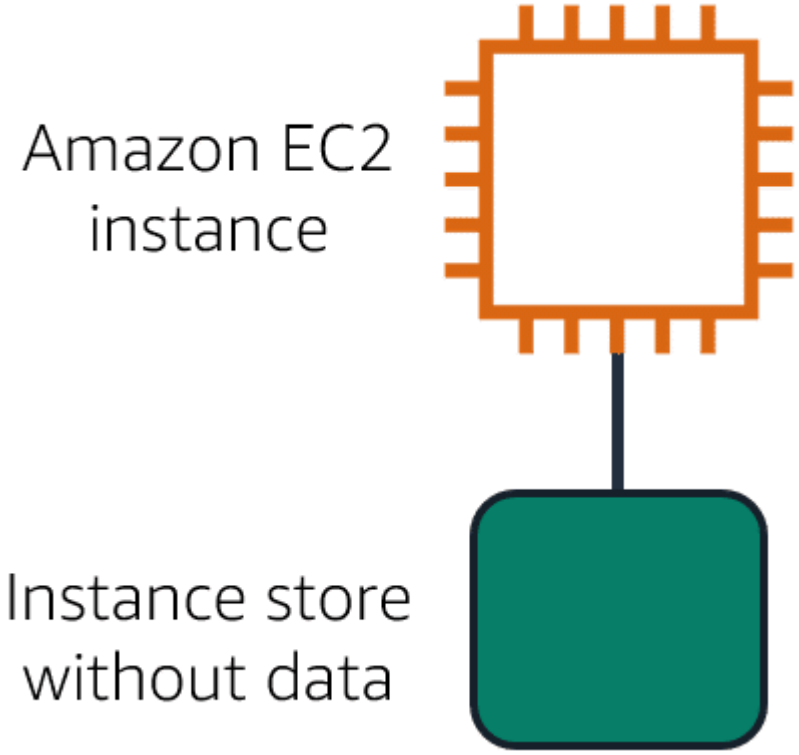
An **instance store** provides temporary block-level storage for an Amazon EC2 instance. An instance store is disk storage that is physically attached to the host computer for an EC2 instance, and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.

An Amazon EC2 instance with an attached instance store is running.



The instance is stopped or terminated.

All data on the attached instance store is deleted.

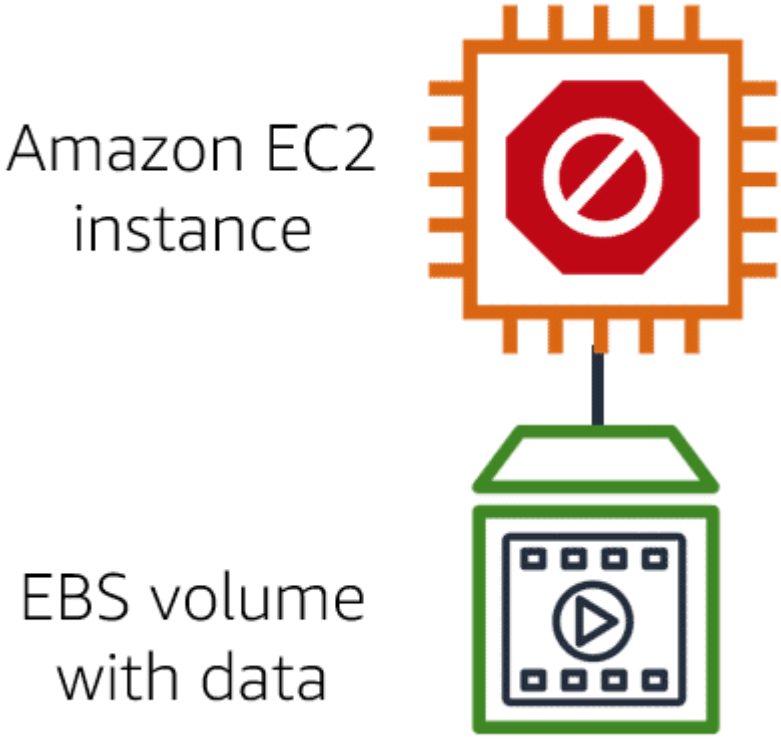


Amazon Elastic Block Storage (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) is a service that provides block-level storage volumes that you can use with Amazon EC2 instances. If you stop or terminate an Amazon EC2 instance, all the data on the attached EBS volume remains available.

To create an EBS volume, you define the configuration (such as volume size and type) and provision it. After you create an EBS volume, it can attach to an Amazon EC2 instance.

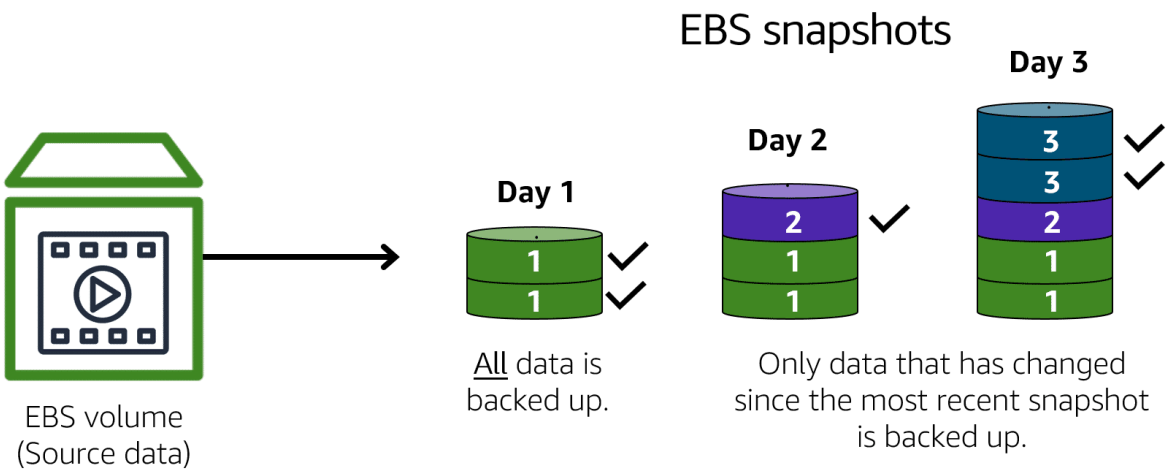
Because EBS volumes are for data that needs to persist, it's important to back up the data. You can take incremental backups of EBS volumes by creating Amazon EBS snapshots.



Amazon EBS Snapshots

An **EBS snapshot** is an incremental backup. This means that the first backup taken of a volume copies all the data. For subsequent backups, only the blocks of data that have changed since the most recent snapshot are saved.

Incremental backups are different from full backups, in which all the data in a storage volume copies each time a backup occurs. The full backup includes data that has not changed since the most recent backup.

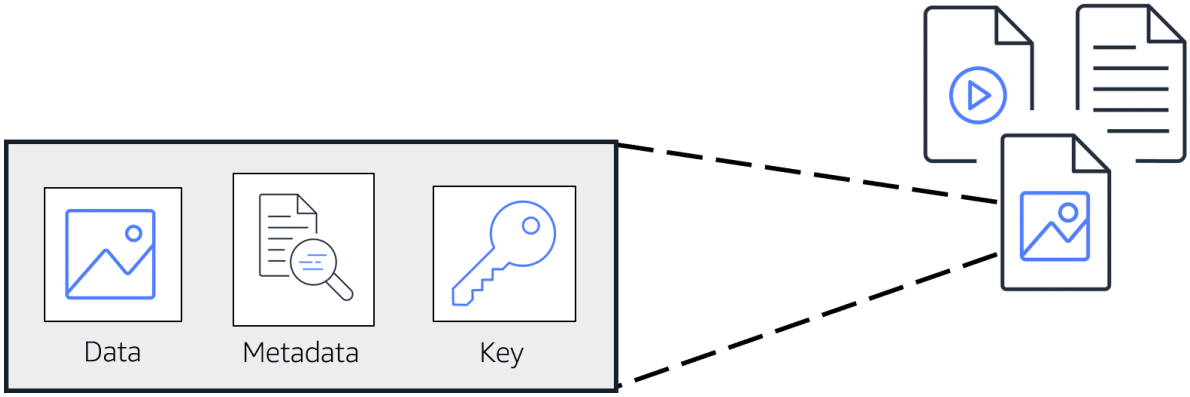


Amazon Simple Storage Service (Amazon S3)

Object Storage

In **object storage**, each object consists of data, metadata, and a key.

The data might be an image, video, text document, or any other type of file. Metadata contains information about what the data is, how it is used, the object size, and so on. An object's key is its unique identifier.



Recall that when you modify a file in block storage, only the pieces that are changed are updated. When a file in object storage is modified, the entire object is updated.

Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) is a service that provides object-level storage. Amazon S3 stores data as objects in buckets.

You can upload any type of file to Amazon S3, such as images, videos, text files, and so on. For example, you might use Amazon S3 to store backup files, media files for a website, or archived documents. Amazon S3 offers unlimited storage space. The maximum file size for an object in Amazon S3 is 5 TB.

When you upload a file to Amazon S3, you can set permissions to control visibility and access to it. You can also use the Amazon S3 versioning feature to track changes to your objects over time.

Amazon S3 Storage Classes

With Amazon S3, you pay only for what you use. You can choose from a range of storage classes to select a fit for your business and cost needs. When selecting an Amazon S3 storage class, consider these two factors:

- How often you plan to retrieve your data
- How available you need your data to be

Amazon S3 Standard

- Designed for frequently accessed data
- Stores data in a minimum of three Availability Zones

Amazon S3 Standard provides high availability for objects. This makes it a good choice for a wide range of use cases, such as websites, content distribution, and data analytics. Amazon S3 Standard has a higher cost than other storage classes intended for infrequently accessed data and archival storage.

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

- Ideal for infrequently accessed data
- Similar to Amazon S3 Standard but has a lower storage price and higher retrieval price

Amazon S3 Standard-IA is ideal for data infrequently accessed but requires high availability when needed. Both Amazon S3 Standard and Amazon S3 Standard-IA store data in a minimum of three Availability Zones. S3 Standard-IA provides the same level of availability as Amazon S3 Standard but with a lower storage price and a higher retrieval price.

Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)

- Stores data in a single Availability Zone
- Has a lower storage price than Amazon S3 Standard-IA

Compared to Amazon S3 Standard and Amazon S3 Standard-IA, which store data in a minimum of three Availability Zones, Amazon S3 One Zone-IA stores data in a single Availability Zone. This makes it a good storage class to consider if the following conditions apply:

- You want to save costs on storage.
- You can easily reproduce your data in the event of an Availability Zone failure.

Amazon S3 Intelligent-Tiering

- Ideal for data with unknown or changing access patterns
- Requires a small monthly monitoring and automation fee per object

In the Amazon S3 Intelligent-Tiering storage class, Amazon S3 monitors objects' access patterns. If you haven't accessed an object for 30 consecutive days, Amazon S3 automatically moves it to the infrequent access tier, Amazon S3 Standard-IA. If you access an object in the infrequent access tier, Amazon S3 automatically moves it to the frequent access tier, Amazon S3 Standard.

Amazon S3 Glacier Instant Retrieval

- Works well for archived data that requires immediate access
- Can retrieve objects within a few milliseconds

When you decide between the options for archival storage, consider how quickly you must retrieve the archived objects. You can retrieve objects stored in the Amazon S3 Glacier Instant Retrieval storage class within milliseconds, with the same performance as Amazon S3 Standard.

Amazon S3 Glacier Flexible Retrieval

- Low-cost storage designed for data archiving
- Able to retrieve objects within a few minutes to hours

Amazon S3 Glacier Flexible Retrieval is a low-cost storage class that is ideal for data archiving. For example, you might use this storage class to store archived customer records or older photos and video files.

Amazon S3 Glacier Deep Archive

- Lowest-cost object storage class ideal for archiving
- Able to retrieve objects within 12 hours

Amazon S3 Deep Archive supports long-term retention and digital preservation for data that might be accessed once or twice in a year. This storage class is the lowest-cost storage in the AWS Cloud, with data retrieval from 12 to 48 hours. All objects from this storage class are replicated and stored across at least three geographically dispersed Availability Zones.

Amazon S3 Outposts

- Creates S3 buckets on Amazon S3 Outposts
- Makes it easier to retrieve, store, and access data on AWS Outposts

Amazon S3 Outposts delivers object storage to your on-premises AWS Outposts environment. Amazon S3 Outposts is designed to store data durably and redundantly across multiple devices and servers on your Outposts. It works well for workloads with local data residency requirements that must satisfy demanding performance needs by keeping data close to on-premises applications.

Amazon Elastic File System (Amazon EFS)

File Storage

In **file storage**, multiple clients (such as users, applications, servers, and so on) can access data that is stored in shared file folders. In this approach, a storage server uses block storage with a local file system to organize files. Clients access data through file paths.

Compared to block storage and object storage, file storage is ideal for use cases in which a large number of services and resources need to access the same data at the same time.

Amazon Elastic File System (Amazon EFS) is a scalable file system used with AWS Cloud services and on-premises resources. As you add and remove files, Amazon EFS grows and shrinks automatically. It can scale on demand to petabytes without disrupting applications.

Comparing Amazon EBS and Amazon EFS

Amazon EBS

- An Amazon EBS volume stores data in a **single** Availability Zone.
- To attach an Amazon EC2 instance to an EBS volume, both the Amazon EC2 instance and the EBS volume must reside within the same Availability Zone.

Amazon EFS

- Amazon EFS is a regional service. It stores data in and across **multiple** Availability Zones.

- The duplicate storage enables you to access data concurrently from all the Availability Zones in the Region where a file system is located. Additionally, on-premises servers can access Amazon EFS using AWS Direct Connect.

Amazon Relational Database Service (Amazon RDS)

Relational databases

In a **relational database**, data is stored in a way that relates it to other pieces of data.

An example of a relational database might be the coffee shop's inventory management system. Each record in the database would include data for a single item, such as product name, size, price, and so on.

Relational databases use **structured query language (SQL)** to store and query data. This approach allows data to be stored in an easily understandable, consistent, and scalable way. For example, the coffee shop owners can write a SQL query to identify all the customers whose most frequently purchased drink is a medium latte.

Example of data in a relational database:

ID	Product Name	Size	Price
1	Medium roast ground coffee	12 oz.	$5.30
2	Dark roast ground coffee	20 oz.	$9.27

Amazon Relational Database Service

Amazon Relational Database Service (Amazon RDS) is a service that enables you to run relational databases in the AWS Cloud.

Amazon RDS is a managed service that automates tasks such as hardware provisioning, database setup, patching, and backups. With these capabilities, you can spend less time completing administrative tasks and more time using data to innovate your applications. You can integrate Amazon RDS with other services to fulfill your business and operational needs, such as using AWS Lambda to query your database from a serverless application.

Amazon RDS provides a number of different security options. Many Amazon RDS database engines offer encryption at rest (protecting data while it is stored) and encryption in transit (protecting data while it is being sent and received).

Amazon RDS database engines

Amazon RDS is available on six database engines, which optimize for memory, performance, or input/output (I/O). Supported database engines include:

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB

- Oracle Database
- Microsoft SQL Server

Amazon Aurora

Amazon Aurora is an enterprise-class relational database. It is compatible with MySQL and PostgreSQL relational databases. It is up to five times faster than standard MySQL databases and up to three times faster than standard PostgreSQL databases.

Amazon Aurora helps to reduce your database costs by reducing unnecessary input/output (I/O) operations, while ensuring that your database resources remain reliable and available.

Consider Amazon Aurora if your workloads require high availability. It replicates six copies of your data across three Availability Zones and continuously backs up your data to Amazon S3.

Amazon DynamoDB

Nonrelational Databases

In a **nonrelational database**, you create tables. A table is a place where you can store and query data.

Nonrelational databases are sometimes referred to as “NoSQL databases” because they use structures other than rows and columns to organize data. One type of structural approach for nonrelational databases is key-value pairs. With key-value pairs, data is organized into items (keys), and items have attributes (values). You can think of attributes as being different features of your data.

In a key-value database, you can add or remove attributes from items in the table at any time. Additionally, not every item in the table has to have the same attributes.

Example of data in a nonrelational database:

Key	Value
1	**Name:** John Doe **Address:** 123 Any Street **Favorite drink:** Medium latte
2	**Name:** Mary Major **Address:** 100 Main Street **Birthday:** July 5, 1994

Amazon DynamoDB

Amazon DynamoDB is a key-value database service. It delivers single-digit millisecond performance at any scale.

Serverless

- DynamoDB is serverless, which means that you do not have to provision, patch, or manage servers.
- You also do not have to install, maintain, or operate software.

Automatic Scaling

- As the size of your database shrinks or grows, DynamoDB automatically scales to adjust for changes in capacity while maintaining consistent performance.
- This makes it a suitable choice for use cases that require high performance while scaling.

Amazon Redshift

Amazon Redshift is a data warehousing service that you can use for big data analytics. It offers the ability to collect data from many sources and helps you to understand relationships and trends across your data.

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) enables you to migrate relational databases, nonrelational databases, and other types of data stores.

With AWS DMS, you move data between a source database and a target database. The source and target databases can be of the same type or different types. During the migration, your source database remains operational, reducing downtime for any applications that rely on the database.

For example, suppose that you have a MySQL database that is stored on premises in an Amazon EC2 instance or in Amazon RDS. Consider the MySQL database to be your source database. Using AWS DMS, you could migrate your data to a target database, such as an Amazon Aurora database.

Other use cases for AWS DMS

Development and test database migrations

- Enabling developers to test applications against production data without affecting production users

Database consolidation

- Combining several databases into a single database

Continuous replication

- Sending ongoing copies of your data to other target sources instead of doing a one-time migration

Additional Database Services

Amazon DocumentDB

Amazon DocumentDB is a document database service that supports MongoDB workloads. (MongoDB is a document database program.)

Amazon Neptune

Amazon Neptune is a graph database service.

You can use Amazon Neptune to build and run applications that work with highly connected datasets, such as recommendation engines, fraud detection, and knowledge graphs.

Amazon Quantum Ledger Database (Amazon QLDB)

Amazon Quantum Ledger Database (Amazon QLDB) is a ledger database service.

You can use Amazon QLDB to review a complete history of all the changes that have been made to your application data.

Amazon Managed Blockchain

Amazon Managed Blockchain is a service that you can use to create and manage blockchain networks with open-source frameworks.

Blockchain is a distributed ledger system that lets multiple parties run transactions and share data without a central authority.

Amazon ElastiCache

Amazon ElastiCache is a service that adds caching layers on top of your databases to help improve the read times of common requests.

It supports two types of data stores: Redis and Memcached.

Amazon DynamoDB Accelerator

Amazon DynamoDB Accelerator (DAX) is an in-memory cache for DynamoDB.

It helps improve response times from single-digit milliseconds to microseconds.

Resources

To learn more about the concepts that were explored in Module 5, review these resources.

- Cloud Storage on AWS
- AWS Storage Blog
- Hands-On Tutorials: Storage

- AWS Customer Stories: Storage
- AWS Database Migration Service
- Databases on AWS
- Category Deep Dive: Databases
- AWS Database Blog
- AWS Customer Stories: Databases

Security

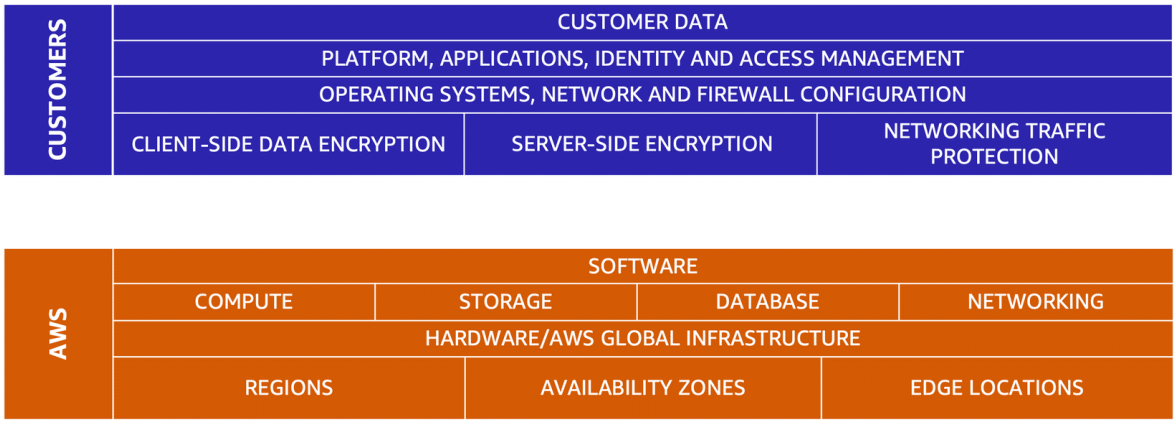
Shared Responsibility Model

The AWS Shared Responsibility Model

Throughout this course, you have learned about a variety of resources that you can create in the AWS Cloud. These resources include Amazon EC2 instances, Amazon S3 buckets, and Amazon RDS databases. Who is responsible for keeping these resources secure: you (the customer) or AWS?

The answer is both. The reason is that you do not treat your AWS environment as a single object. Rather, you treat the environment as a collection of parts that build upon each other. AWS is responsible for some parts of your environment and you (the customer) are responsible for other parts. This concept is known as the **shared responsibility model**.

The shared responsibility model divides into customer responsibilities (commonly referred to as “security in the cloud”) and AWS responsibilities (commonly referred to as “security of the cloud”).



You can think of this model as being similar to the division of responsibilities between a homeowner and a homebuilder. The builder (AWS) is responsible for constructing your house and ensuring that it is solidly built. As the homeowner (the customer), it is your responsibility to secure everything in the house by ensuring that the doors are closed and locked.

Customers: Security in the Cloud

Customers are responsible for the security of everything that they create and put *in* the AWS Cloud.

When using AWS services, you, the customer, maintain complete control over your content. You are responsible for managing security requirements for your content, including which content you choose to store on AWS, which AWS services you use, and who has access to that content. You also control how access rights are granted, managed, and revoked.

The security steps that you take will depend on factors such as the services that you use, the complexity of your systems, and your company's specific operational and security needs. Steps include selecting, configuring, and patching the operating systems that will run on Amazon EC2 instances, configuring security groups, and managing user accounts.

AWS: Security of the Cloud

AWS is responsible for security *of* the cloud.

AWS operates, manages, and controls the components at all layers of infrastructure. This includes areas such as the host operating system, the virtualization layer, and even the physical security of the data centers from which services operate.

AWS is responsible for protecting the global infrastructure that runs all of the services offered in the AWS Cloud. This infrastructure includes AWS Regions, Availability Zones, and edge locations.

AWS manages the security of the cloud, specifically the physical infrastructure that hosts your resources, which include:

- Physical security of data centers
- Hardware and software infrastructure
- Network infrastructure
- Virtualization infrastructure

Although you cannot visit AWS data centers to see this protection firsthand, AWS provides several reports from third-party auditors. These auditors have verified its compliance with a variety of computer security standards and regulations.

User Permission and Access

AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely.

IAM gives you the flexibility to configure access based on your company's specific operational and security needs. You do this by using a combination of IAM features, which are explored in detail in this lesson:

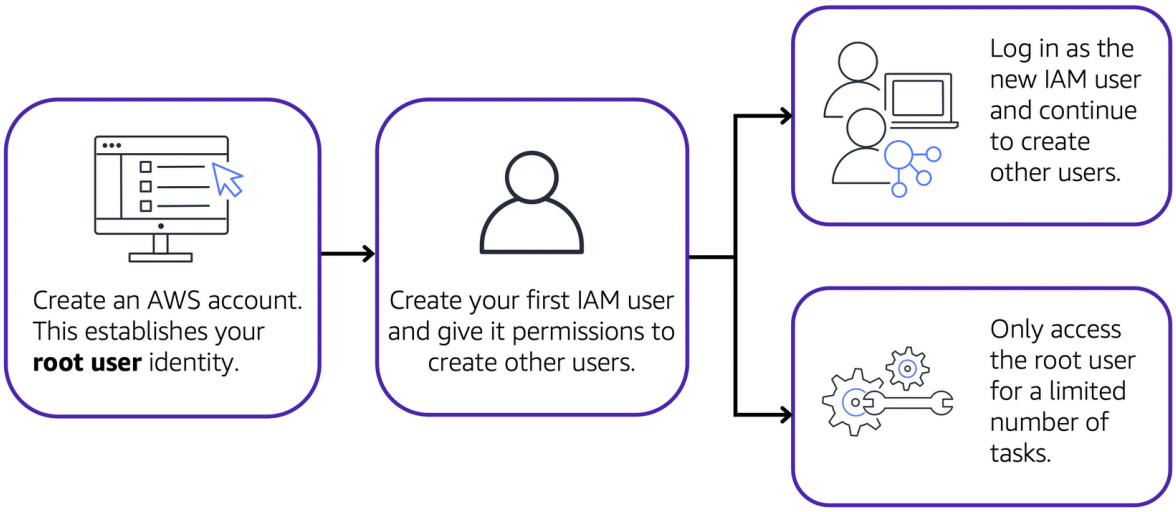
- IAM users, groups, and roles
- IAM policies
- Multi-factor authentication

You will also learn best practices for each of these features.

AWS Account Root User

When you first create an AWS account, you begin with an identity known as the **root user**.

The root user is accessed by signing in with the email address and password that you used to create your AWS account. You can think of the root user as being similar to the owner of the coffee shop. It has complete access to all the AWS services and resources in the account.



Best practice:

Do **not** use the root user for everyday tasks.

Instead, use the root user to create your first IAM user and assign it permissions to create other users.

Then, continue to create other IAM users, and access those identities for performing regular tasks throughout AWS. Only use the root user when you need to perform a limited number of tasks that are only available to the root user. Examples of these tasks include changing your root user email address and changing your AWS support plan.

IAM users

An **IAM user** is an identity that you create in AWS. It represents the person or application that interacts with AWS services and resources. It consists of a name and credentials.

By default, when you create a new IAM user in AWS, it has no permissions associated with it. To allow the IAM user to perform specific actions in AWS, such as launching an Amazon EC2 instance or creating an Amazon S3 bucket, you must grant the IAM user the necessary permissions.

Best practice:

We recommend that you create individual IAM users for each person who needs to access AWS.

Even if you have multiple employees who require the same level of access, you should create individual IAM users for each of them. This provides additional security by allowing each IAM user to have a unique set of security credentials.

IAM policies

An **IAM policy** is a document that allows or denies permissions to AWS services and resources.

IAM policies enable you to customize users' levels of access to resources. For example, you can allow users to access all of the Amazon S3 buckets within your AWS account, or only a specific bucket.

Best practice:

Follow the security principle of **least privilege** when granting permissions.

By following this principle, you help to prevent users or roles from having more permissions than needed to perform their tasks.

For example, if an employee needs access to only a specific bucket, specify the bucket in the IAM policy. Do this instead of granting the employee access to all of the buckets in your AWS account.

Example: IAM policy

Here's an example of how IAM policies work. Suppose that the coffee shop owner has to create an IAM user for a newly hired cashier. The cashier needs access to the receipts kept in an Amazon S3 bucket with the ID: AWSDOC-EXAMPLE-BUCKET.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListObject",
    "Resource": "arn:aws:s3:::
AWSDOC-EXAMPLE-BUCKET"
  }
}
```

In this example, the IAM policy is allowing a specific action within Amazon S3: ListObject. The policy also mentions a specific bucket ID: AWSDOC-EXAMPLE-BUCKET. When the owner attaches this policy to the cashier's IAM user, it will allow the cashier to view all of the objects in the AWSDOC-EXAMPLE-BUCKET bucket.

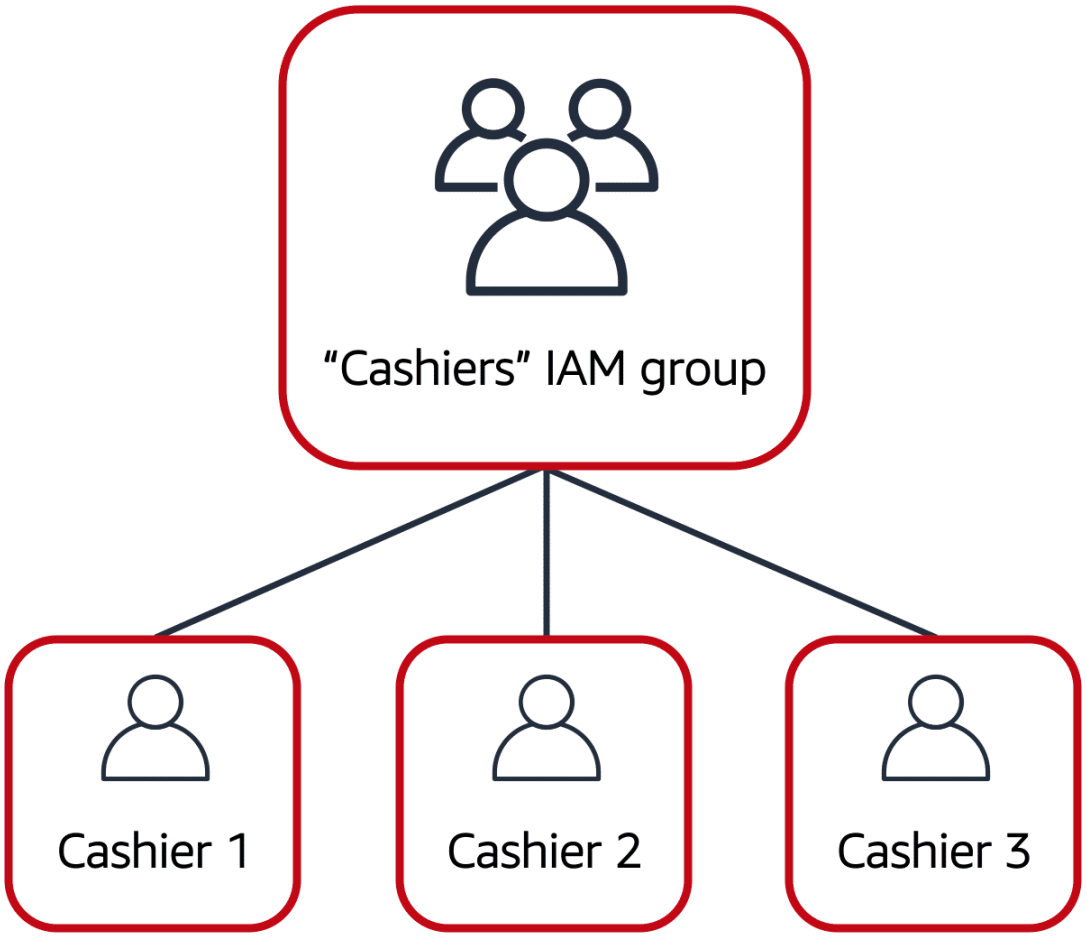
If the owner wants the cashier to be able to access other services and perform other actions in AWS, the owner must attach additional policies to specify these services and actions.

Now, suppose that the coffee shop has hired a few more cashiers. Instead of assigning permissions to each individual IAM user, the owner places the users into an **IAM group**.

IAM Groups

An IAM group is a collection of IAM users. When you assign an IAM policy to a group, all users in the group are granted permissions specified by the policy.

Here's an example of how this might work in the coffee shop. Instead of assigning permissions to cashiers one at a time, the owner can create a "Cashiers" IAM group. The owner can then add IAM users to the group and then attach permissions at the group level.



Assigning IAM policies at the group level also makes it easier to adjust permissions when an employee transfers to a different job. For example, if a cashier becomes an inventory specialist, the coffee shop owner removes them from the "Cashiers" IAM group and adds them into the "Inventory Specialists" IAM group. This ensures that employees have only the permissions that are required for their current role.

What if a coffee shop employee hasn't switched jobs permanently, but instead, rotates to different workstations throughout the day? This employee can get the access they need through **IAM roles**.

IAM Roles

In the coffee shop, an employee rotates to different workstations throughout the day. Depending on the staffing of the coffee shop, this employee might perform several duties: work at the cash register, update the inventory system, process online orders, and so on.

When the employee needs to switch to a different task, they give up their access to one workstation and gain access to the next workstation. The employee can easily switch between workstations, but at any given point in time, they can have access to only a single workstation. This same concept exists in AWS with IAM roles.

An IAM role is an identity that you can assume to gain temporary access to permissions.

Before an IAM user, application, or service can assume an IAM role, they must be granted permissions to switch to the role. When someone assumes an IAM role, they abandon all previous permissions that they had under a previous role and assume the permissions of the new role.

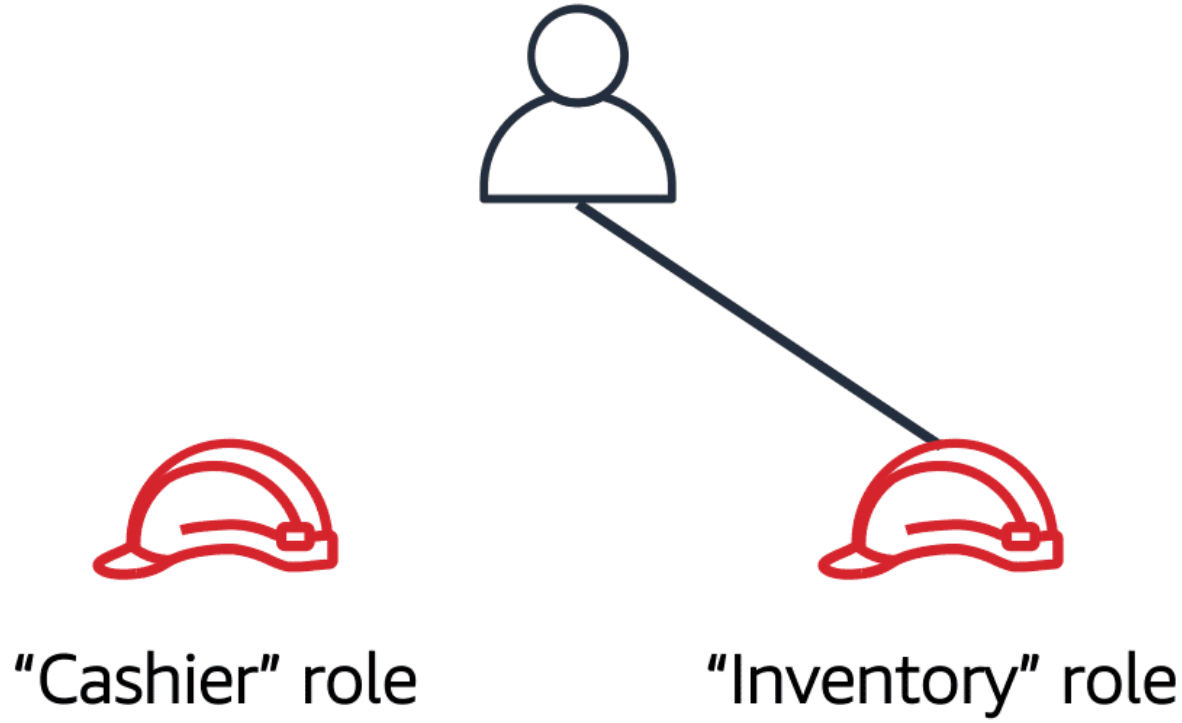
Best practice:

IAM roles are ideal for situations in which access to services or resources needs to be granted temporarily, instead of long-term.

Example: IAM Roles

Review an example of how IAM roles could be used in the coffee shop:

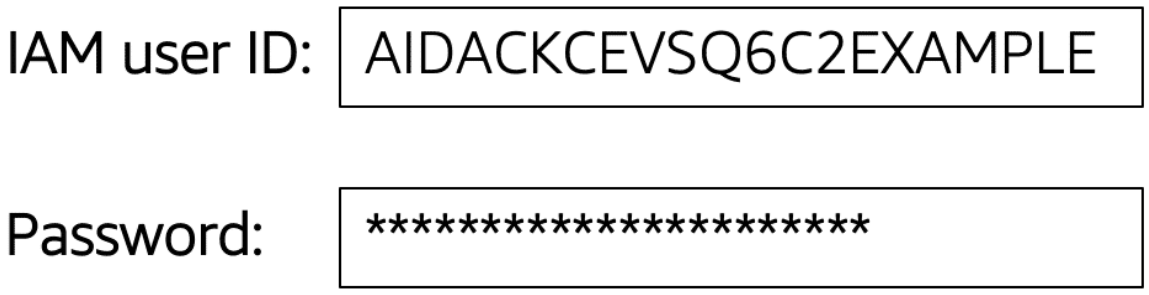
1. First, the owner grants the employee permissions to switch to a role for each workstation in the coffee shop.
2. Next, the employee begins their day by assuming the "Cashier" role. This grants them access to the cash register system.
3. Later in the day, the employee needs to update the inventory system. They assume the "Inventory" role. This grants the employee access to the inventory system and also revokes their access to the cash register system.



Multi-factor Authentication

Have you ever signed in to a website that required you to provide multiple pieces of information to verify your identity? You might have needed to provide your password and then a second form of authentication, such as a random code sent to your phone. This is an example of **multi-factor authentication**.

In IAM, multi-factor authentication (MFA) provides an extra layer of security for your AWS account.



1. First, when a user signs in to an AWS website, they enter their IAM user ID and password.
2. Next, the user is prompted for an authentication response from their AWS MFA device. This device could be a hardware security key, a hardware device, or an MFA application on a device such as a smartphone.
3. When the user has been successfully authenticated, they are able to access the requested AWS services or resources.

You can enable MFA for the root user and IAM users. As a best practice, enable MFA for the root user and all IAM users in your account. By doing this, you can keep your AWS account safe from unauthorized access.

AWS Organizations

AWS Organizations

Suppose that your company has multiple AWS accounts. You can use **AWS Organizations** to consolidate and manage multiple AWS accounts within a central location.

When you create an organization, AWS Organizations automatically creates a **root**, which is the parent container for all the accounts in your organization.

In AWS Organizations, you can centrally control permissions for the accounts in your organization by using **service control policies (SCPs)**. SCPs enable you to place restrictions on the AWS services, resources, and individual API actions that users and roles in each account can access.

Consolidated billing is another feature of AWS Organizations. You will learn about consolidated billing in a later module.

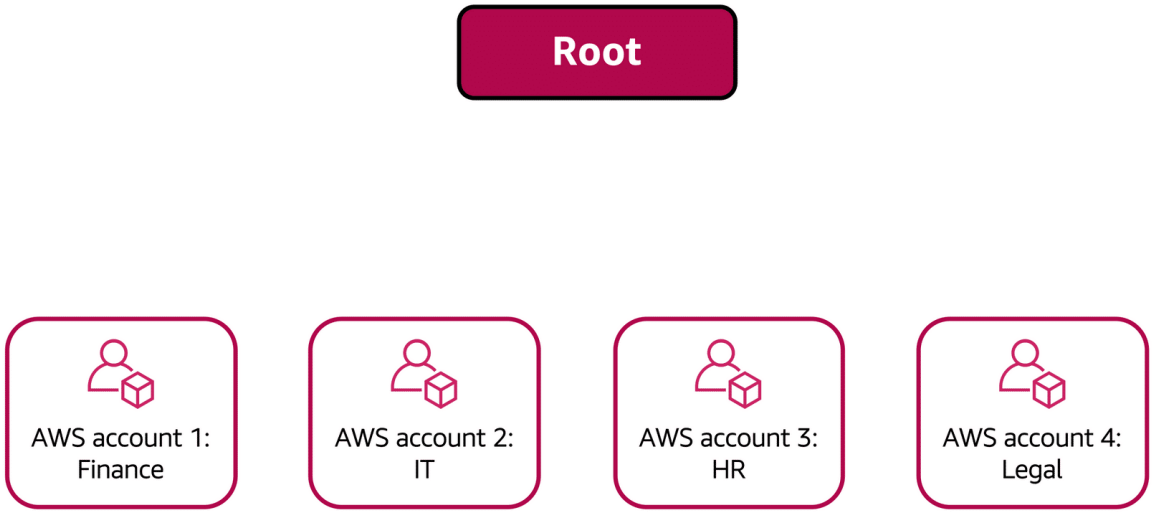
Organizational Units

In AWS Organizations, you can group accounts into organizational units (OUs) to make it easier to manage accounts with similar business or security requirements. When you apply a policy to an OU, all the accounts in the OU automatically inherit the permissions specified in the policy.

By organizing separate accounts into OUs, you can more easily isolate workloads or applications that have specific security requirements. For instance, if your company has accounts that can access only the AWS services that meet certain regulatory requirements, you can put these accounts into one OU. Then, you can attach a policy to the OU that blocks access to all other AWS services that do not meet the regulatory requirements.

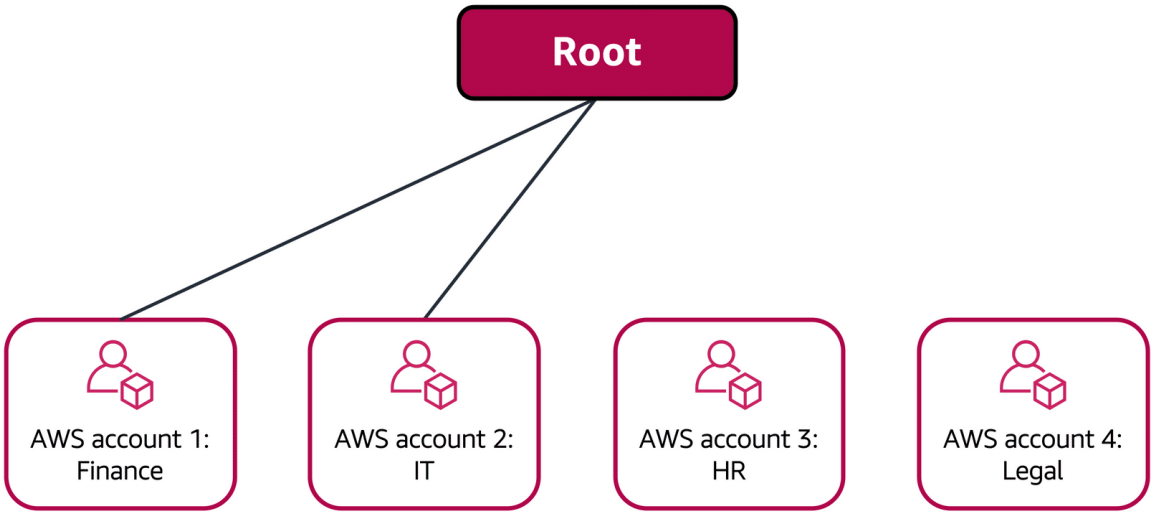
Example: AWS Organizations

Review an example of how a company might use AWS Organizations:

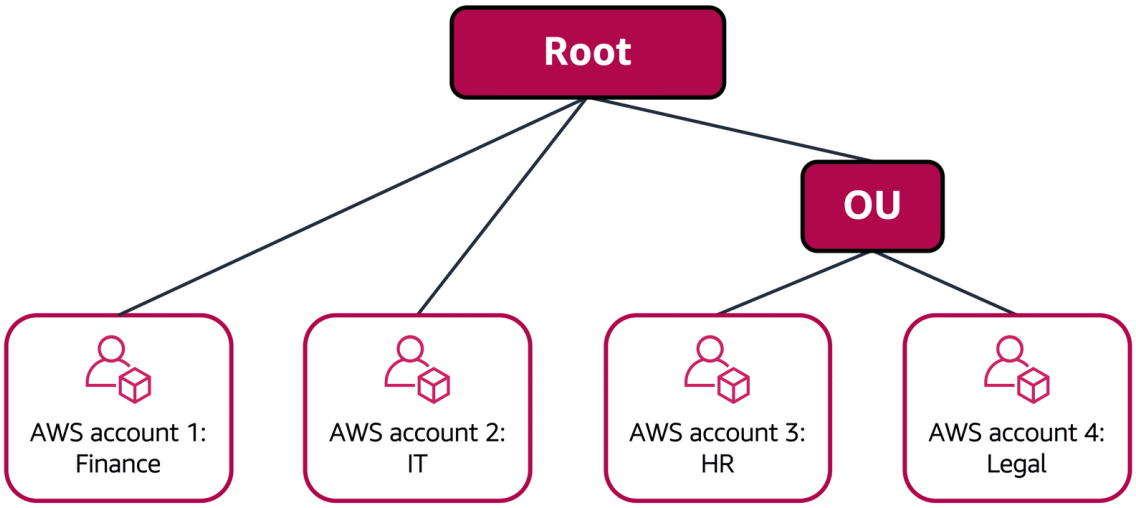


Imagine that your company has separate AWS accounts for the finance, information technology (IT), human resources (HR), and legal departments. You decide to consolidate these accounts into a single organization so that you can administer them from a central location. When you create the organization, this establishes the root.

In designing your organization, you consider the business, security, and regulatory needs of each department. You use this information to decide which departments group together in OUs.



The finance and IT departments have requirements that do not overlap with those of any other department. You bring these accounts into your organization to take advantage of benefits such as consolidated billing, but you do not place them into any OUs.



The HR and legal departments need to access the same AWS services and resources, so you place them into an OU together. Placing them into an OU enables you to attach policies that apply to both the HR and legal departments' AWS accounts.

Even though you have placed these accounts into OUs, you can continue to provide access for users, groups, and roles through IAM.

By grouping your accounts into OUs, you can more easily give them access to the services and resources that they need. You also prevent them from accessing any services or resources that they do not need.

Compliance

AWS Artifact

Depending on your company's industry, you may need to uphold specific standards. An audit or inspection will ensure that the company has met those standards.

AWS Artifact is a service that provides on-demand access to AWS security and compliance reports and select online agreements. AWS Artifact consists of two main sections: AWS Artifact Agreements and AWS Artifact Reports.

AWS Artifact Agreements

Suppose that your company needs to sign an agreement with AWS regarding your use of certain types of information throughout AWS services. You can do this through **AWS Artifact Agreements**.

In AWS Artifact Agreements, you can review, accept, and manage agreements for an individual account and for all your accounts in AWS Organizations. Different types of agreements are offered to address the needs of customers who are subject to specific regulations, such as the Health Insurance Portability and Accountability Act (HIPAA).

AWS Artifact Reports

Next, suppose that a member of your company's development team is building an application and needs more information about their responsibility for complying with certain regulatory standards. You can advise them to access this information in **AWS Artifact Reports**.

AWS Artifact Reports provide compliance reports from third-party auditors. These auditors have tested and verified that AWS is compliant with a variety of global, regional, and industry-specific security standards and regulations. AWS Artifact Reports remains up to date with the latest reports released. You can provide the AWS audit artifacts to your auditors or regulators as evidence of AWS security controls.

The following are some of the compliance reports and regulations that you can find within AWS Artifact. Each report includes a description of its contents and the reporting period for which the document is valid.

Denial-of-Service Attacks

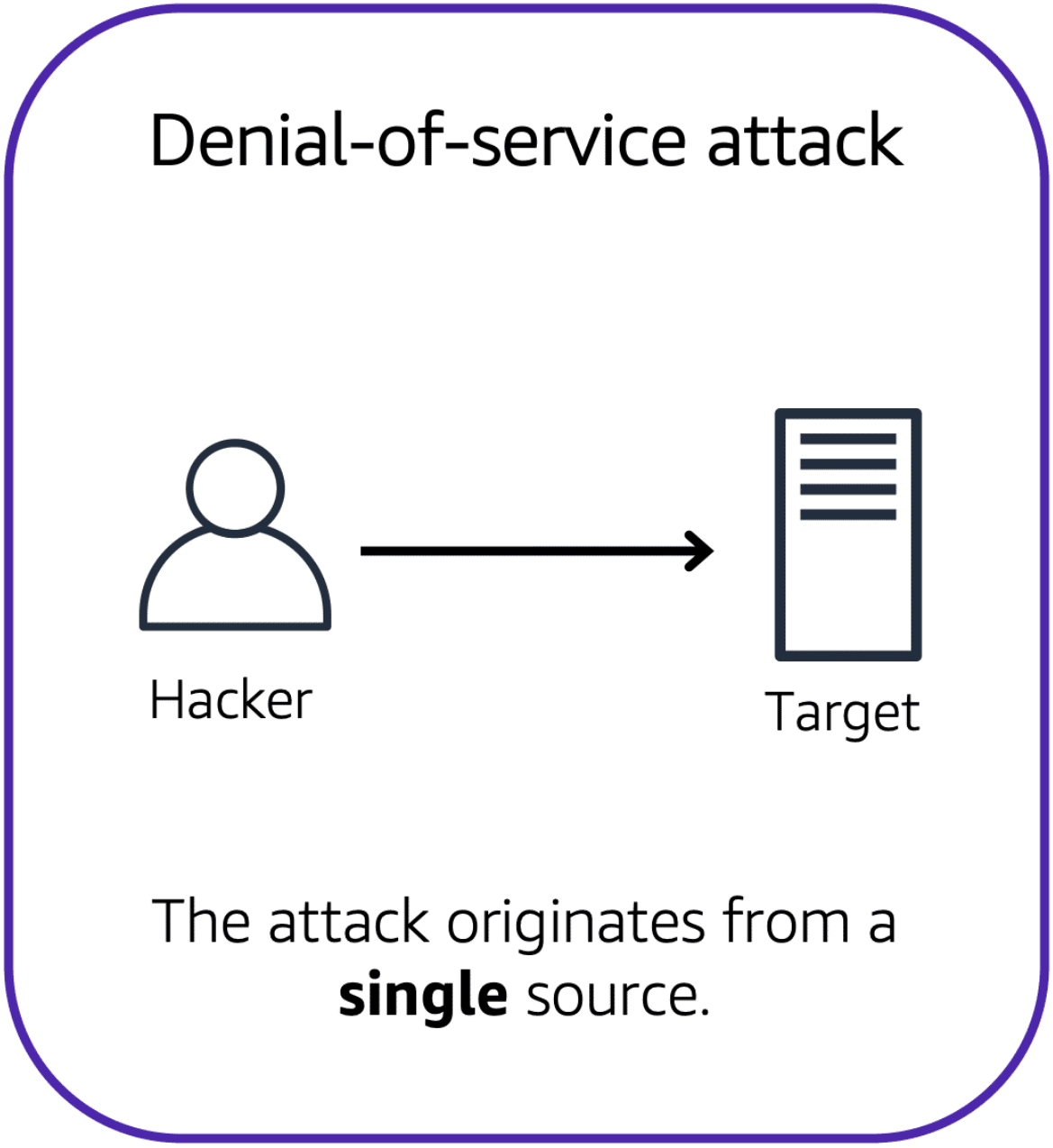
Customers can call the coffee shop to place their orders. After answering each call, a cashier takes the order and gives it to the barista.

However, suppose that a prankster is calling in multiple times to place orders but is never picking up their drinks. This causes the cashier to be unavailable to take other customers' calls. The coffee shop can attempt to stop the false requests by blocking the phone number that the prankster is using.

In this scenario, the prankster's actions are similar to a **denial-of-service attack**.

Denial-of-Service Attacks

A **denial-of-service (DoS) attack** is a deliberate attempt to make a website or application unavailable to users.

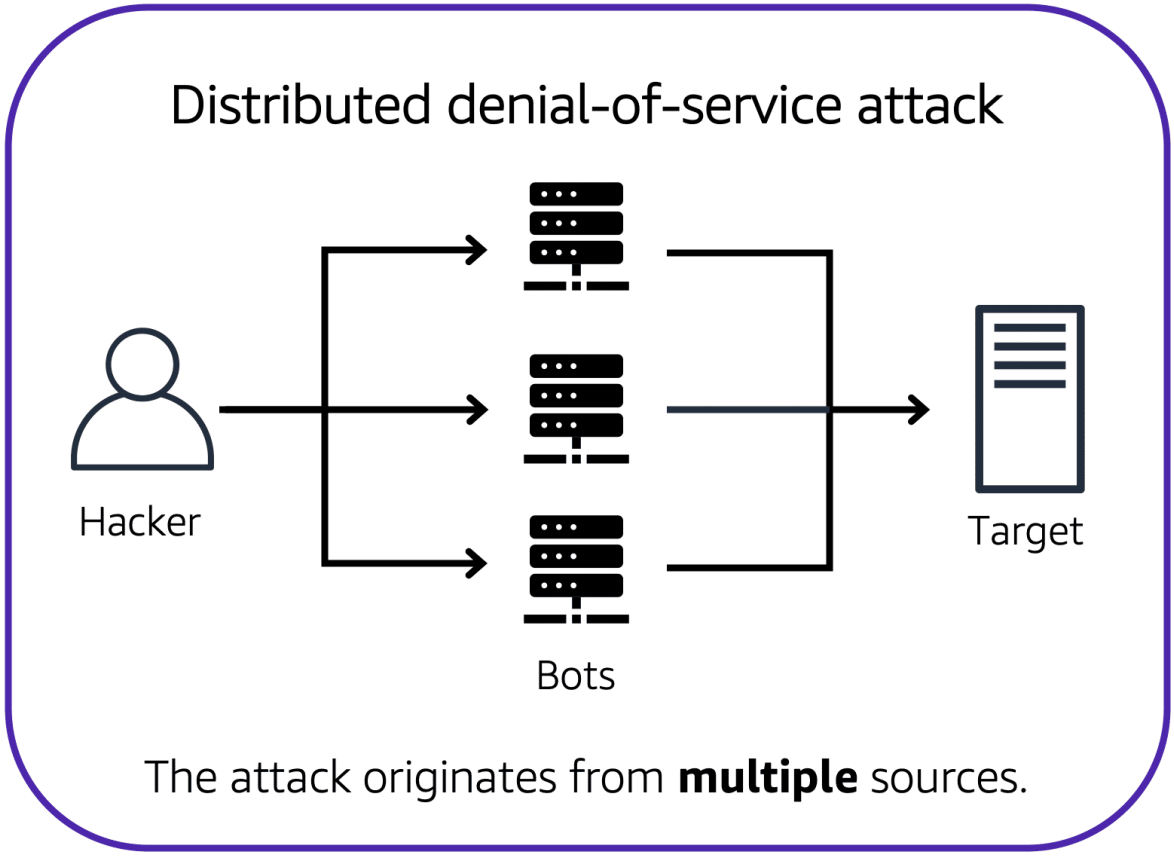


For example, an attacker might flood a website or application with excessive network traffic until the targeted website or application becomes overloaded and is no longer able to respond. If the website or application becomes unavailable, this denies service to users who are trying to make legitimate requests.

Distributed Denial-of-Service Attacks

Now, suppose that the prankster has enlisted the help of friends.

The prankster and their friends repeatedly call the coffee shop with requests to place orders, even though they do not intend to pick them up. These requests are coming in from different phone numbers, and it's impossible for the coffee shop to block them all. Additionally, the influx of calls has made it increasingly difficult for customers to be able to get their calls through. This is similar to a **distributed denial-of-service attack**.



In a distributed denial-of-service (DDoS) attack, multiple sources are used to start an attack that aims to make a website or application unavailable. This can come from a group of attackers, or even a single attacker. The single attacker can use multiple infected computers (also known as “bots”) to send excessive traffic to a website or application.

To help minimize the effect of DoS and DDoS attacks on your applications, you can use **AWS Shield**.

AWS Shield

AWS Shield is a service that protects applications against DDoS attacks. AWS Shield provides two levels of protection: Standard and Advanced.

AWS Shield Standard

AWS Shield Standard automatically protects all AWS customers at no cost. It protects your AWS resources from the most common, frequently occurring types of DDoS attack.

As network traffic comes into your applications, AWS Shield Standard uses a variety of analysis techniques to detect malicious traffic in real time and automatically mitigates it.

AWS Shield Advanced

AWS Shield Advanced is a paid service that provides detailed attack diagnostics and the ability to detect and mitigate sophisticated DDoS attacks.

It also integrates with other services such as Amazon CloudFront, Amazon Route 53, and Elastic Load Balancing. Additionally, you can integrate AWS Shield with AWS WAF by writing custom rules to mitigate complex DDoS attacks.

Additional Security Services

AWS Key Management Service (AWS KMS)

The coffee shop has many items, such as coffee machines, pastries, money in the cash registers, and so on. You can think of these items as data. The coffee shop owners want to ensure that all of these items are secure, whether they're sitting in the storage room or being transported between shop locations.

In the same way, you must ensure that your applications' data is secure while in storage **(encryption at rest)** and while it is transmitted, known as **encryption in transit**.

AWS Key Management Service (AWS KMS) enables you to perform encryption operations through the use of **cryptographic keys**. A cryptographic key is a random string of digits used for locking (encrypting) and unlocking (decrypting) data. You can use AWS KMS to create, manage, and use cryptographic keys. You can also control the use of keys across a wide range of services and in your applications.

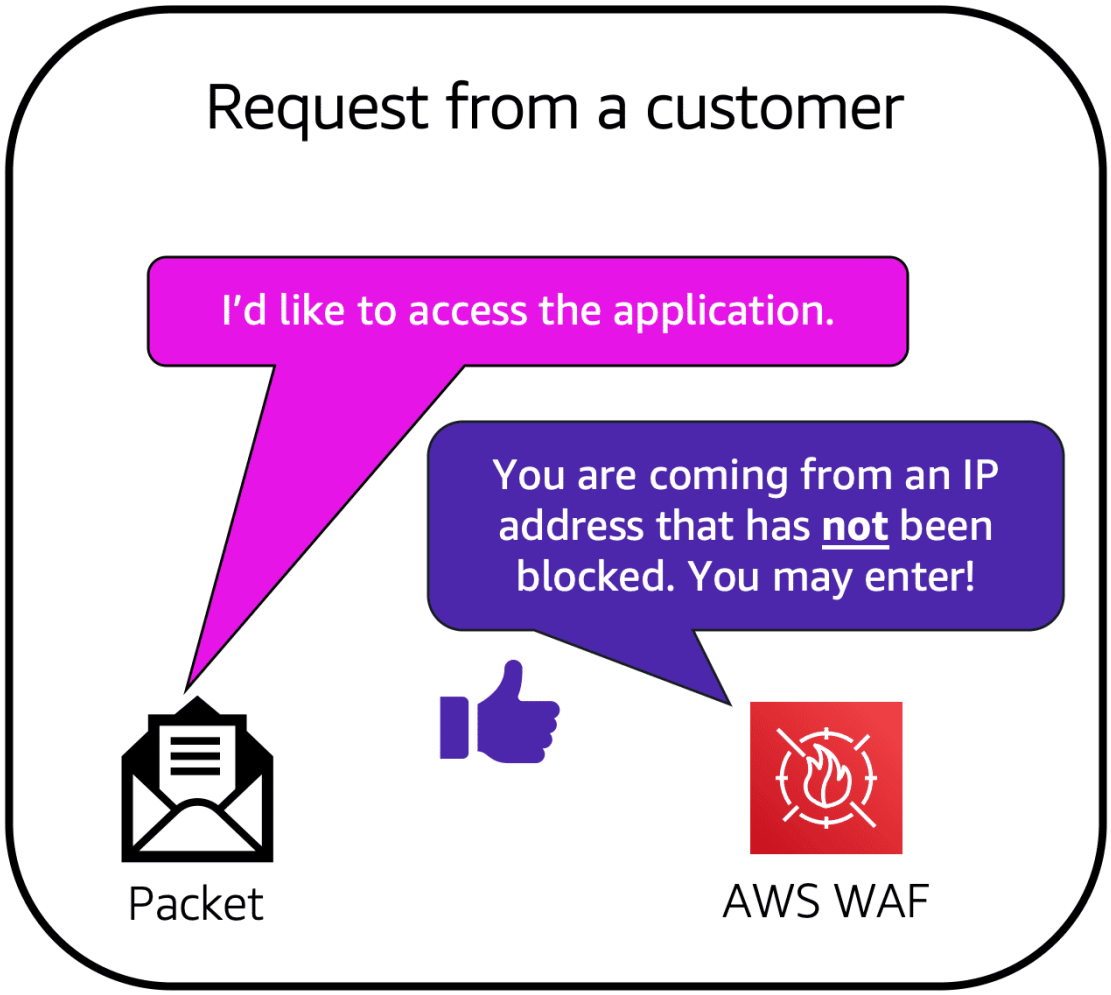
With AWS KMS, you can choose the specific levels of access control that you need for your keys. For example, you can specify which IAM users and roles are able to manage keys. Alternatively, you can temporarily disable keys so that they are no longer in use by anyone. Your keys never leave AWS KMS, and you are always in control of them.

AWS WAF

AWS WAF is a web application firewall that lets you monitor network requests that come into your web applications.

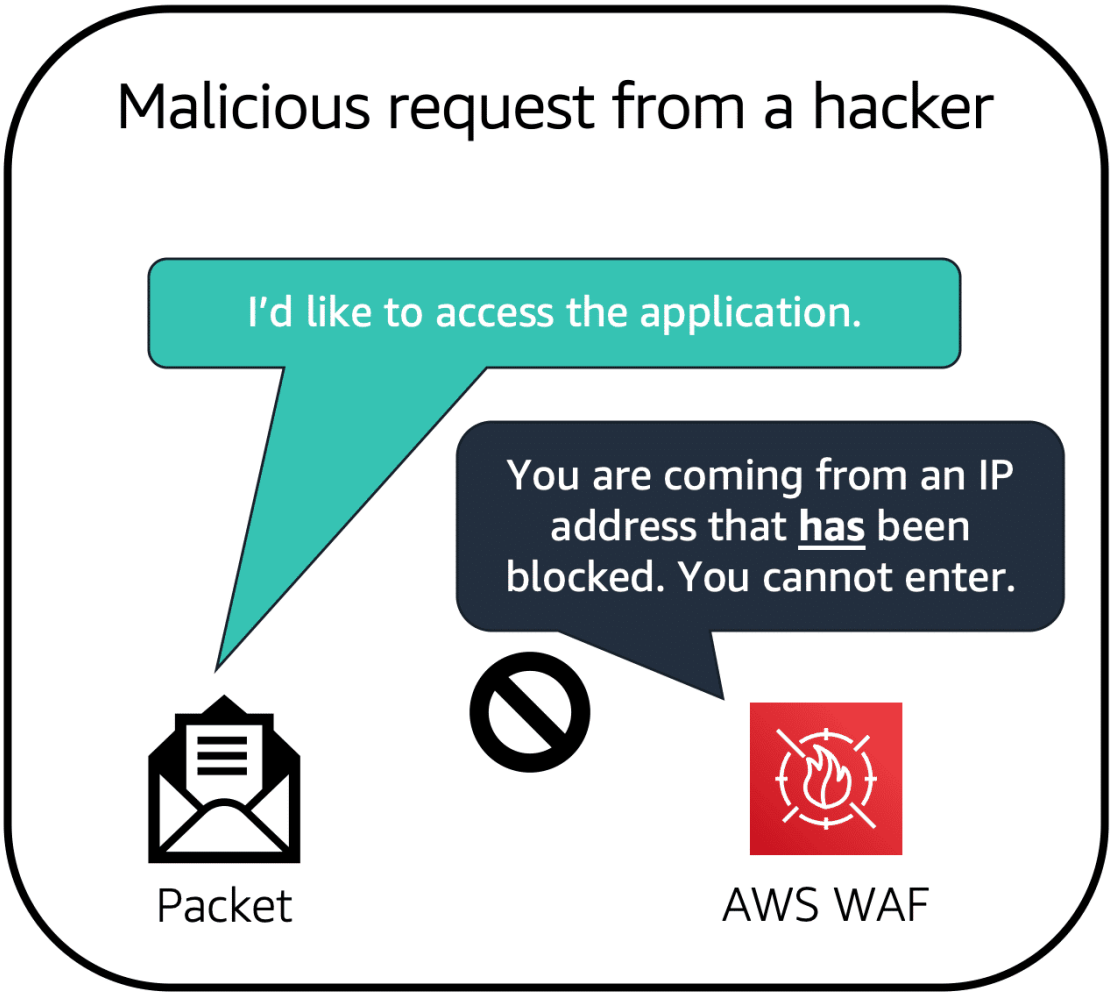
AWS WAF works together with Amazon CloudFront and an Application Load Balancer. Recall the network access control lists that you learned about in an earlier module. AWS WAF works in a similar way to block or allow traffic. However, it does this by using a **web access control list (ACL)** to protect your AWS resources.

Here's an example of how you can use AWS WAF to allow and block specific requests.



Suppose that your application has been receiving malicious network requests from several IP addresses. You want to prevent these requests from continuing to access your application, but you also want to ensure that legitimate users can still access it. You configure the web ACL to allow all requests except those from the IP addresses that you have specified.

When a request comes into AWS WAF, it checks against the list of rules that you have configured in the web ACL. If a request did not come from one of the blocked IP addresses, it allows access to the application.



However, if a request came from one of the blocked IP addresses that you have specified in the web ACL, it is denied access.

Amazon Inspector

Suppose that the developers at the coffee shop are developing and testing a new ordering application. They want to make sure that they are designing the application in accordance with security best practices. However, they have several other applications to develop, so they cannot spend much time conducting manual assessments. To perform automated security assessments, they decide to use **Amazon Inspector**.

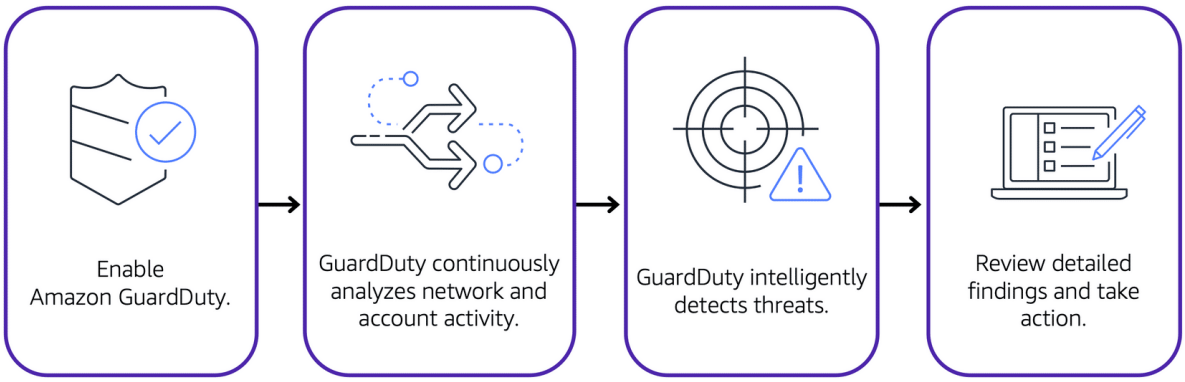
Amazon Inspector helps to improve the security and compliance of applications by running automated security assessments. It checks applications for security vulnerabilities and deviations from security best practices, such as open access to Amazon EC2 instances and installations of vulnerable software versions.

After Amazon Inspector has performed an assessment, it provides you with a list of security findings. The list prioritizes by severity level, including a detailed description of each security issue and a recommendation for how to fix it. However, AWS does not guarantee that following the provided recommendations resolves every potential security issue. Under

the shared responsibility model, customers are responsible for the security of their applications, processes, and tools that run on AWS services.

Amazon GuardDuty

Amazon GuardDuty is a service that provides intelligent threat detection for your AWS infrastructure and resources. It identifies threats by continuously monitoring the network activity and account behavior within your AWS environment.



After you have enabled GuardDuty for your AWS account, GuardDuty begins monitoring your network and account activity. You do not have to deploy or manage any additional security software. GuardDuty then continuously analyzes data from multiple AWS sources, including VPC Flow Logs and DNS logs.

If GuardDuty detects any threats, you can review detailed findings about them from the AWS Management Console. Findings include recommended steps for remediation. You can also configure AWS Lambda functions to take remediation steps automatically in response to GuardDuty's security findings.

Resources

To learn more about the concepts that were explored in Module 6, review these resources.

- Security, Identity, and Compliance on AWS
- Whitepaper: Introduction to AWS Security
- Whitepaper: Amazon Web Services - Overview of Security Processes
- AWS Security Blog
- AWS Compliance
- AWS Customer Stories: Security, Identity, and Compliance

Monitoring and Analytics

Amazon CloudWatch

Amazon CloudWatch

Amazon CloudWatch is a web service that enables you to monitor and manage various metrics and configure alarm actions based on data from those metrics.

CloudWatch uses **metrics** to represent the data points for your resources. AWS services send metrics to CloudWatch. CloudWatch then uses these metrics to create graphs automatically that show how performance has changed over time.

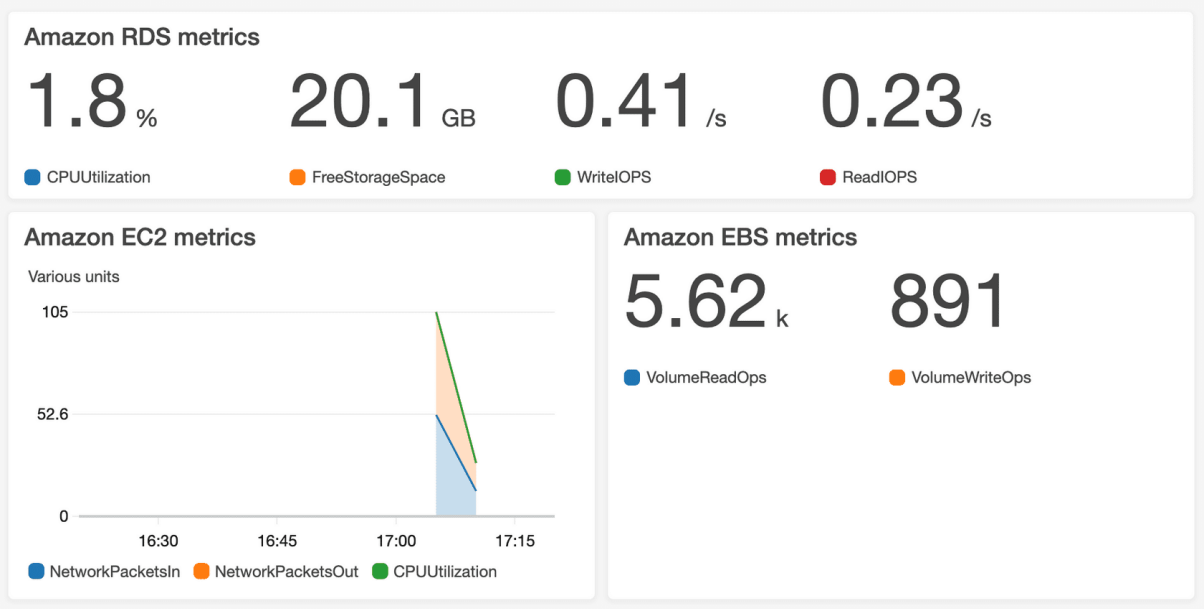
CloudWatch Alarms

With CloudWatch, you can create **alarms** that automatically perform actions if the value of your metric has gone above or below a predefined threshold.

For example, suppose that your company's developers use Amazon EC2 instances for application development or testing purposes. If the developers occasionally forget to stop the instances, the instances will continue to run and incur charges.

In this scenario, you could create a CloudWatch alarm that automatically stops an Amazon EC2 instance when the CPU utilization percentage has remained below a certain threshold for a specified period. When configuring the alarm, you can specify to receive a notification whenever this alarm is triggered.

CloudWatch Dashboard



The CloudWatch **dashboard** feature enables you to access all the metrics for your resources from a single location. For example, you can use a CloudWatch dashboard to monitor the CPU utilization of an Amazon EC2 instance, the total number of requests made to an Amazon S3 bucket, and more. You can even customize separate dashboards for different business purposes, applications, or resources.

AWS CloudTrail

AWS CloudTrail

AWS CloudTrail records API calls for your account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, and

more. You can think of CloudTrail as a “trail” of breadcrumbs (or a log of actions) that someone has left behind them.

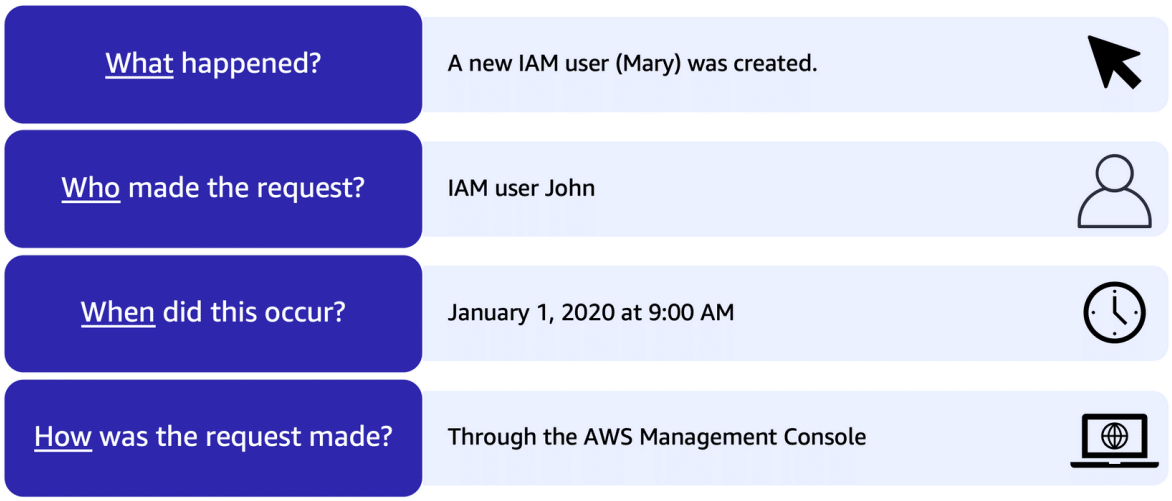
Recall that you can use API calls to provision, manage, and configure your AWS resources. With CloudTrail, you can view a complete history of user activity and API calls for your applications and resources.

Events are typically updated in CloudTrail within 15 minutes after an API call. You can filter events by specifying the time and date that an API call occurred, the user who requested the action, the type of resource that was involved in the API call, and more.

Example: AWS CloudTrail Event

Suppose that the coffee shop owner is browsing through the AWS Identity and Access Management (IAM) section of the AWS Management Console. They discover that a new IAM user named Mary was created, but they do not who, when, or which method created the user.

To answer these questions, the owner navigates to AWS CloudTrail.



In the CloudTrail Event History section, the owner applies a filter to display only the events for the “CreateUser” API action in IAM. The owner locates the event for the API call that created an IAM user for Mary. This event record provides complete details about what occurred:

On January 1, 2020 at 9:00 AM, IAM user John created a new IAM user (Mary) through the AWS Management Console.

CloudTrail Insights

Within CloudTrail, you can also enable **CloudTrail Insights**. This optional feature allows CloudTrail to automatically detect unusual API activities in your AWS account.

For example, CloudTrail Insights might detect that a higher number of Amazon EC2 instances than usual have recently launched in your account. You can then review the full event details to determine which actions you need to take next.

AWS Trusted Advisor

AWS Trusted Advisor

AWS Trusted Advisor is a web service that inspects your AWS environment and provides real-time recommendations in accordance with AWS best practices.

Trusted Advisor compares its findings to AWS best practices in five categories: cost optimization, performance, security, fault tolerance, and service limits. For the checks in each category, Trusted Advisor offers a list of recommended actions and additional resources to learn more about AWS best practices.

The guidance provided by AWS Trusted Advisor can benefit your company at all stages of deployment. For example, you can use AWS Trusted Advisor to assist you while you are creating new workflows and developing new applications. Or you can use it while you are making ongoing improvements to existing applications and resources.

AWS Trusted Advisor Dashboard

When you access the Trusted Advisor dashboard on the AWS Management Console, you can review completed checks for cost optimization, performance, security, fault tolerance, and service limits.

For each category:

- The green check indicates the number of items for which it detected **no problems**.
- The orange triangle represents the number of recommended **investigations**.
- The red circle represents the number of recommended **actions**.

Resources

To learn more about the concepts that were explored in Module 7, review these resources.

- Management and Governance on AWS
- Monitoring and Observability
- Configuration, Compliance, and Auditing
- AWS Management & Governance Blog
- Whitepaper: AWS Governance at Scale

Pricing and Support

AWS Free Tier

AWS Free Tier

The AWS Free Tier enables you to begin using certain services without having to worry about incurring costs for the specified period.

Three types of offers are available:

- Always Free
- 12 Months Free
- Trials

For each free tier offer, make sure to review the specific details about exactly which resource types are included.

Always Free

These offers do not expire and are available to all AWS customers.

For example, AWS Lambda allows 1 million free requests and up to 3.2 million seconds of compute time per month. Amazon DynamoDB allows 25 GB of free storage per month.

12 Months Free

These offers are free for 12 months following your initial sign-up date to AWS.

Examples include specific amounts of Amazon S3 Standard Storage, thresholds for monthly hours of Amazon EC2 compute time, and amounts of Amazon CloudFront data transfer out.

Trials

Short-term free trial offers start from the date you activate a particular service. The length of each trial might vary by number of days or the amount of usage in the service.

For example, Amazon Inspector offers a 90-day free trial. Amazon Lightsail (a service that enables you to run virtual private servers) offers 750 free hours of usage over a 30-day period.

AWS Pricing Concepts

How AWS Pricing Works

AWS offers a range of cloud computing services with pay-as-you-go pricing.

Pay for what you use.

For each service, you pay for exactly the amount of resources that you actually use, without requiring long-term contracts or complex licensing.

Pay less when you reserve.

Some services offer reservation options that provide a significant discount compared to On-Demand Instance pricing.

For example, suppose that your company is using Amazon EC2 instances for a workload that needs to run continuously. You might choose to run this workload on Amazon EC2 Instance Savings Plans, because the plan allows you to save up to 72% over the equivalent On-Demand Instance capacity.

Pay less with volume-based discounts when you use more.

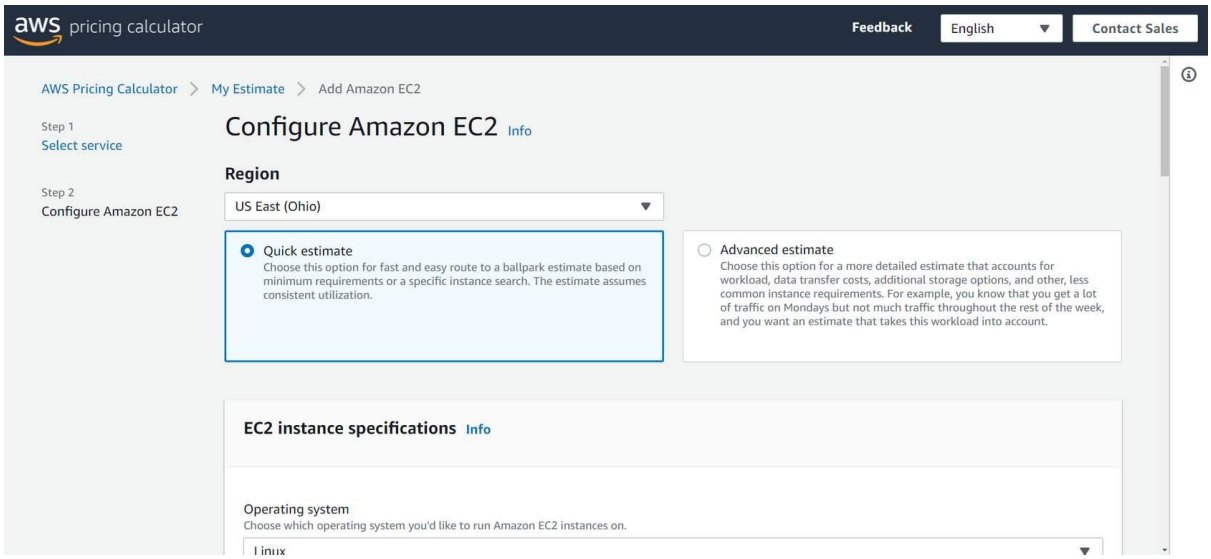
Some services offer tiered pricing, so the per-unit cost is incrementally lower with increased usage.

For example, the more Amazon S3 storage space you use, the less you pay for it per GB.

AWS Pricing Calculator

The **AWS Pricing Calculator** lets you explore AWS services and create an estimate for the cost of your use cases on AWS. You can organize your AWS estimates by groups that you define. A group can reflect how your company is organized, such as providing estimates by cost center.

When you have created an estimate, you can save it and generate a link to share it with others.



Suppose that your company is interested in using Amazon EC2. However, you are not yet sure which AWS Region or instance type would be the most cost-efficient for your use case. In the AWS Pricing Calculator, you can enter details such as the kind of operating system you need, memory requirements, and input/output (I/O) requirements. By using the AWS Pricing Calculator, you can review an estimated comparison of different EC2 instance types across AWS Regions.

AWS Pricing Examples

This section presents a few examples of pricing in AWS services.

AWS Lambda Pricing

For AWS Lambda, you are charged based on the number of requests for your functions and the time that it takes for them to run.

AWS Lambda allows 1 million free requests and up to 3.2 million seconds of compute time per month.

You can save on AWS Lambda costs by signing up for a Compute Savings Plan. A Compute Savings Plan offers lower compute costs in exchange for committing to a consistent amount of usage over a 1-year or 3-year term. This is an example of **paying less when you reserve**.

AWS Lambda Pricing Example

If you have used AWS Lambda in multiple AWS Regions, you can view the itemized charges by Region on your bill.

In this example, all the AWS Lambda usage occurred in the Northern Virginia Region. The bill lists separate charges for the number of requests for functions and their duration.

Both the number of requests and the total duration of requests in this example are under the thresholds in the AWS Free Tier, so the account owner would not have to pay for any AWS Lambda usage in this month.

Lambda		**$0.00**
US East (N. Virginia)		**$0.00**
AWS Lambda Lambda-GB-Second		$0.00
AWS Lambda - Compute Free Tier - 400,000 GB-Seconds - US East (Northern Virginia)	254.575 seconds	$0.00
AWS Lambda Request		$0.00
AWS Lambda - Requests Free Tier - 1,000,000 Requests - US East (Northern Virginia)	680.000 Requests	$0.00

Amazon EC2 Pricing

With Amazon EC2, you pay for only the compute time that you use while your instances are running.

For some workloads, you can significantly reduce Amazon EC2 costs by using Spot Instances. For example, suppose that you are running a batch processing job that is able to withstand interruptions. Using a Spot Instance would provide you with up to 90% cost savings while still meeting the availability requirements of your workload.

You can find additional cost savings for Amazon EC2 by considering Savings Plans and Reserved Instances.

Amazon EC2 Pricing Example

The service charges in this example include details for the following items:

- Each Amazon EC2 instance type that has been used
- The amount of Amazon EBS storage space that has been provisioned
- The length of time that Elastic Load Balancing has been used

In this example, all the usage amounts are under the thresholds in the AWS Free Tier, so the account owner would not have to pay for any Amazon EC2 usage in this month.

▾ **Elastic Compute Cloud**		**$0.00**
▾ **US East (N. Virginia)**		**$0.00**
Amazon Elastic Compute Cloud running Linux/UNIX		$0.00
$0.00 per Linux t2.micro instance-hour (or partial hour) under monthly free tier	106.512 Hrs	$0.00
EBS		$0.00
$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier	11.294 GB-Mo	$0.00
Elastic Load Balancing - Application		$0.00
$0.00 per Application LoadBalancer-hour (or partial hour) under monthly free tier	268.000 Hrs	$0.00

Amazon S3 Pricing

For Amazon S3 pricing, consider the following cost components:

- **Storage -** You pay for only the storage that you use. You are charged the rate to store objects in your Amazon S3 buckets based on your objects' sizes, storage classes, and how long you have stored each object during the month.
- **Requests and data retrievals -** You pay for requests made to your Amazon S3 objects and buckets. For example, suppose that you are storing photo files in Amazon S3 buckets and hosting them on a website. Every time a visitor requests the website that includes these photo files, this counts towards requests you must pay for.
- **Data transfer -** There is no cost to transfer data between different Amazon S3 buckets or from Amazon S3 to other services within the same AWS Region. However, you pay for data that you transfer into and out of Amazon S3, with a few exceptions. There is no cost for data transferred into Amazon S3 from the internet or out to Amazon CloudFront. There is also no cost for data transferred out to an Amazon EC2 instance in the same AWS Region as the Amazon S3 bucket.
- **Management and replication -** You pay for the storage management features that you have enabled on your account's Amazon S3 buckets. These features include Amazon S3 inventory, analytics, and object tagging.

Amazon S3 Pricing Example

The AWS account in this example has used Amazon S3 in two Regions: Northern Virginia and Ohio. For each Region, itemized charges are based on the following factors:

- The number of requests to add or copy objects into a bucket
- The number of requests to retrieve objects from a bucket
- The amount of storage space used

All the usage for Amazon S3 in this example is under the AWS Free Tier limits, so the account owner would not have to pay for any Amazon S3 usage in this month.

Simple Storage Service		**$0.00**
US East (N. Virginia)		**$0.00**
Amazon Simple Storage Service Requests-Tier1		$0.00
$0.00 per request - PUT, COPY, POST, or LIST requests under the monthly global free tier	185.000 Requests	$0.00
Amazon Simple Storage Service Requests-Tier2		$0.00
$0.00 per request - GET and all other requests under the monthly global free tier	923.000 Requests	$0.00
Amazon Simple Storage Service TimedStorage-ByteHrs		$0.00
$0.000 per GB - storage under the monthly global free tier	0.159 GB-Mo	$0.00
US East (Ohio)		**$0.00**
Amazon Simple Storage Service USE2-Requests-Tier2		$0.00
$0.00 per request - GET and all other requests under the monthly global free tier	4.000 Requests	$0.00
Amazon Simple Storage Service USE2-TimedStorage-ByteHrs		$0.00
$0.000 per GB - storage under the monthly global free tier	0.000001 GB-Mo	$0.00

Billing Dashboard

Use the **AWS Billing & Cost Management dashboard** to pay your AWS bill, monitor your usage, and analyze and control your costs.

- Compare your current month-to-date balance with the previous month, and get a forecast of the next month based on current usage.
- View month-to-date spend by service.
- View Free Tier usage by service.
- Access Cost Explorer and create budgets.
- Purchase and manage Savings Plans.
- Publish AWS Cost and Usage Reports.

Consolidated Billing

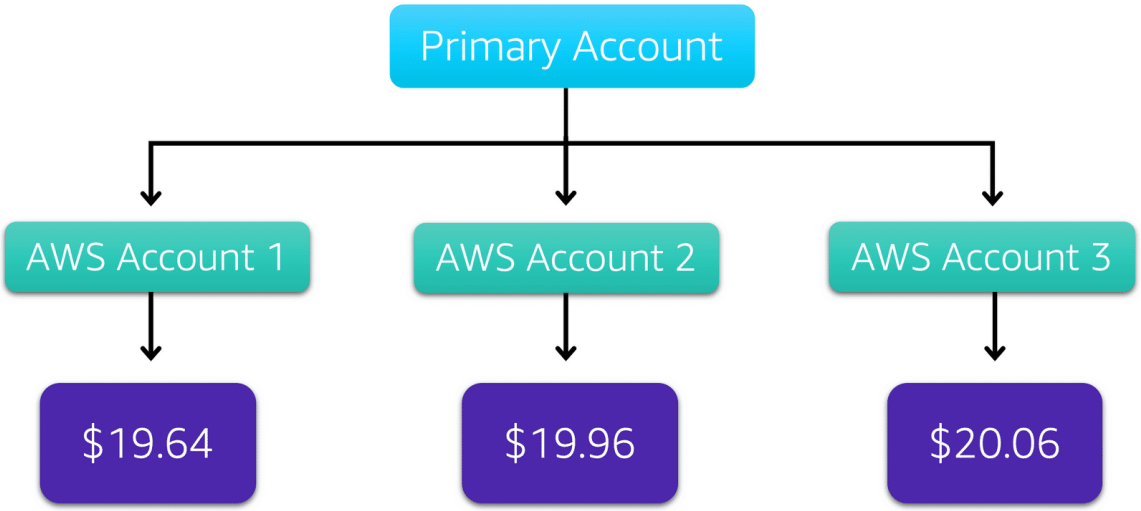
In an earlier module, you learned about AWS Organizations, a service that enables you to manage multiple AWS accounts from a central location. AWS Organizations also provides the option for **consolidated billing**.

The consolidated billing feature of AWS Organizations enables you to receive a single bill for all AWS accounts in your organization. By consolidating, you can easily track the combined costs of all the linked accounts in your organization. The default maximum number of accounts allowed for an organization is 4, but you can contact AWS Support to increase your quota, if needed.

On your monthly bill, you can review itemized charges incurred by each account. This enables you to have greater transparency into your organization's accounts while still maintaining the convenience of receiving a single monthly bill.

Another benefit of consolidated billing is the ability to share bulk discount pricing, Savings Plans, and Reserved Instances across the accounts in your organization. For instance, one account might not have enough monthly usage to qualify for discount pricing. However, when multiple accounts are combined, their aggregated usage may result in a benefit that applies across all accounts in the organization.

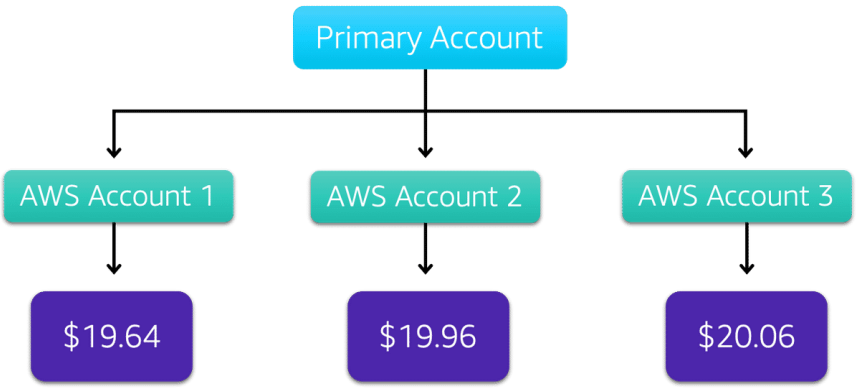
Example: Consolidated Billing



Suppose that you are the business leader who oversees your company's AWS billing.

Your company has three AWS accounts used for separate departments. Instead of paying each location's monthly bill separately, you decide to create an organization and add the three accounts.

You manage the organization through the primary account.

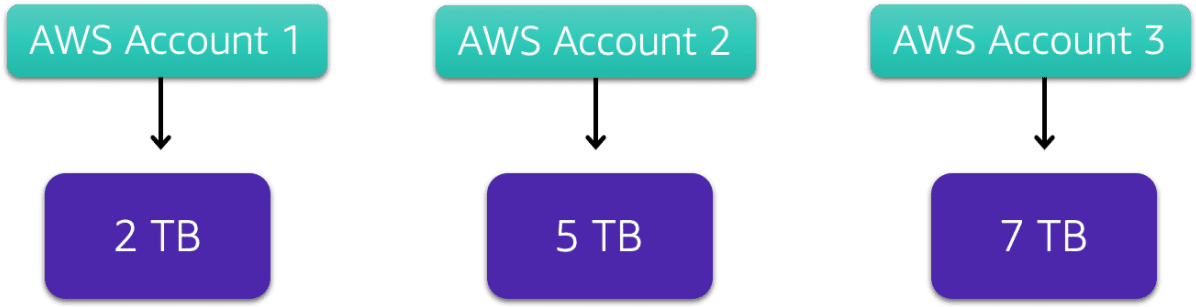


Monthly Consolidated Bill	
Primary Account	$14.14
AWS Account 1	$19.64
AWS Account 2	$19.96
AWS Account 3	$20.06
Total charged to paying account:	**$73.80**

Each month, AWS charges your primary payer account for all the linked accounts in a consolidated bill. Through the primary account, you can also get a detailed cost report for each linked account.

The monthly consolidated bill also includes the account usage costs incurred by the primary account. This cost is not a premium charge for having a primary account.

The consolidated bill shows the costs associated with any actions of the primary account (such as storing files in Amazon S3 or running Amazon EC2 instances).



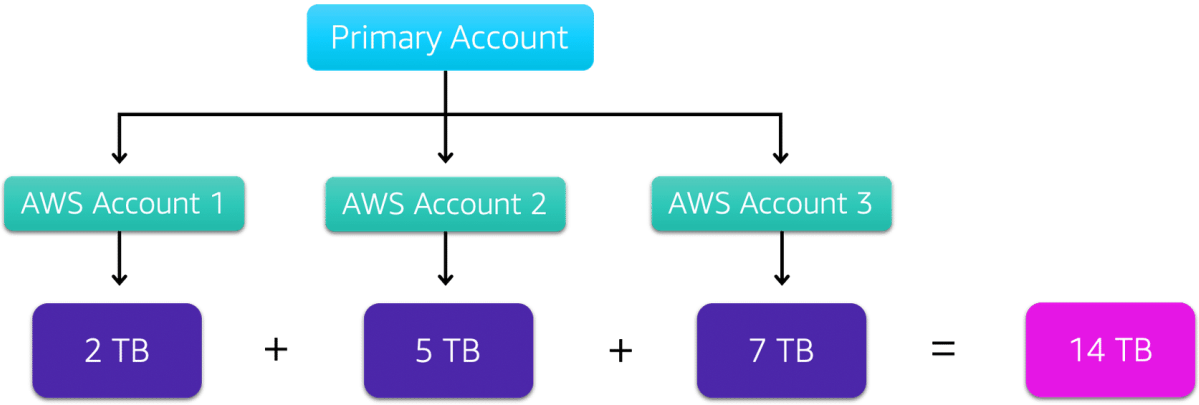
Consolidated billing also enables you to share volume pricing discounts across accounts.

Some AWS services, such as Amazon S3, provide volume pricing discounts that give you lower prices the more that you use the service. In Amazon S3, after customers have transferred 10 TB of data in a month, they pay a lower per-GB transfer price for the next 40 TB of data transferred.

In this example, there are three separate AWS accounts that have transferred different amounts of data in Amazon S3 during the current month:

- Account 1 has transferred 2 TB of data.
- Account 2 has transferred 5 TB of data.
- Account 3 has transferred 7 TB of data.

Because no single account has passed the 10 TB threshold, none of them is eligible for the lower per-GB transfer price for the next 40 TB of data transferred.



Now, suppose that these three separate accounts are brought together as linked accounts within a single AWS organization and are using consolidated billing.

When the Amazon S3 usage for the three linked accounts is combined (2+5+7), this results in a combined data transfer amount of 14 TB. This exceeds the 10-TB threshold.

With consolidated billing, AWS combines the usage from all accounts to determine which volume pricing tiers to apply, giving you a lower overall price whenever possible. AWS then allocates each linked account a portion of the overall volume discount based on the account's usage.

In this example, Account 3 would receive a greater portion of the overall volume discount because at 7 TB, it has transferred more data than Account 1 (at 2 TB) and Account 2 (at 5 TB).

AWS Budgets

AWS Budgets

In **AWS Budgets**, you can create budgets to plan your service usage, service costs, and instance reservations.

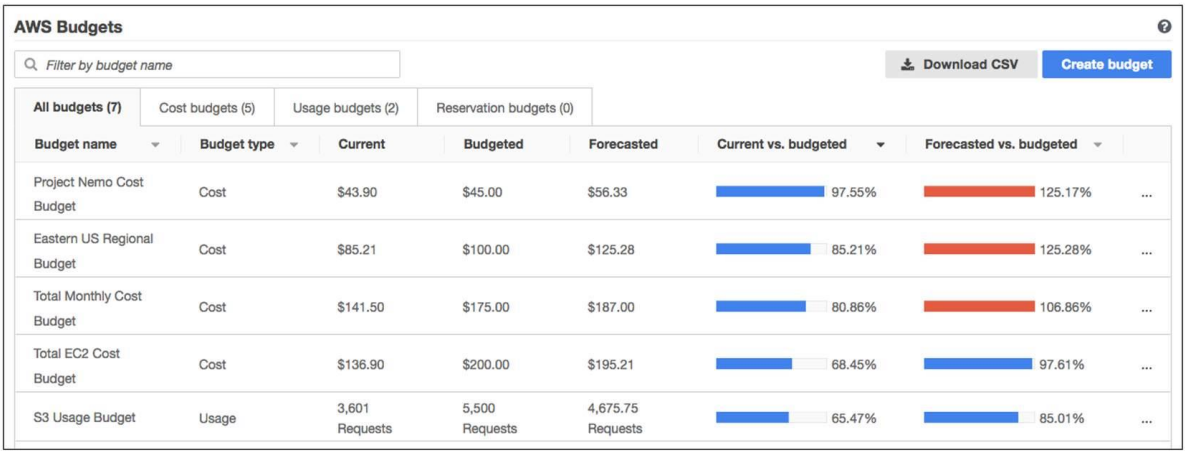
The information in AWS Budgets updates three times a day. This helps you to accurately determine how close your usage is to your budgeted amounts or to the AWS Free Tier limits.

In AWS Budgets, you can also set custom alerts when your usage exceeds (or is forecasted to exceed) the budgeted amount.

Example: AWS Budgets

Suppose that you have set a budget for Amazon EC2. You want to ensure that your company's usage of Amazon EC2 does not exceed $200 for the month.

In AWS Budgets, you could set a custom budget to notify you when your usage has reached half of this amount ($100). This setting would allow you to receive an alert and decide how you would like to proceed with your continued use of Amazon EC2.



In this sample budget, you can review the following important details:

- The current amount that you have incurred for Amazon EC2 so far this month ($136.90)
- The amount that you are forecasted to spend for the month ($195.21), based on your usage patterns.

You can also review comparisons of your current vs. budgeted usage, and forecasted vs. budgeted usage.

For example, in the top row of this sample budget, the forecasted vs. budgeted bar is at 125.17%. The reason for the increase is that the forecasted amount ($56.33) exceeds the amount that had been budgeted for that item for the month ($45.00).

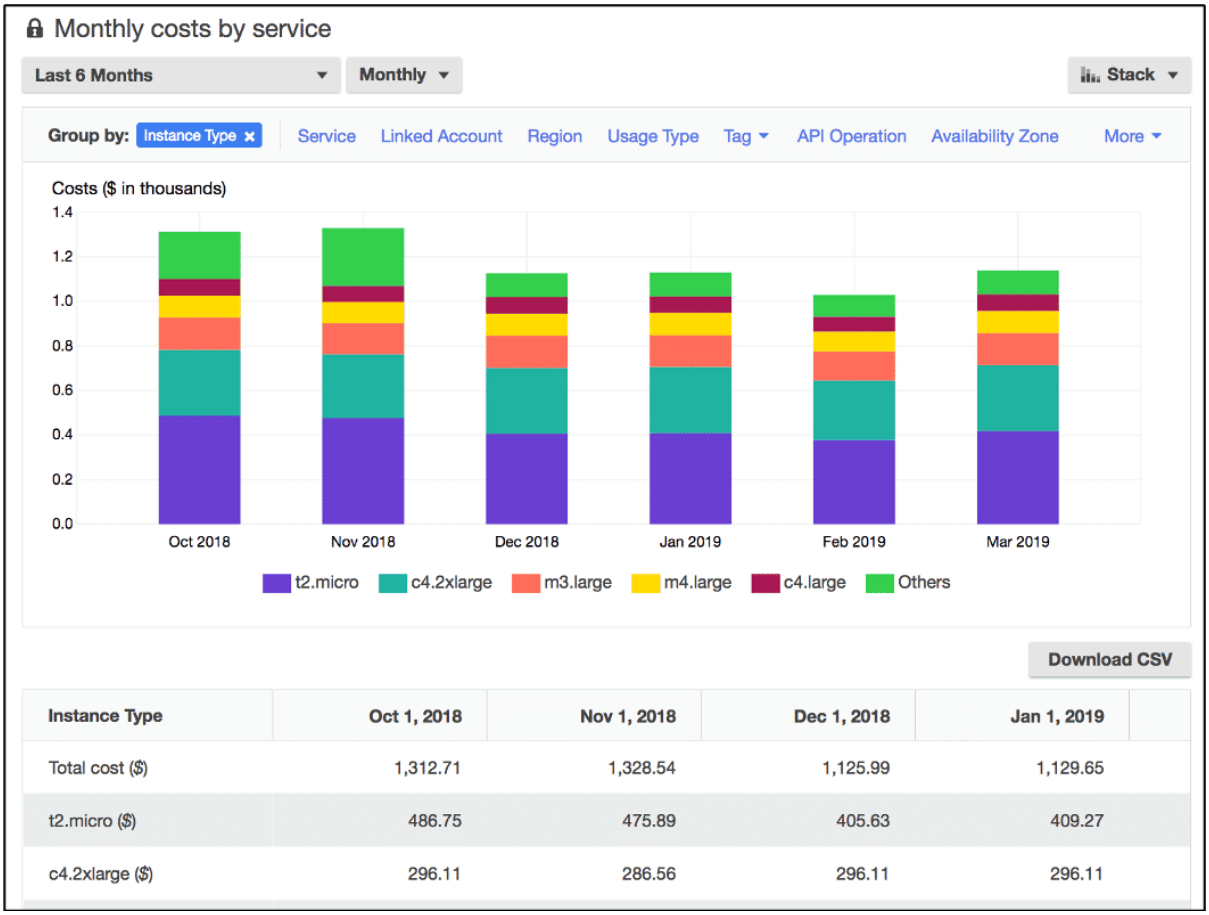
AWS Cost Explorer

AWS Cost Explorer

AWS Cost Explorer is a tool that enables you to visualize, understand, and manage your AWS costs and usage over time.

AWS Cost Explorer includes a default report of the costs and usage for your top five cost-accruing AWS services. You can apply custom filters and groups to analyze your data. For example, you can view resource usage at the hourly level.

Example: AWS Cost Explorer



Instance Type	Oct 1, 2018	Nov 1, 2018	Dec 1, 2018	Jan 1, 2019
Total cost ($)	1,312.71	1,328.54	1,125.99	1,129.65
t2.micro ($)	486.75	475.89	405.63	409.27
c4.2xlarge ($)	296.11	286.56	296.11	296.11

This example of the AWS Cost Explorer dashboard displays monthly costs for Amazon EC2 instances over a 6-month period. The bar for each month separates the costs for different Amazon EC2 instance types (such as t2.micro or m3.large).

By analyzing your AWS costs over time, you can make informed decisions about future costs and how to plan your budgets.

AWS Support Plans

AWS Support

AWS offers four different **Support plans** to help you troubleshoot issues, lower costs, and efficiently use AWS services.

You can choose from the following Support plans to meet your company's needs:

- Basic
- Developer
- Business
- Enterprise On-Ramp
- Enterprise

Basic Support

Basic Support is free for all AWS customers. It includes access to whitepapers, documentation, and support communities. With Basic Support, you can also contact AWS for billing questions and service limit increases.

With Basic Support, you have access to a limited selection of AWS Trusted Advisor checks. Additionally, you can use the **AWS Personal Health Dashboard**, a tool that provides alerts and remediation guidance when AWS is experiencing events that may affect you.

If your company needs support beyond the Basic level, you could consider purchasing Developer, Business, Enterprise On-Ramp, or Enterprise Support.

Developer, Business, Enterprise On-Ramp, and Enterprise Support

The Developer, Business, Enterprise On-Ramp, and Enterprise Support plans include all the benefits of Basic Support, in addition to the ability to open an unrestricted number of technical support cases. These three Support plans have pay-by-the-month pricing and require no long-term contracts.

The information in this course highlights only a selection of details for each Support plan. A complete overview of what is included in each Support plan, including pricing for each plan, is available on the AWS Support site.

In general, for pricing, the Developer plan has the lowest cost, the Business and Enterprise On-Ramp plans are in the middle, and the Enterprise plan has the highest cost.

Developer Support

Customers in the **Developer Support** plan have access to features such as:

- Best practice guidance
- Client-side diagnostic tools
- Building-block architecture support, which consists of guidance for how to use AWS offerings, features, and services together

For example, suppose that your company is exploring AWS services. You've heard about a few different AWS services. However, you're unsure of how to potentially use them together to build applications that can address your company's needs. In this scenario, the building-block architecture support that is included with the Developer Support plan could help you to identify opportunities for combining specific services and features.

Business Support

Customers with a **Business Support** plan have access to additional features, including:

- Use-case guidance to identify AWS offerings, features, and services that can best support your specific needs
- All AWS Trusted Advisor checks
- Limited support for third-party software, such as common operating systems and application stack components

Suppose that your company has the Business Support plan and wants to install a common third-party operating system onto your Amazon EC2 instances. You could contact AWS Support for assistance with installing, configuring, and troubleshooting the operating system. For advanced

topics such as optimizing performance, using custom scripts, or resolving security issues, you may need to contact the third-party software provider directly.

Enterprise On-Ramp Support

In November 2021, AWS opened enrollment into AWS Enterprise On-Ramp Support plan. In addition to all the features included in the Basic, Developer, and Business Support plans, customers with an Enterprise On-Ramp Support plan have access to:

- A pool of Technical Account Managers to provide proactive guidance and coordinate access to programs and AWS experts
- A Cost Optimization workshop (one per year)
- A Concierge support team for billing and account assistance
- Tools to monitor costs and performance through Trusted Advisor and Health API/Dashboard

Enterprise On-Ramp Support plan also provides access to a specific set of proactive support services, which are provided by a pool of Technical Account Managers.

- Consultative review and architecture guidance (one per year)
- Infrastructure Event Management support (one per year)
- Support automation workflows
- 30 minutes or less response time for business-critical issues

Enterprise Support

In addition to all features included in the Basic, Developer, Business, and Enterprise On-Ramp support plans, customers with Enterprise Support have access to:

- A designated Technical Account Manager to provide proactive guidance and coordinate access to programs and AWS experts
- A Concierge support team for billing and account assistance
- Operations Reviews and tools to monitor health
- Training and Game Days to drive innovation
- Tools to monitor costs and performance through Trusted Advisor and Health API/Dashboard

The Enterprise plan also provides full access to proactive services, which are provided by a designated Technical Account Manager:

- Consultative review and architecture guidance
- Infrastructure Event Management support
- Cost Optimization Workshop and tools
- Support automation workflows
- 15 minutes or less response time for business-critical issues

Technical Account Manager (TAM)

The Enterprise On-Ramp and Enterprise Support plans include access to a **Technical Account Manager (TAM)**. The TAM is your primary point of contact at AWS. If your company subscribes to Enterprise Support or Enterprise On-Ramp, your TAM educates, empowers, and evolves your cloud journey across the full range of AWS services. TAMs provide expert engineering guidance, help you design solutions that efficiently integrate AWS services, assist with cost-effective and resilient architectures, and provide direct access to AWS programs and a broad community of experts. For example, suppose that you are interested in developing an application that uses several AWS services together. Your TAM could provide insights into how to best use the

services together. They achieve this, while aligning with the specific needs that your company is hoping to address through the new application.

AWS Marketplace

AWS Marketplace

AWS Marketplace is a digital catalog that includes thousands of software listings from independent software vendors. You can use AWS Marketplace to find, test, and buy software that runs on AWS.

For each listing in AWS Marketplace, you can access detailed information on pricing options, available support, and reviews from other AWS customers.

You can also explore software solutions by industry and use case. For example, suppose that your company is in the healthcare industry. In AWS Marketplace, you can review use cases that software helps you to address, such as implementing solutions to protect patient records or using machine learning models to analyze a patient's medical history and predict possible health risks.

AWS Marketplace Categories

AWS Marketplace offers products in several categories, such as Infrastructure Software, DevOps, Data Products, Professional Services, Business Applications, Machine Learning, Industries, and Internet of Things (IoT).

Within each category, you can narrow your search by browsing through product listings in subcategories. For example, subcategories in the DevOps category include areas such as Application Development, Monitoring, and Testing.

Resources

To learn more about the concepts that were explored in Module 8, review these resources.

- AWS Pricing
- AWS Free Tier
- AWS Cost Management

- Whitepaper: How AWS Pricing Works
- Whitepaper: Introduction to AWS Economics
- AWS Support
- AWS Knowledge Center

Migration and Innovation

AWS Cloud Adoption Framework (AWS CAF)

Six core perspectives of the Cloud Adoption Framework

At the highest level, the **AWS Cloud Adoption Framework (AWS CAF)** organizes guidance into six areas of focus, called **Perspectives**. Each Perspective addresses distinct responsibilities. The planning process helps the right people across the organization prepare for the changes ahead.

In general, the **Business**, **People**, and **Governance** Perspectives focus on business capabilities, whereas the **Platform**, **Security**, and **Operations** Perspectives focus on technical capabilities.

Business Perspective

The **Business Perspective** ensures that IT aligns with business needs and that IT investments link to key business results.

Use the Business Perspective to create a strong business case for cloud adoption and prioritize cloud adoption initiatives. Ensure that your business strategies and goals align with your IT strategies and goals.

Common roles in the Business Perspective include:

- Business managers
- Finance managers
- Budget owners
- Strategy stakeholders

People Perspective

The **People Perspective** supports development of an organization-wide change management strategy for successful cloud adoption.

Use the People Perspective to evaluate organizational structures and roles, new skill and process requirements, and identify gaps. This helps prioritize training, staffing, and organizational changes.

Common roles in the People Perspective include:

- Human resources
- Staffing
- People managers

Governance Perspective

The **Governance Perspective** focuses on the skills and processes to align IT strategy with business strategy. This ensures that you maximize the business value and minimize risks.

Use the Governance Perspective to understand how to update the staff skills and processes necessary to ensure business governance in the cloud. Manage and measure cloud investments to evaluate business outcomes.

Common roles in the Governance Perspective include:

- Chief Information Officer (CIO)
- Program managers
- Enterprise architects
- Business analysts
- Portfolio managers

Platform Perspective

The **Platform Perspective** includes principles and patterns for implementing new solutions on the cloud, and migrating on-premises workloads to the cloud.

Use a variety of architectural models to understand and communicate the structure of IT systems and their relationships. Describe the architecture of the target state environment in detail.

Common roles in the Platform Perspective include:

- Chief Technology Officer (CTO)
- IT managers
- Solutions architects

Security Perspective

The **Security Perspective** ensures that the organization meets security objectives for visibility, auditability, control, and agility.

Use the AWS CAF to structure the selection and implementation of security controls that meet the organization’s needs.

Common roles in the Security Perspective include:

- Chief Information Security Officer (CISO)

- IT security managers
- IT security analysts

Operations Perspective

The **Operations Perspective** helps you to enable, run, use, operate, and recover IT workloads to the level agreed upon with your business stakeholders.

Define how day-to-day, quarter-to-quarter, and year-to-year business is conducted. Align with and support the operations of the business. The AWS CAF helps these stakeholders define current operating procedures and identify the process changes and training needed to implement successful cloud adoption.

Common roles in the Operations Perspective include:

- IT operations managers
- IT support managers

Migration Strategies

6 Strategies for Migration

When migrating applications to the cloud, six of the most common migration strategies that you can implement are:

- Rehosting
- Replatforming
- Refactoring/re-architecting
- Repurchasing
- Retaining
- Retiring

Rehosting

Rehosting also known as “lift-and-shift” involves moving applications without changes.

In the scenario of a large legacy migration, in which the company is looking to implement its migration and scale quickly to meet a business case, the majority of applications are rehosted.

Replatforming

Replatforming, also known as “lift, tinker, and shift,” involves making a few cloud optimizations to realize a tangible benefit. Optimization is achieved without changing the core architecture of the application.

Refactoring/re-architecting

Refactoring (also known as **re-architecting**) involves reimagining how an application is architected and developed by using cloud-native features. Refactoring is driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application's existing environment.

Repurchasing

Repurchasing involves moving from a traditional license to a software-as-a-service model.

For example, a business might choose to implement the repurchasing strategy by migrating from a customer relationship management (CRM) system to Salesforce.com.

Retaining

Retaining consists of keeping applications that are critical for the business in the source environment. This might include applications that require major refactoring before they can be migrated, or, work that can be postponed until a later time.

Retiring

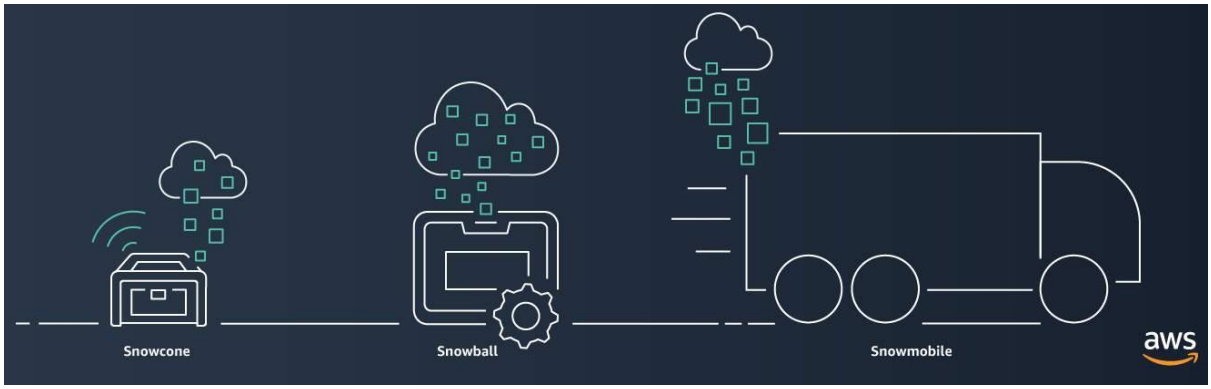
Retiring is the process of removing applications that are no longer needed.

AWS Snow Family

AWS Snow Family Members

The **AWS Snow Family** is a collection of physical devices that help to physically transport up to exabytes of data into and out of AWS.

AWS Snow Family is composed of **AWS Snowcone**, **AWS Snowball**, and **AWS Snowmobile**.



These devices offer different capacity points, and most include built-in computing capabilities. AWS owns and manages the Snow Family devices and integrates with AWS security, monitoring, storage management, and computing capabilities.

AWS Snowcone

AWS Snowcone is a small, rugged, and secure edge computing and data transfer device.

It features 2 CPUs, 4 GB of memory, and 8 TB of usable storage.

AWS Snowball

AWS Snowball offers two types of devices:

- **Snowball Edge Storage Optimized** devices are well suited for large-scale data migrations and recurring transfer workflows, in addition to local computing with higher capacity needs. Snowball Edge Storage Optimized provides 80 TB of HDD capacity for block volumes and Amazon S3-compatible object storage, and 1 TB of SATA SSD for block volumes.
- **Snowball Edge Compute Optimized** provides powerful computing resources for use cases such as machine learning, full motion video analysis, analytics, and local computing stacks.

AWS Snowmobile

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

You can transfer up to 100 petabytes of data per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi trailer truck.

Innovate with AWS

When examining how to use AWS services, it is important to focus on the desired outcomes. You are properly equipped to drive innovation in the cloud if you can clearly articulate the following conditions:

- The current state
- The desired state
- The problems you are trying to solve

Consider some of the paths you might explore in the future as you continue on your cloud journey.

Serverless Applications

With AWS, **serverless** refers to applications that don't require you to provision, maintain, or administer servers. You don't need to worry about fault tolerance or availability. AWS handles these capabilities for you.

AWS Lambda is an example of a service that you can use to run serverless applications. If you design your architecture to trigger Lambda functions to run your code, you can bypass the need to manage a fleet of servers.

Building your architecture with serverless applications enables your developers to focus on their core product instead of managing and operating servers.

Artificial Intelligence

AWS offers a variety of services powered by **artificial intelligence (AI)**.

For example, you can perform the following tasks:

- Convert speech to text with Amazon Transcribe.

- Discover patterns in text with Amazon Comprehend.
- Identify potentially fraudulent online activities with Amazon Fraud Detector.
- Build voice and text chatbots with Amazon Lex.

Machine Learning

Traditional **machine learning (ML)** development is complex, expensive, time consuming, and error prone. AWS offers Amazon SageMaker to remove the difficult work from the process and empower you to build, train, and deploy ML models quickly.

You can use ML to analyze data, solve complex problems, and predict outcomes before they happen.

Resources

To learn more about the concepts that were explored in Module 9, review these resources.

- Migration & Transfer on AWS
- A Process for Mass Migrations to the Cloud
- 6 Strategies for Migrating Applications to the Cloud
- AWS Cloud Adoption Framework
- AWS Fundamentals: Core Concepts
- AWS Cloud Enterprise Strategy Blog
- Modernizing with AWS Blog
- AWS Customer Stories: Data Center Migration

The Cloud Journey

The AWS Well-Architected Framework

The **AWS Well-Architected Framework** helps you understand how to design and operate reliable, secure, efficient, and cost-effective systems in the AWS Cloud. It provides a way for you to consistently measure your architecture against best practices and design principles and identify areas for improvement.

The Well-Architected Framework is based on six pillars:

- Operational excellence
- Security
- Reliability
- Performance efficiency
- Cost optimization
- Sustainability

Operational Excellence

Operational excellence is the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

Design principles for operational excellence in the cloud include performing operations as code, annotating documentation, anticipating failure, and frequently making small, reversible changes.

Security

The **Security** pillar is the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

When considering the security of your architecture, apply these best practices:

- Automate security best practices when possible.
- Apply security at all layers.
- Protect data in transit and at rest.

Reliability

Reliability is the ability of a system to do the following:

- Recover from infrastructure or service disruptions
- Dynamically acquire computing resources to meet demand

- Mitigate disruptions such as misconfigurations or transient network issues

Reliability includes testing recovery procedures, scaling horizontally to increase aggregate system availability, and automatically recovering from failure.

Performance Efficiency

Performance efficiency is the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

Evaluating the performance efficiency of your architecture includes experimenting more often, using serverless architectures, and designing systems to be able to go global in minutes.

Cost Optimization

Cost optimization is the ability to run systems to deliver business value at the lowest price point.

Cost optimization includes adopting a consumption model, analyzing and attributing expenditure, and using managed services to reduce the cost of ownership.

Sustainability

In December 2021, AWS introduced a sustainability pillar as part of the AWS Well-Architected Framework.

Sustainability is the ability to continually improve sustainability impacts by reducing energy consumption and increasing efficiency across all components of a workload by maximizing the benefits from the provisioned resources and minimizing the total resources required.

To facilitate good design for sustainability:

- Understand your impact
- Establish sustainability goals
- Maximize utilization
- Anticipate and adopt new, more efficient hardware and software offerings
- Use managed services
- Reduce the downstream impact of your cloud workloads

Benefits of the AWS Cloud

Operating in the AWS Cloud offers many benefits over computing in on-premises or hybrid environments.

In this section, you will learn about six advantages of cloud computing:

- Trade upfront expense for variable expense.
- Benefit from massive economies of scale.
- Stop guessing capacity.
- Increase speed and agility.
- Stop spending money running and maintaining data centers.
- Go global in minutes.

Trade upfront expense for variable expense.

Upfront expenses include data centers, physical servers, and other resources that you would need to invest in before using computing resources.

Instead of investing heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources.

Benefit from massive economies of scale.

By using cloud computing, you can achieve a lower variable cost than you can get on your own.

Because usage from hundreds of thousands of customers aggregates in the cloud, providers such as AWS can achieve higher economies of scale. Economies of scale translate into lower pay-as-you-go prices.

Stop guessing capacity.

With cloud computing, you don't have to predict how much infrastructure capacity you will need before deploying an application.

For example, you can launch Amazon Elastic Compute Cloud (Amazon EC2) instances when needed and pay only for the compute time you use. Instead of paying for resources that are unused or dealing with limited capacity, you can access only the capacity that you need, and scale in or out in response to demand.

Increase speed and agility.

The flexibility of cloud computing makes it easier for you to develop and deploy applications.

This flexibility also provides your development teams with more time to experiment and innovate.

Stop spending money running and maintaining data centers.

Cloud computing in data centers often requires you to spend more money and time managing infrastructure and servers.

A benefit of cloud computing is the ability to focus less on these tasks and more on your applications and customers.

Go global in minutes.

The AWS Cloud global footprint enables you to quickly deploy applications to customers around the world, while providing them with low latency.

Resources

The six pillars of the AWS Well-Architected Framework:

- Operational excellence
- Security
- Reliability
- Performance efficiency
- Cost optimization
- Sustainability

Six advantages of cloud computing:

- Trade upfront expense for variable expense.
- Benefit from massive economies of scale.
- Stop guessing capacity.
- Increase speed and agility.
- Stop spending money running and maintaining data centers.
- Go global in minutes.

Additional resources

To learn more about the concepts that were explored in Module 10, review these resources.

- AWS Well-Architecte
- AWS Well-Architected Framework
- AWS Architecture Center
- Six Advantages of Cloud Computing
- AWS Architecture Blog

AWS Certified Cloud Practitioner Basics

Exam Details

Exam Domains

The AWS Certified Cloud Practitioner exam includes four domains:

1. Cloud Concepts
2. Security and Compliance
3. Technology
4. Billing and Pricing

The areas covered describe each domain in the Exam Guide for the AWS Certified Cloud Practitioner certification. For a description of each domain, review the AWS Certified Cloud Practitioner website. You are encouraged to read the information in the Exam Guide as part of your preparation for the exam.

Each domain in the exam is weighted. The weight represents the percentage of questions in the exam that correspond to that particular domain. These are approximations, so the questions on your exam may not match these percentages exactly. The exam does not indicate the domain associated with a question. In fact, some questions can potentially fall under multiple domains.

Percentage of Exam

Domain	**Percentage of Exam**
Domain 1: Cloud Concepts	26%
Domain 2: Security and Compliance	25%
Domain 3: Technology	33%
Domain 4: Billing and Pricing	16%

Domain	Percentage of Exam
Total	100%

You are encouraged to use these benchmarks to help you determine how to allocate your time studying for the exam.

Recommended Experience

Candidates for the AWS Certified Cloud Practitioner exam should have a basic understanding of IT services and their uses in the AWS Cloud platform.

We recommend that you have at least six months of experience with the AWS Cloud in any role, including project managers, IT managers, sales managers, decision makers, and marketers. These roles are in addition to those working in finance, procurement, and legal departments.

Exam Details

The AWS Certified Cloud Practitioner exam consists of **65 questions** to be completed in **90 minutes**. The minimum passing score is **70%**.

Two types of questions are included on the exam: multiple choice and multiple response.

- A **multiple-choice** question has one correct response and three incorrect responses, or distractors.
- A **multiple-response** question has two or more correct responses out of five or more options.

On the exam, there is no penalty for guessing. Any questions that you do not answer are scored as incorrect. If you are not sure of what the correct answer is, it's always best for you to guess instead of leaving any questions unanswered.

The exam enables you to flag any questions that you'd like to review before submitting the exam. This helps you to use your time during the exam efficiently, knowing that you can always go back and review any questions that you were initially unsure of.

Whitepapers and Resources

As part of your preparation for the AWS Certified Cloud Practitioner exam, we recommend that you review the following whitepapers and resources:

- Overview of Amazon Web Services
- How AWS Pricing Works
- Compare AWS Support Plans