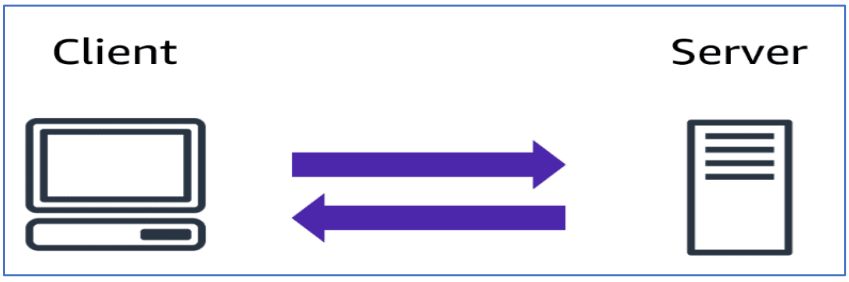
Introduction to AWS Cloud Practitioner Essentials

Course overview: AWS Cloud concepts, AWS services, security, architecture, pricing, and support

Module 1: Introduction to Amazon Web Services

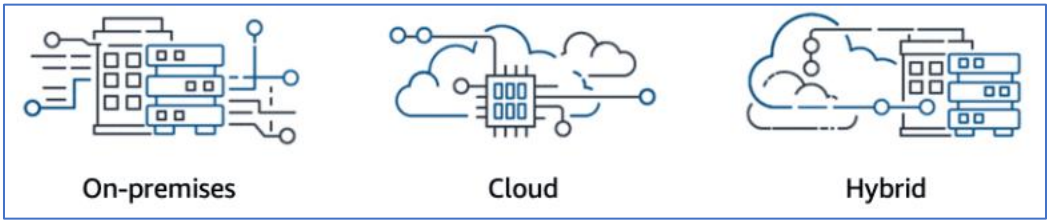
- **Client- Server model:**



- 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.

Cloud Computing:

- ***Selecting a cloud strategy***: 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.



1. 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.
- You can build them using higher-level services that reduce the management, architecting, and scaling requirements of the core infrastructure.

2. On-Premises deployments (*private cloud* deployment):

- Companies and organizations hosted and maintained hardware such as compute, storage, and Networking equipment in their own data centers.
- Deploy resources by using virtualization and resource management tools.
- Increase resource utilization by using application management and virtualization technologies.

3. Hybrid-Deployment:

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

- 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.

Six advantages of cloud computing

1. **Pay-as-you-go**
2. **Benefit from massive economies of scale**
3. **Stop guessing capacity**
4. **Increase speed and agility**
5. **Realize cost savings**
6. **Go global in minutes**

Benefits of cloud computing:

1. **Trade upfront expense for variable expense**: Upfront expense sure 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.
2. **Stop spending money to run and maintain data cent**ers: 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.
3. **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.
4. **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.
5. **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.
6. **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.

Amazon Elastic Compute Cloud (Amazon EC2)

➔ Provides secure, resizable compute capacity in the cloud as Amazon EC2 instances.
➔ With traditional on-premises resources, you must do the following:

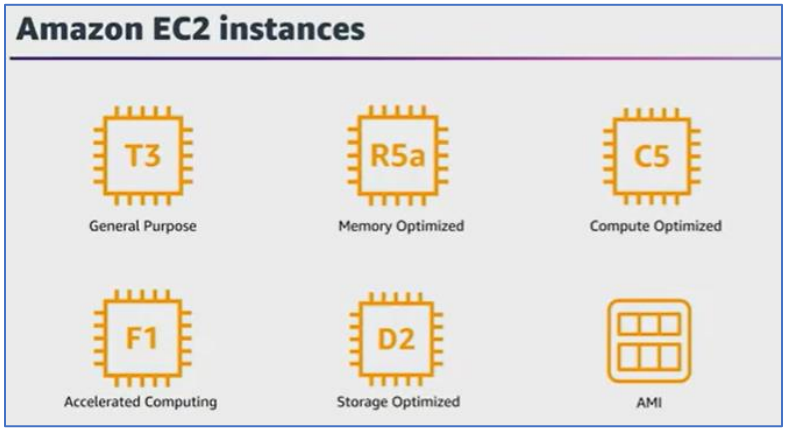
- Spend money upfront to purchase hardware.
- Wait for the servers to be delivered to you.
- Install the servers in your physical data center.
- Make all the necessary configurations.

➔ By comparison, with an Amazon EC2 instance:

- You can 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 – Connect -Use**

Amazon EC2 Instance Types:



1. **General purpose instances (T3):** 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

2. **Compute optimized instances (C5)**: These 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 computing 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.

3. **Memory optimized instances (R5a):** 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 can run, it is loaded from storage into memory. This preloading process gives the CPU direct access to the computer program.
   -> Memory optimized instances enable you to run workloads with high memory needs and receive great performance.

4. **Accelerated computing instances (F1)**: 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.

5. **Storage optimized instances (D2)** 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: 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.

1. **On-Demand:** 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.

2. **Reserved instances**: are a billing discount applied to the use of On-Demand Instances in your account. There are two available types of Reserved Instances:

- Standard Reserved Instances
- Convertible Reserved Instances

You can purchase Standard Reserved and Convertible Reserved Instances for a 1-year or 3-year term. You realize greater cost savings with the 3-year option.

Standard Reserved Instances: This option is a good fit if you know the EC2 instance type and size you need for your steady-state applications and in which AWS Region you plan to run them. Reserved Instances require you to state the following qualifications:

- **Instance type and size:** For example, m5.xlarge
- **Platform description (operating system):** For example, Microsoft Windows Server or Red Hat Enterprise Linux
- **Tenancy:** Default tenancy or dedicated tenancy

You have the option to specify an Availability Zone for your EC2 Reserved Instances. If you make this specification, you get EC2 capacity reservation. This ensures that your desired amount of EC2 instances will be available when you need them.

Convertible Reserved Instances: If you need to run your EC2 instances in different Availability Zones or different instance types, then Convertible Reserved Instances might be right for you. Note: You trade in a deeper discount when you require flexibility to run your EC2 instances.

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 family and size, Region, platform, and tenancy).

3. **EC2 Instance saving plans:** AWS offers Savings Plans for a few compute services, including Amazon EC2. EC2 Instance Savings Plans reduce your EC2 instance costs when you make an hourly spend commitment to an instance family and Region for a 1-year or 3-year term. This term commitment results in savings of up to 72 percent compared to On-Demand rates. Any usage up to the commitment is charged at the discounted Savings Plans rate (for example, $10 per hour). Any usage beyond the commitment is charged at regular On-Demand rates.

➔ The EC2 Instance Savings Plans are a good option if you need flexibility in your Amazon EC2 usage over the duration of the commitment term. You have the benefit of saving costs on running any EC2 instance within an EC2 instance family in a chosen Region (for example, M5 usage in N. Virginia) regardless of Availability Zone, instance size, OS, or tenancy. The savings with EC2 Instance Savings Plans are like the savings provided by Standard Reserved Instances.

➔ Unlike Reserved Instances, however, you don't need to specify up front what EC2 instance type and size (for example, m5. xlarge), OS, and tenancy to get a discount. Further, you don't need to commit to a certain number of EC2 instances over a 1-year or 3-year term. Additionally, the EC2 Instance Savings Plans don't include an EC2 capacity reservation option.

- ➔ Later in this course, you'll review AWS Cost Explorer, which you can use to visualize, understand, and manage your AWS costs and usage over time. If you're considering your options for Savings Plans, you can use AWS Cost Explorer to 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.

4. **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.

5. **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:

1. **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.
2. **Amazon EC2 Auto Scaling**: 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 can maintain a greater sense of application availability.
   -> 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.

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

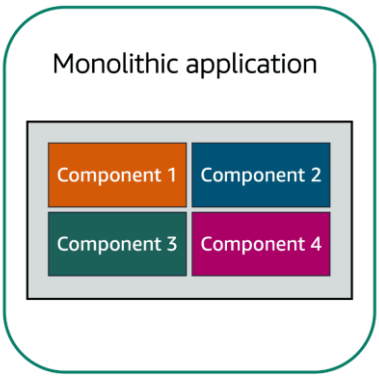
- ***Dynamic scaling*** **responds to changing demand.**
- ***Predictive scaling*** **automatically schedules the right number of Amazon EC2 instances based on predicted demand.**

Directing Traffic with Elastic Load Balancing

1. 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.

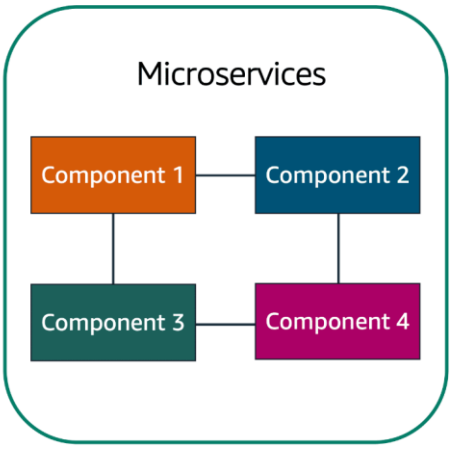
-> Although Elastic Load Balancing and Amazon EC2 Auto Scaling are separate services, they work together to help ensure that applications running in Amazon EC2 can provide high performance and availability.

Messaging and Queuing:



- ➔ 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.

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) 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.

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.

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:

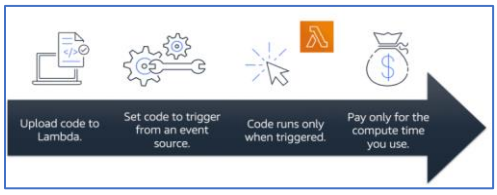
1. Provision instances (virtual servers).
2. Upload your code.
3. Continue to manage the instances while your application is running.

Comparison between computing with virtual servers (thinking about servers and code) and serverless computing (thinking only about code).

- ➔ 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**.

1. AWS Lambda: (Function as a service, Accept functions, Event-Driven and Serverless)

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.

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.

➔ Container orchestration services help you to deploy, manage, and scale your containerized applications. Next, you will learn about two services that provide container orchestration: **Amazon Elastic Container Service and Amazon Elastic Kubernetes Service**.

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.

Amazon Elastic Kubernetes Service (Amazon EKS)

Amazon Elastic Kubernetes Service (Amazon EKS) is a fully managed service that you can use to run Kubernetes on AWS.

Kubernetes is open-source software that enables you to deploy and manage containerized applications at scale. A large community of volunteers maintains Kubernetes, and AWS actively works together with the Kubernetes community. As new features and functionalities release for Kubernetes applications, you can easily apply these updates to your applications managed by Amazon EKS.

2. AWS Fargate

AWS Fargate is a serverless compute engine for containers. It works with both Amazon ECS and Amazon EKS.

When using AWS Fargate, you do not need to provision or manage servers. AWS Fargate manages your server infrastructure for you. You can focus more on innovating and developing your applications, and you pay only for the resources that are required to run your containers.

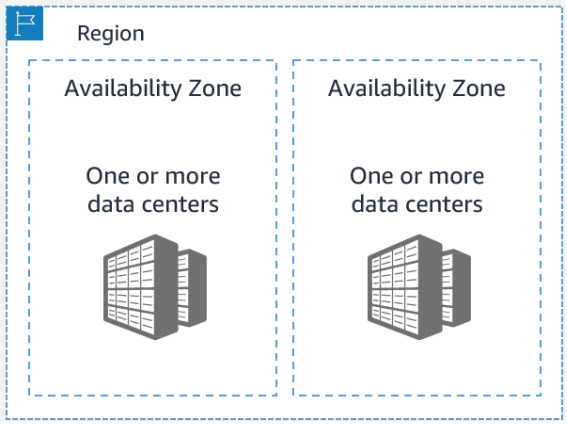
Additional resources

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
- Amazon EC2 Reserved Instances
- How Savings Plans apply to usage

Module3: AWS Global Infrastructure

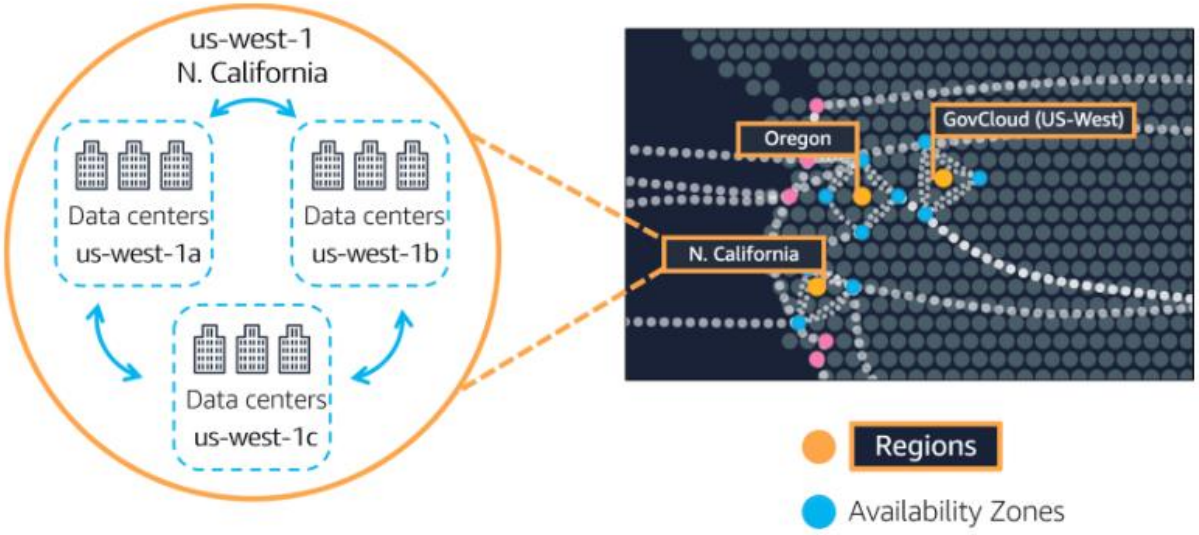
Regions



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

1. **Compliance with data governance and legal requirements (**Data Compliance**)**
2. **Proximity to your customers (**Latency: delay between a request for data and the response**)**
3. **Available services within a Region (**Service Availability**)**
4. **Pricing**

Availability Zones



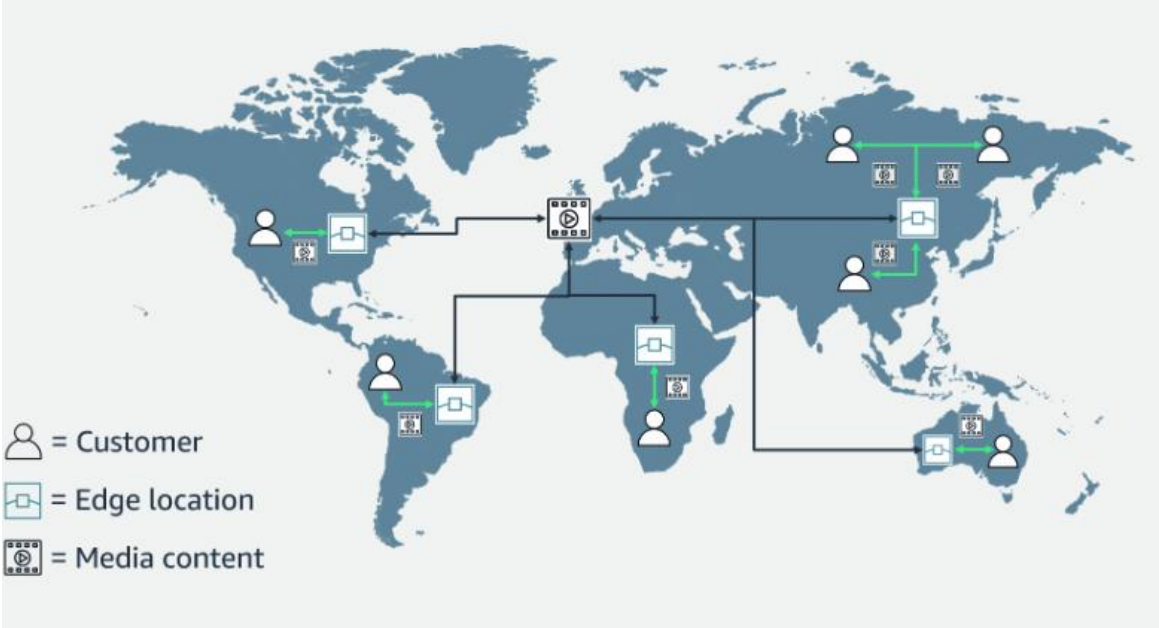
Spotlight on the us-west-1 Region. Northern California, Oregon, and GovCloud (US-West) are separate Regions. The Northern California Region is called us-west-1, and this Region contains three AZs (1a, 1b, and 1c). Then, within each AZ there are three data centers.

➔ 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.

➔ **Planning for failure and deploying applications across multiple Availability Zones is an important part of building a resilient and highly available architecture**.

Edge Locations: Amazon CloudFront

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



1. **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.

2. **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.

3. **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.

How to Provision AWS Resources/Ways to interact with AWS services:

1. **AWS Management Console**: It is a web-based interface for accessing and managing AWS services.

2. **AWS Command Line Interface (AWS CLI):** 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.

3. **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

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. It determines the right operations to perform when managing your stack and rolls back changes automatically if it detects errors.

AWS Outposts: AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources such as EC2 instances, EBS volumes, ECS clusters, and RDS instances. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.

Additional 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

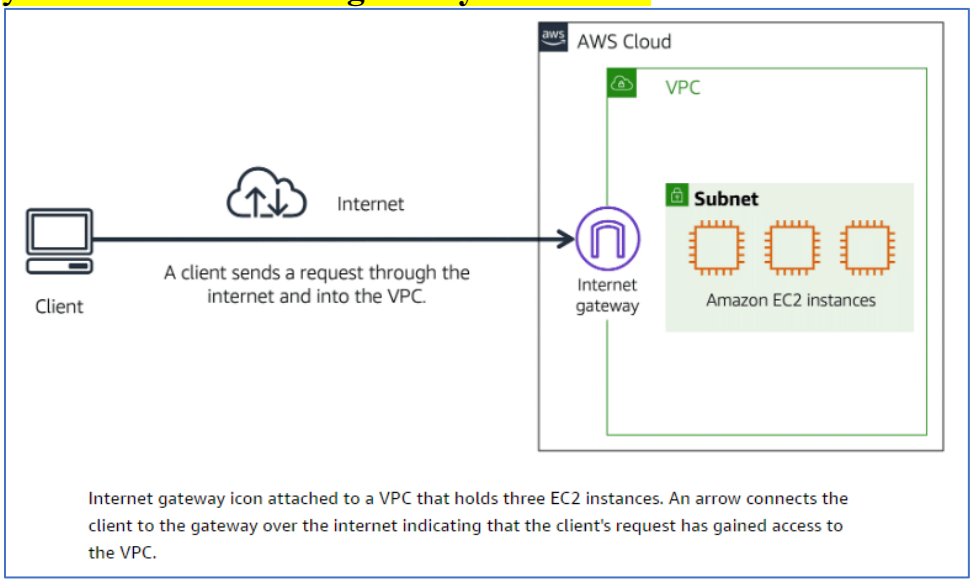
Module4: NETWORKING: Connectivity to AWS

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

- ➔ 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.
- ➔ **VPC and subnets:**

Name	Default	Adjustable
VPCs per Region	5	Yes
Subnets per VPC	200	Yes
IPv4 CIDR blocks per VPC	5	Yes (**up to 50**)
IPv6 CIDR blocks per VPC	5	Yes (up to 50)

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



Internet gateway icon attached to a VPC that holds three EC2 instances. An arrow connects the client to the gateway over the internet indicating that the client's request has gained access 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.

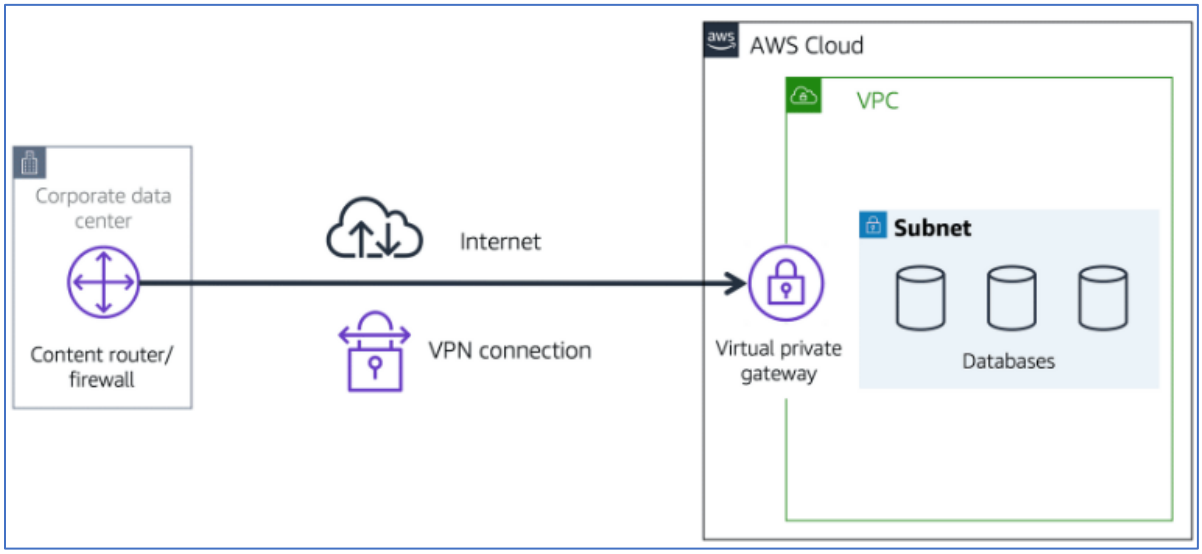
➔ What if you have a VPC that includes only private resources?

➔ **Virtual private gateway:** To access private resources in a VPC, you can use a **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.



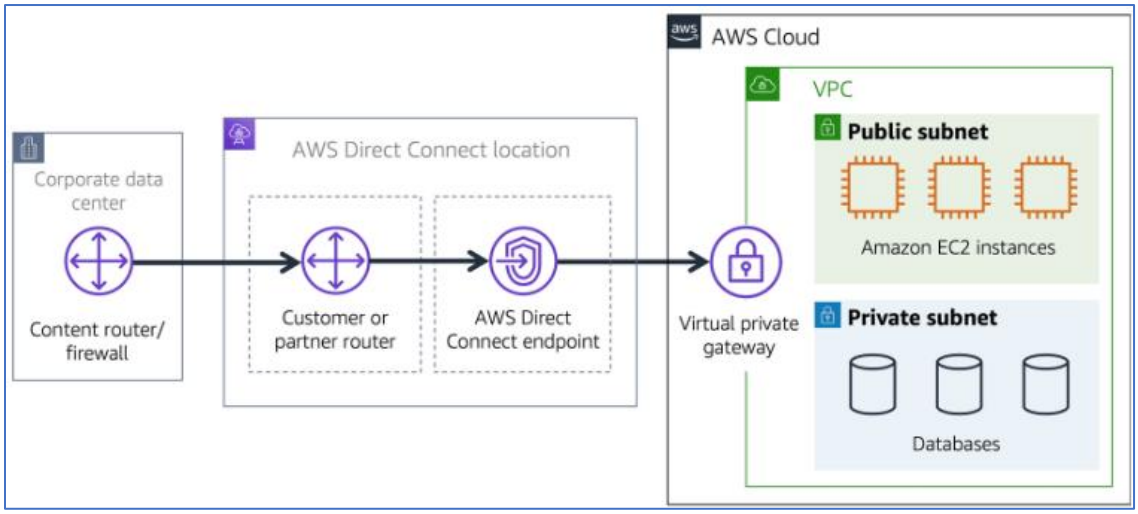
➔ A virtual private gateway enables you to establish a virtual private Network (VPN) connection between your VPC and a private Network, such as an on-premises data center or internal corporate Network.

➔ A virtual private gateway allows traffic into the VPC only if it is coming from an approved Network.

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

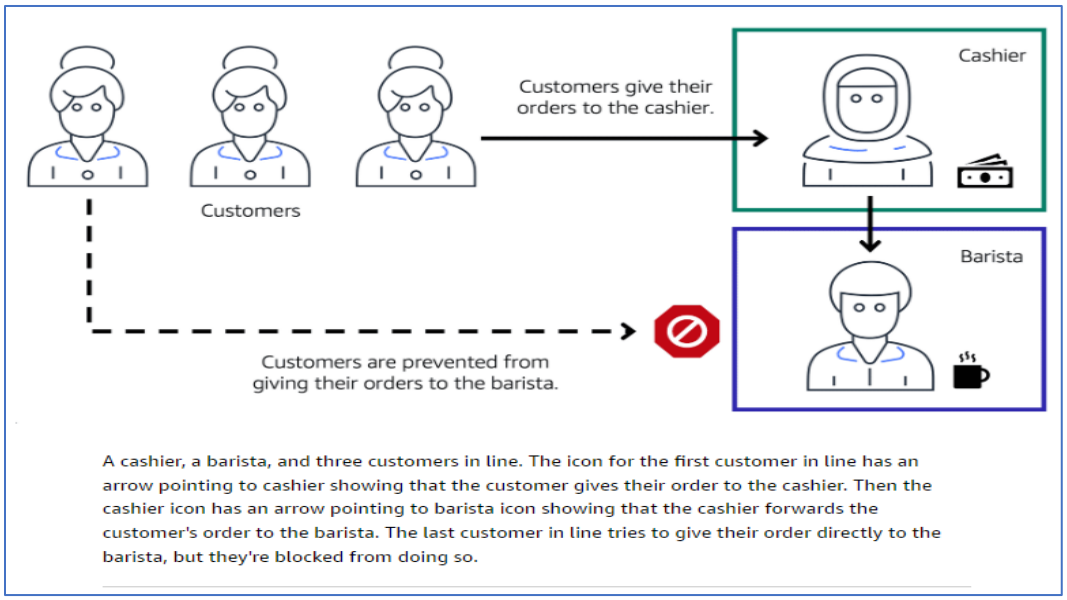
➔ Suppose that there is an apartment building with a hallway directly linking the building to the coffee shop. Only the residents of the apartment building can travel through this hallway.

➔ 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.

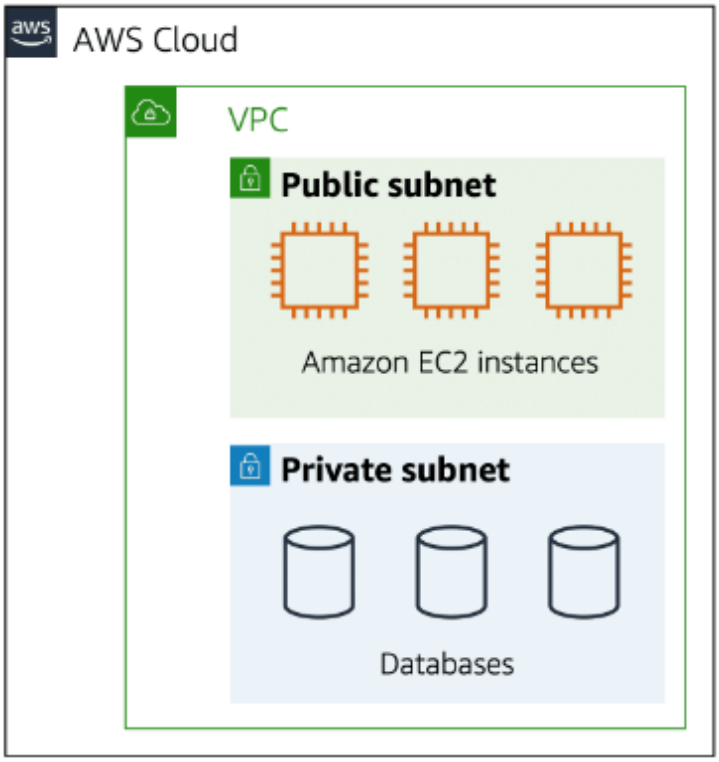


➔ The private connection that AWS Direct Connect provides helps you to reduce Network costs and increase the amount of bandwidth that can travel through your Network.

Subnets and Network Access Control Lists:



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 in a private subnet.

1. VPC Peering

- VPC Peering is a connection between two VPCs that allows them to route traffic between each other using private IP addresses. Peering works across different regions (inter-region peering) as well as within the same region (intra-region peering).
- Use Case: Suitable for connecting two VPCs in the same or different AWS accounts to share resources, like EC2 instances, databases, etc.
- Key Features:
    - Traffic between the VPCs remains private, and it doesn't traverse the internet.
    - You must update the route tables in each VPC to allow communication.
    - Peering connections are one-to-one (i.e., two VPCs can only peer with each other, but they cannot transitively connect to other VPCs through the peering connection).
- Limitations:
    - Cannot overlap CIDR blocks between VPCs.
    - No transitive peering (one VPC can't route traffic through another VPC).

Example:

- VPC A and VPC B are connected directly via VPC peering. Each VPC must have a route in its route table to the other VPC's CIDR block via the peering connection.

2. AWS Transit Gateway

- AWS Transit Gateway is a highly scalable service that allows you to interconnect multiple VPCs and on-premises networks through a central hub

(the Transit Gateway). It acts as a hub to facilitate communication between all attached VPCs and even your on-premises networks.

- Use Case: Ideal for connecting multiple VPCs in large architectures or multi-account environments where you want simplified and scalable management of inter-VPC traffic.
- Key Features:
  - Allows for transitive routing (one VPC can route traffic to another VPC through the Transit Gateway).
  - Supports inter-region peering, allowing communication between VPCs in different regions.
  - Simplifies network architecture by consolidating VPC connections through a single service.
  - You can also connect on-premises networks via Direct Connect or VPN to the Transit Gateway.
- Limitations:
  - More complex to set up than VPC peering, especially for small-scale architectures.
  - Additional cost compared to VPC peering.

Example:

- VPC A, VPC B, and VPC C are all attached to a Transit Gateway, and VPC A can communicate with VPC B or VPC C through the central Transit Gateway.

VPC Endpoints: VPC Endpoints are a crucial AWS networking feature that allows you to connect your VPC to AWS services and other VPCs securely, without traversing the public internet. VPC Endpoints provide private, secure access to AWS services from within a VPC, ensuring that traffic between your VPC and the services remains on the AWS internal network.
There are two types of VPC endpoints:

1. Interface Endpoints (AWS PrivateLink)
2. Gateway Endpoints

1. Interface VPC Endpoints (AWS PrivateLink)
-> Interface Endpoints provide private connectivity between your VPC and supported AWS services or third-party services hosted in VPCs using AWS PrivateLink.
-- Private Connectivity: Traffic to the endpoint is routed through private IPs within the VPC, ensuring that data does not traverse the internet.

-- Supports: AWS services like S3, DynamoDB, SNS, EC2, and Custom services hosted in VPCs via AWS PrivateLink.
-- Mechanism: The interface endpoint uses an ENI (Elastic Network Interface) in your VPC to provide connectivity to the service.
Use Case:
You can use Interface Endpoints to connect to services like S3, DynamoDB, or any other AWS service that supports private connectivity (like services inside your VPC or third-party services). This ensures that traffic remains private and does not leave AWS's internal network.

Example:

- Service: S3
- Endpoint Type: Interface VPC Endpoint (PrivateLink)
- How it works: When an EC2 instance in a private subnet accesses an S3 bucket via the Interface Endpoint, the traffic is routed privately using AWS's internal network without going through the public internet.

Steps to Create Interface Endpoint:

1. In the AWS Console, go to **VPC Dashboard**.
2. Under **Endpoints**, click **Create Endpoint**.
3. Choose **Interface** as the endpoint type.
4. Select the AWS service you want to connect to (for example, S3, DynamoDB, etc.).
5. Choose the **VPC** and **subnet** where you want to create the endpoint.
6. Configure **security groups** to control access to the endpoint.
7. Click **Create Endpoint**.

2. Gateway VPC Endpoints
-> Gateway VPC Endpoints allow you to connect your VPC to certain AWS services, such as Amazon S3 and DynamoDB, without using public IPs or a NAT device. These endpoints use gateway-type route tables and allow private connectivity to the specified services.

- Private Connectivity: The traffic from your VPC to the service is routed over the AWS backbone, without needing a NAT Gateway, VPN, or public IP addresses.
- Supports: Only for Amazon S3 and DynamoDB.
- Mechanism: The gateway endpoint allows you to update the route table to send traffic to the AWS service using the endpoint instead of routing it through the internet or NAT.

Use Case:

- You can use Gateway Endpoints for S3 or DynamoDB to ensure secure and private access to these services without needing to use public IPs.

Example:

- Service: Amazon S3
- Endpoint Type: Gateway VPC Endpoint
- How it works: When an EC2 instance in your VPC accesses an S3 bucket, the traffic is routed directly to S3 over the AWS internal network without traversing the public internet.

Steps to Create Gateway Endpoint:

1. Go to the VPC Dashboard in the AWS Console.
2. Under Endpoints, click Create Endpoint.
3. Select Gateway as the endpoint type.
4. Choose the AWS service (S3 or DynamoDB).
5. Choose the VPC and select the route table(s) to update for the traffic to route through the endpoint.
6. Click Create Endpoint.

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 a VPC through an internet gateway. Before a packet can enter 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 ACLs: A Network ACL is a virtual firewall that controls inbound and outbound traffic at the subnet level.

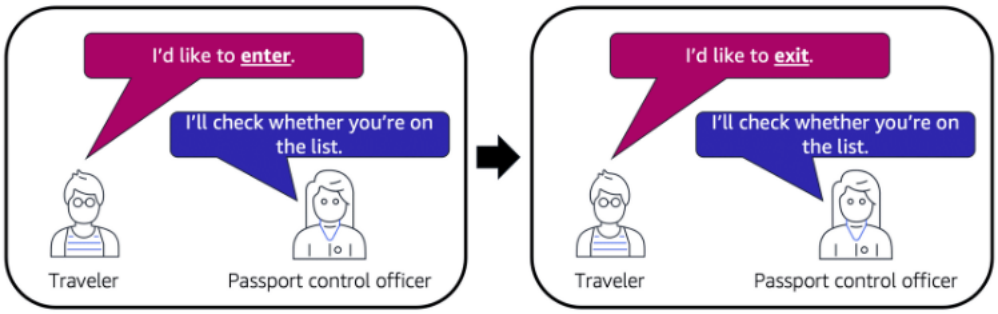
Each AWS account includes a default Network ACL. When configuring your VPC, you can use your account's default Network ACL or create custom Network ACLs.

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.

NACL 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.

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.

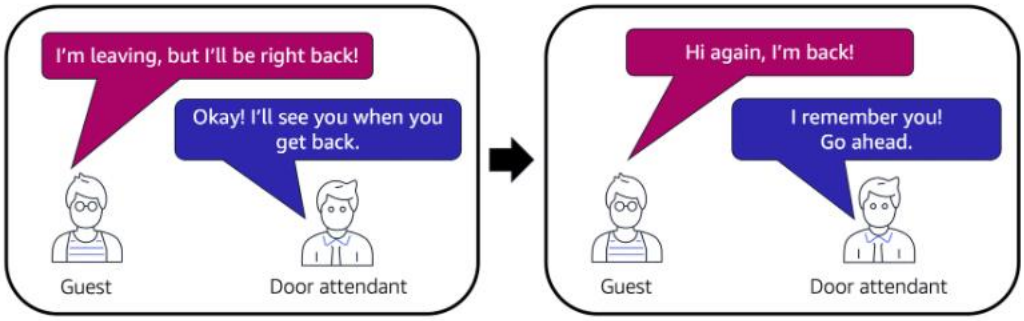
The VPC component that checks packet permissions for an Amazon EC2 instance is a **security group**.

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

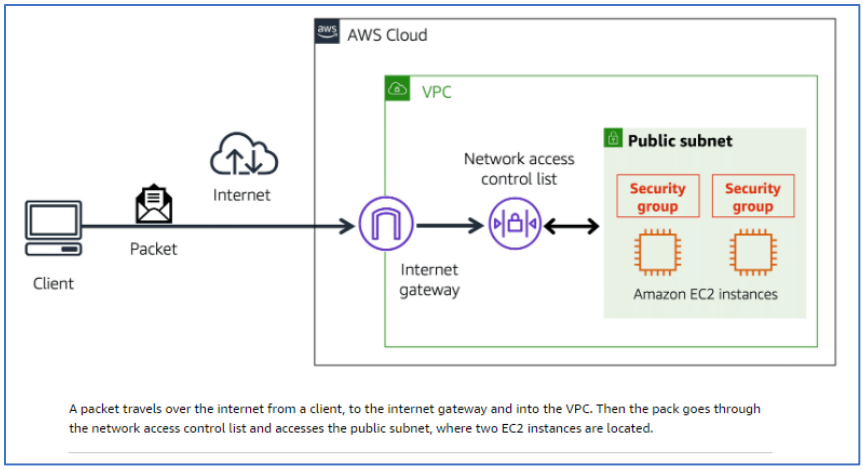
- ➔ By default, a security group denies all inbound traffic and allows all outbound traffic. You can add custom rules to configure which traffic should be allowed; any other traffic would then be denied.
- ➔ If you have multiple Amazon EC2 instances within the same VPC, you can associate them with the 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.
- ➔ 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.



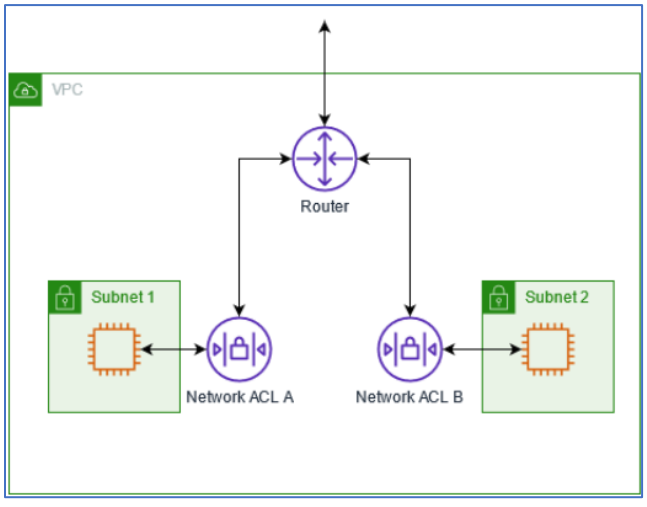
With both Network ACLs and security groups, you can configure custom rules for the traffic in your VPC.



Control traffic to subnets using Network ACLs: A Network access control list (NACL) allows or denies specific inbound or outbound traffic at the subnet level. You can use the default Network ACL for your VPC, or you can create a custom Network ACL for your VPC with rules that are like the rules for your security groups in order to add an additional layer of security to your VPC.

There is no additional charge for using Network ACLs.

The following diagram shows a VPC with two subnets. Each subnet has a Network ACL. When traffic enters the VPC (for example, from a peered VPC, VPN connection, or the internet), the router sends the traffic to its destination. Network ACL A determines which traffic destined for subnet 1 is allowed to enter subnet 1, and which traffic destined for a location outside subnet 1 is allowed to leave subnet 1. Similarly, Network ACL B determines which traffic is allowed to enter and leave subnet 2.



Network ACL basics

The following are the basic things that you need to know about Network ACLs:

- Your VPC automatically comes with a modifiable default Network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
- You can create a custom Network ACL and associate it with a subnet to allow or deny specific inbound or outbound traffic at the subnet level.
- Each subnet in your VPC must be associated with a Network ACL. If you don't explicitly associate a subnet with a Network ACL, the subnet is automatically associated with the default Network ACL.
- You can associate a Network ACL with multiple subnets. However, a subnet can be associated with only one Network ACL at a time. When you associate a Network ACL with a subnet, the previous association is removed.
- A Network ACL has inbound rules and outbound rules. Each rule can either allow or deny traffic. Each **rule has a number from 1 to 32766**. We evaluate the rules in order, starting with the lowest numbered rule, when deciding whether allow or deny traffic. If the traffic matches a rule, the rule is applied, and we do not evaluate any additional rules. We recommend that you start by creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules later on, if needed.
- We evaluate the Network ACL rules when traffic enters and leaves the subnet, not as it is routed within a subnet.
- **NACLs are *stateless***, which means that information about previously sent or received traffic is not saved. If, for example, you create a NACL rule to allow specific inbound traffic to a subnet, responses to that traffic are not automatically allowed. This is in contrast to how security groups work. **Security groups are *stateful*,** which means that information about previously sent or received traffic is saved. If, for example, a security group allows inbound traffic to an EC2 instance, responses are automatically allowed regardless of outbound security group rules.
- **Network ACLs can't block DNS requests to or from the Route 53 Resolver** (also known as the VPC+2 IP address or AmazonProvidedDNS). To filter DNS requests through the Route 53 Resolver, you can

enable Route 53 Resolver DNS Firewall in the *Amazon Route 53 Developer Guide*.

- Network ACLs can't block traffic to the Instance Metadata Service (IMDS). To manage access to IMDS, see Configure the instance metadata options in the *Amazon EC2 User Guide*.
- Network ACLs do not filter traffic destined to and from the following:
  - Amazon Domain Name Services (DNS)
  - Amazon Dynamic Host Configuration Protocol (DHCP)
  - Amazon EC2 instance metadata
  - Amazon ECS task metadata endpoints
  - License activation for Windows instances
  - Amazon Time Sync Service
  - Reserved IP addresses used by the default VPC router
- There are quotas (also known as limits) for the number of Network ACLs per VPC and the number of rules per Network ACL. For more information, see Amazon VPC quotas.

Network ACL rules

You can add or remove rules from the default Network ACL or create additional Network ACLs for your VPC. When you add or remove rules from a Network ACL, the changes are automatically applied to the subnets that it's associated with.

The following are the parts of a Network ACL rule:

- **Rule number**. Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that might contradict it.
- **Type**. The type of traffic; for example, SSH. You can also specify all traffic or a custom range.
- **Protocol**. You can specify any protocol that has a standard protocol number. For more information, see Protocol Numbers. If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.
- **Port range**. The listening port or port range for the traffic. For example, 80 for HTTP traffic.
- **Source**. [Inbound rules only] The source of the traffic (CIDR range).

- **Destination**. [Outbound rules only] The destination for the traffic (CIDR range).
- **Allow/Deny**. Whether to *allow* or *deny* the specified traffic.

If you add a rule using a command line tool or the Amazon EC2 API, the CIDR range is automatically modified to its canonical form. For example, if you specify 100.68.0.18/18 for the CIDR range, we create a rule with a 100.68.0.0/18 CIDR range.

Default Network ACL:

The default Network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated. Each Network ACL also includes a rule whose rule number is an asterisk (*). This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

The following is an example default Network ACL for a VPC that supports IPv4 only.

Inbound					
Rule #	Type	Protocol	Port range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	Type	Protocol	Port range	Destination	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

If you create a VPC with an IPv6 CIDR block or if you associate an IPv6 CIDR block with your existing VPC, we automatically add rules that allow all IPv6 traffic to flow in and out of your subnet. We also add rules whose rule numbers are an asterisk that ensures that a packet is denied if it doesn't match any of the other numbered rules. You can't modify or remove these rules. The following is an example default Network ACL for a VPC that supports IPv4 and IPv6.

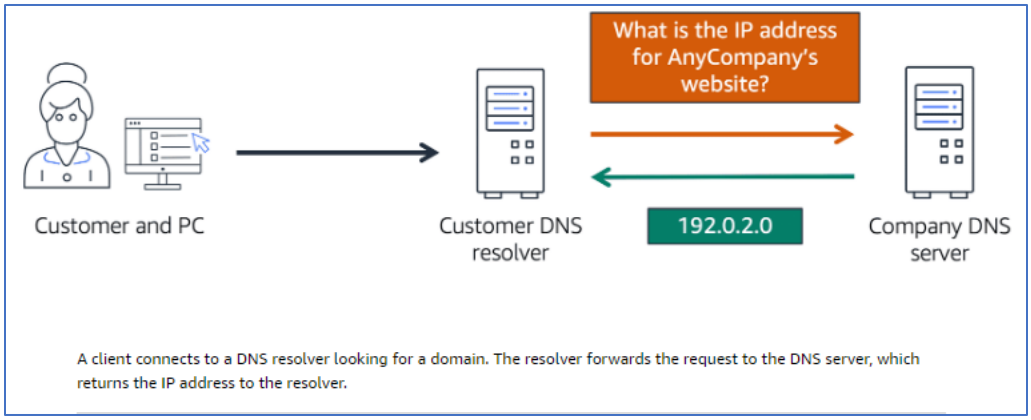
Inbound					
Rule #	Type	Protocol	Port range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
101	All IPv6 traffic	All	All	::/0	ALLOW
*	All traffic	All	All	0.0.0.0/0	DENY
*	All IPv6 traffic	All	All	::/0	DENY
Outbound					
Rule #	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	ALLOW
101	All IPv6 traffic	All	All	::/0	ALLOW
*	All traffic	All	All	0.0.0.0/0	DENY
*	All IPv6 traffic	All	All	::/0	DENY

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 customer DNS resolver communicating with a company DNS 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.



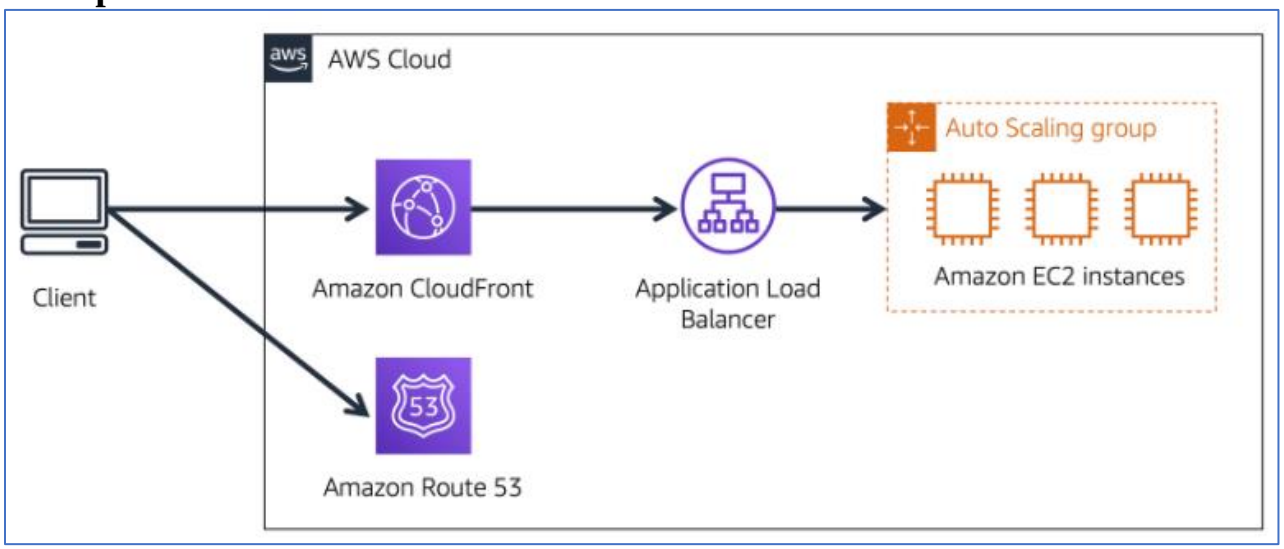
For example, suppose that you want to visit AnyCompany's website.

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

Amazon Route 53 (DNS, Register domains and host zones, Routing policies)

- **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.

- A customer requests data from the application by going to AnyCompany's website.
- 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.
- The customer's request is sent to the nearest edge location through Amazon CloudFront.
- Amazon CloudFront connects to the Application Load Balancer, which sends the incoming packet to an Amazon EC2 instance.

Additional resources: Networking and Content Delivery on AWS, AWS Networking & Content Delivery Blog, Amazon Virtual Private Cloud, What is Amazon VPC?, How Amazon VPC works

Module 5 Introduction: Storage and Databases

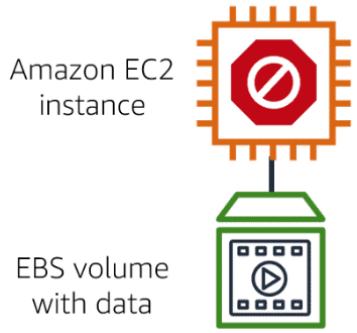
Learning objectives

- Summarize the basic concept of storage and databases.
- Describe the benefits of Amazon Elastic Block Store (Amazon EBS).
- Describe the benefits of Amazon Simple Storage Service (Amazon S3).
- Describe the benefits of Amazon Elastic File System (Amazon EFS).
- Summarize various storage solutions.
- Describe the benefits of Amazon Relational Database Service (Amazon RDS).
- Describe the benefits of Amazon DynamoDB.
- Summarize various database services.

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.
- ➔ Amazon EC2 instances are virtual servers. If you start an instance from a stopped state, the instance might start on another host, where the previously used instance store volume does not exist. Therefore, AWS recommends instance stores for use cases that involve temporary data that you do not need in the long term.

Amazon Elastic Block Store (Amazon EBS)

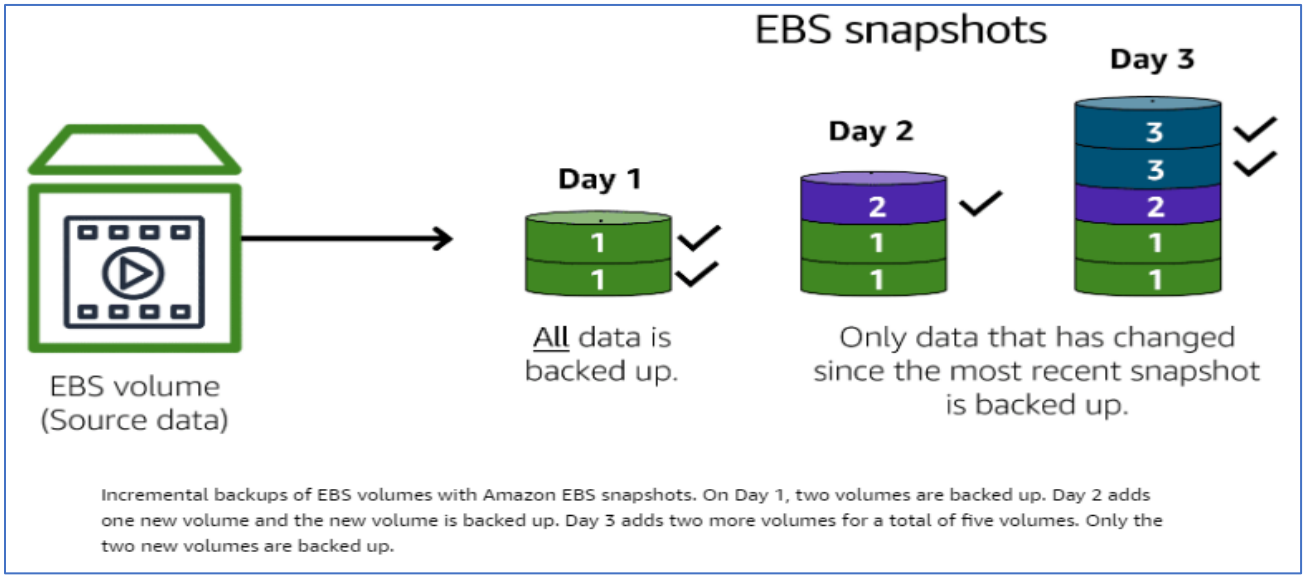


Amazon Elastic Block Store (Amazon EBS) is a service that provides block-level storage volumes that you can use with Amazon EC2 instances. If you stop or terminate an Amazon EC2 instance, all the data on the attached EBS volume remains available.

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:



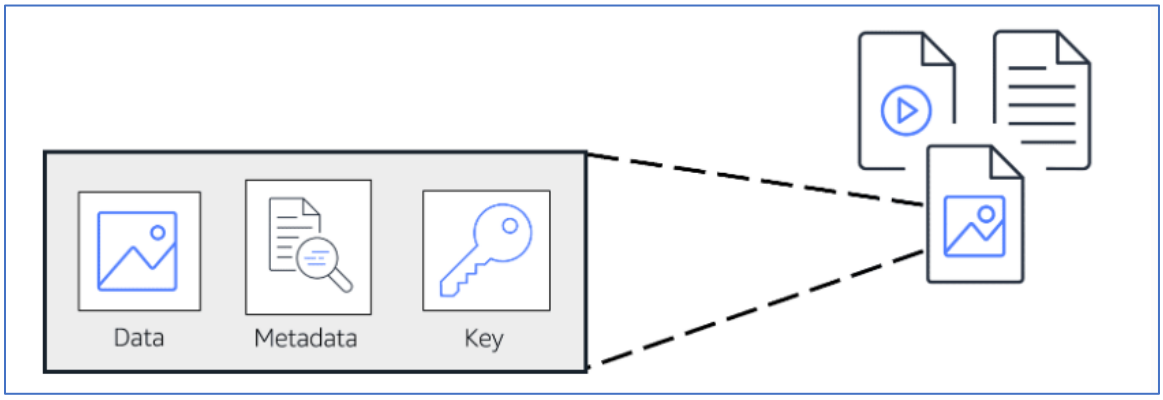
Incremental backups of EBS volumes with Amazon EBS snapshots. On Day 1, two volumes are backed up. Day 2 adds one new volume and the new volume is backed up. Day 3 adds two more volumes for a total of five volumes. Only the two new volumes are backed up.

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.

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!

To learn more about Amazon S3 storage classes, expand each of the following eight categories.
S3 offers multiple storage classes, each designed for different use cases and performance requirements:

Feature	S3 Standard	S3 Standard-IA	One Zone-IA	S3 Glacier	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive	S3 Outposts	S3 Intelligent-Tiering
Cost per GB per month	$0.02	$0.01	$0.01	$0.00	$0.00	$0.00	$0.00	$0.03	$0.015–0.025
Access time	1-15 seconds	3-5 minutes	3-5 minutes	12-48 hours	1-5 minutes	1-5 minutes	12-48 hours	Varies	Varies
Durability	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Availability	99.90%	99.90%	99.90%	99.90%	99.90%	99.90%	99.90%	99.90%	99.90%
Minimum storage duration	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

a. **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.

b. **S3 Standard-IA (S3 Standard-Infrequent Access)**
- 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.

c. **S3 One Zone-IA (S3 One Zone-Infrequent Access):**
- Stores data in a single Availability Zone
- Has a lower storage price than Amazon S3 Standard-IA

Compared to S3 Standard and S3 Standard-IA, which store data in a minimum of three Availability Zones, 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.

d. S3 Intelligent-Tiering:

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

In the 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, S3 Standard-IA. If you access an object in the infrequent access tier, Amazon S3 automatically moves it to the frequent access tier, S3 Standard.

e. S3 Glacier Instant Retrieval:

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

e. S3 Glacier Flexible Retrieval

-> Low-cost storage designed for data archiving.
-> Able to retrieve objects within a few minutes to hours. (1-12hrs)

For example, you might use this storage class to store archived customer records or older photos and video files. You can retrieve your data from S3 Glacier Flexible Retrieval from 1 minute to 12 hours.

g. S3 Glacier Deep Archive

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

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

h. 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.

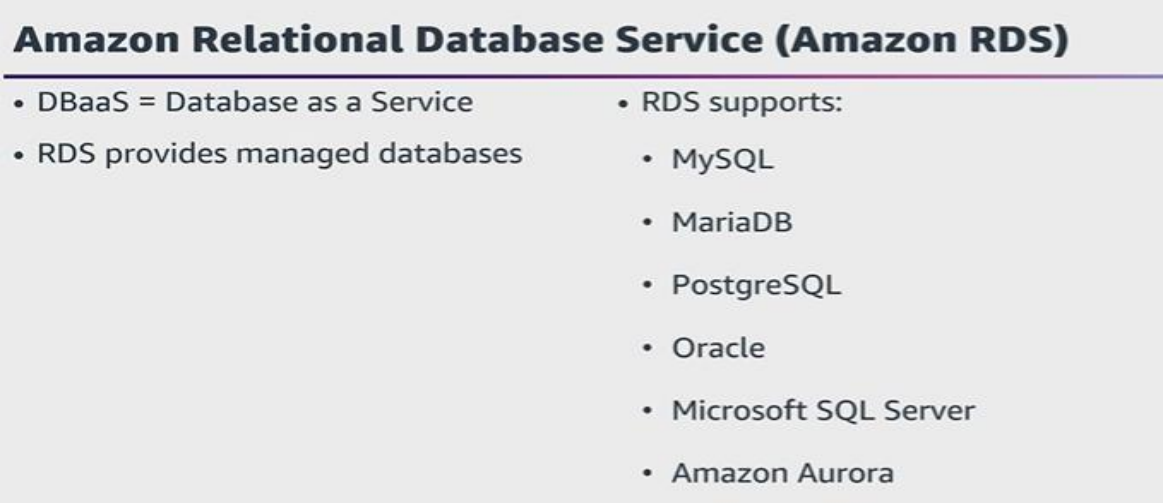
Comparing Amazon EBS and Amazon S3

Category	S3	EBS	EFS
Storage Type	Object Storage	Block Storage	File Storage
Pricing	Pay as you Use	Pay for provisioned capacity	Pay as you Use
Storage Size	Unlimited Storage	Limited storage	Unlimited Storage
Scalability	Unlimited Scalability	Increase/decrease size manaually	Unlimited Scalability
Durability	Stored redundantly across multiple Azs	Stored redundantly in a Single AZ	Stored redundantly across multiple Azs
Availability	Max is 99.99% with S3 Standard	99.99%	No SLAs
Security	Supports Data at Rest and Data in Transit encryption	Supports Data at Rest and Data in Transit encryption	Supports Data at Rest and Data in Transit encryption
Back up and Restore	Use Versioning or cross-region replication	Automated Backups and Snapshots	EFS to EFS replication
Performance	Slower than EBS and EFS	Faster than S3 and EFS	Faster than S3, Slower than EBS
Accessibility	Publicly and Privately accessible	Accessible only via the attached EC2 instance	Accessible simulatenously from multiple EC2 and on-premises instance
Interface	Web Interface	File System Interface	Web and File System Interface
Use cases	Media, Entertainment, Big data analytics, backups and archives, web serving and content management	Boot volumes, transactional and NoSQL databases, data warehousing ETL	Media, Entertainment, Big data analytics, backups and archives, web serving and content management, home directories

Comparing Amazon EBS and Amazon EFS

- ➔ 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 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)



- ➔ 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

ID	Product name	Size	Price
2	Dark roast ground coffee	20 oz.	$9.27

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

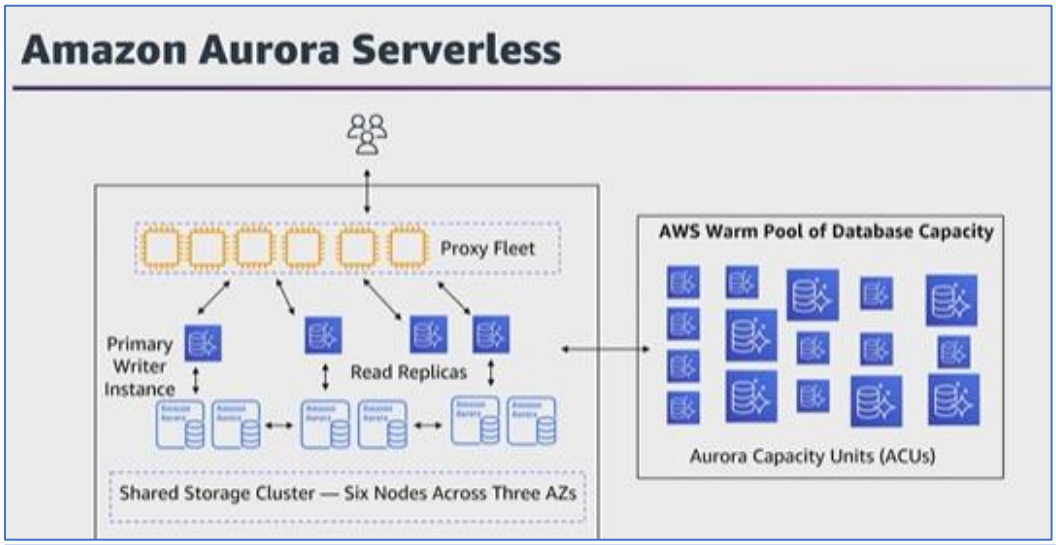
Amazon RDS is available on six database engines, which optimize for memory, performance, or input/output (I/O). Supported database engines include:

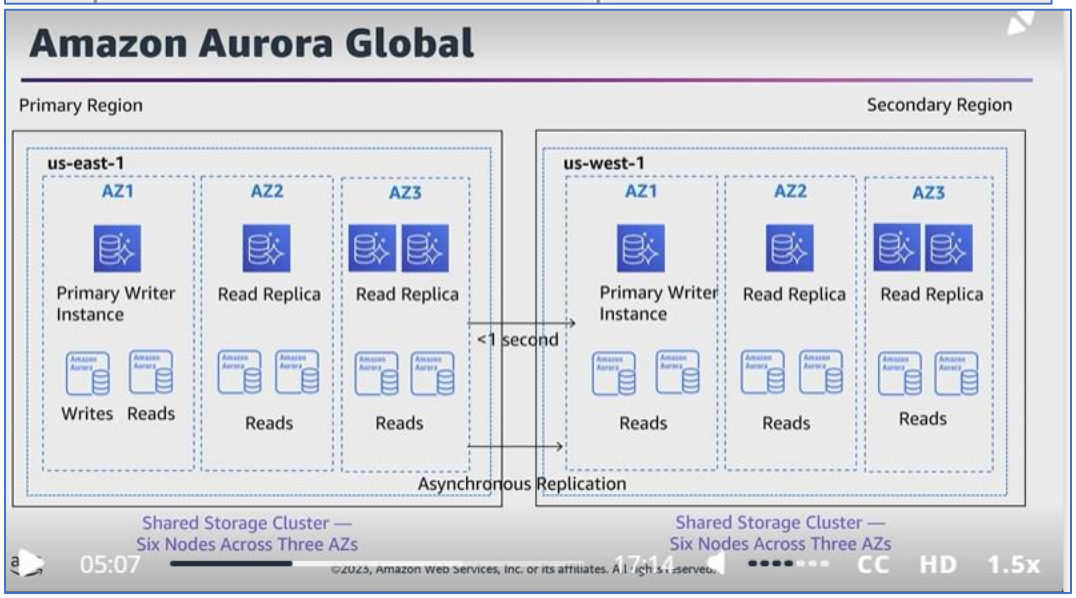
- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle Database
- Microsoft SQL Server

Amazon Aurora (5x faster than std MYSQL & 3xfaster than Std PostgresSQL)

- **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:

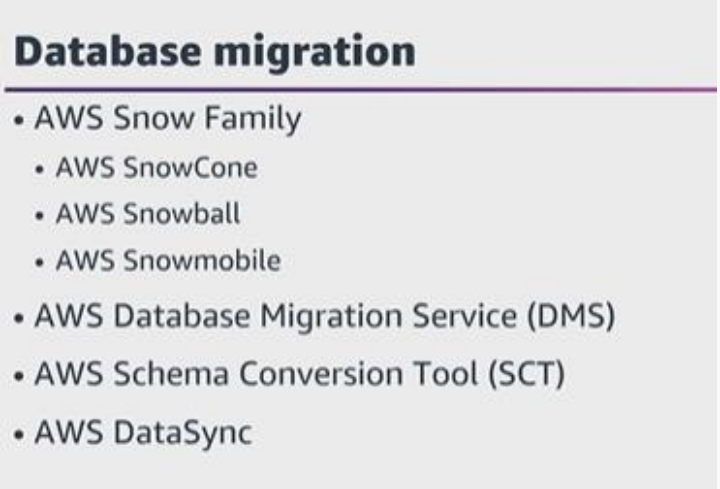
Key	Value
1	**Name**: John Doe **Address**: 123 Any Street **Favorite drink**: Medium latte
2	**Name**: Mary Major **Address**: 100 Main Street **Birthday**: July 5, 1994

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

1. **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.
2. **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 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 (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:

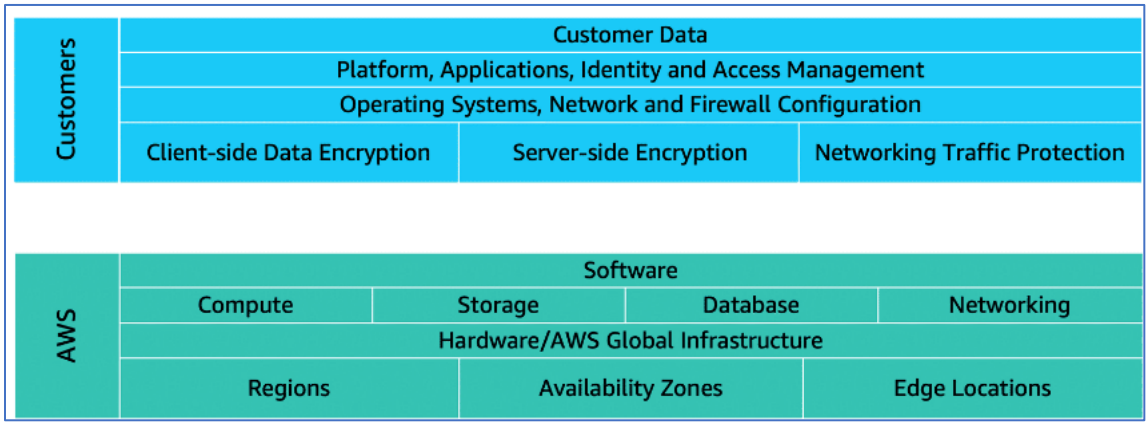
- ➔ **Amazon DocumentDB** is a document database service that supports MongoDB workloads. (MongoDB is a document database program.)
- ➔ **Amazon Neptune** is a graph database service.
  You can use Amazon Neptune to build and run applications that work with highly connected datasets, such as recommendation engines, fraud detection, and knowledge graphs.
- ➔ **Amazon Quantum Ledger Database (Amazon QLDB)** 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** 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** 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 (DAX)** is an in-memory cache for DynamoDB.
  It helps improve response times from single-digit milliseconds to microseconds.

Module 6 Introduction: Security

AWS Shared Responsibility Model:

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").



- **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.

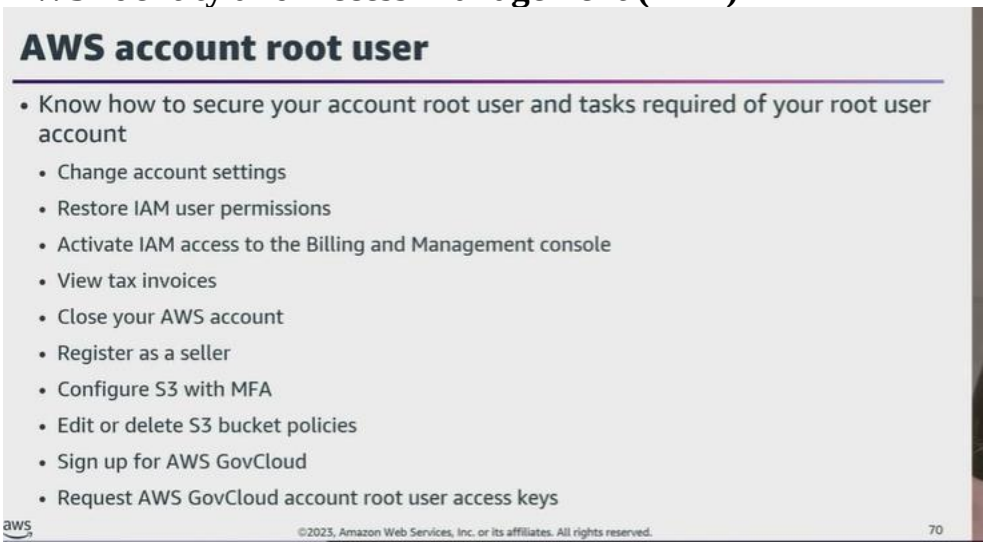
➔ AWS is responsible for protecting the global infrastructure that runs all the services offered in the AWS Cloud. This infrastructure includes AWS Regions, Availability Zones, and edge locations.

➔ 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 Permissions and Access:
AWS Identity and Access Management (IAM)



- IAM users
- IAM groups
- IAM roles
  - Temporary credentials
- IAM policies
  - Policy types
  - IAM Policy Simulator
- IAM integration with other AWS services

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

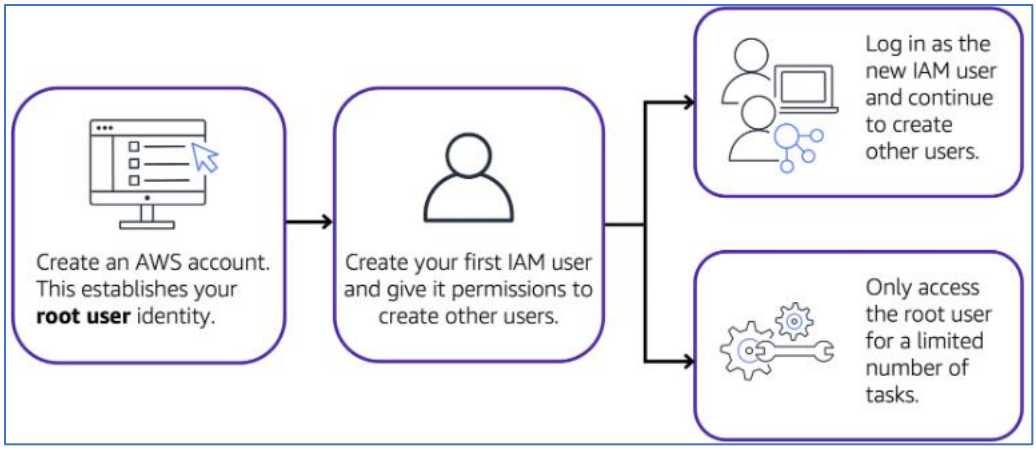
IAM gives you the flexibility to configure access based on your company's specific operational and security needs. You do this by using a combination of IAM features, which are explored in detail in this lesson:

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

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.



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.

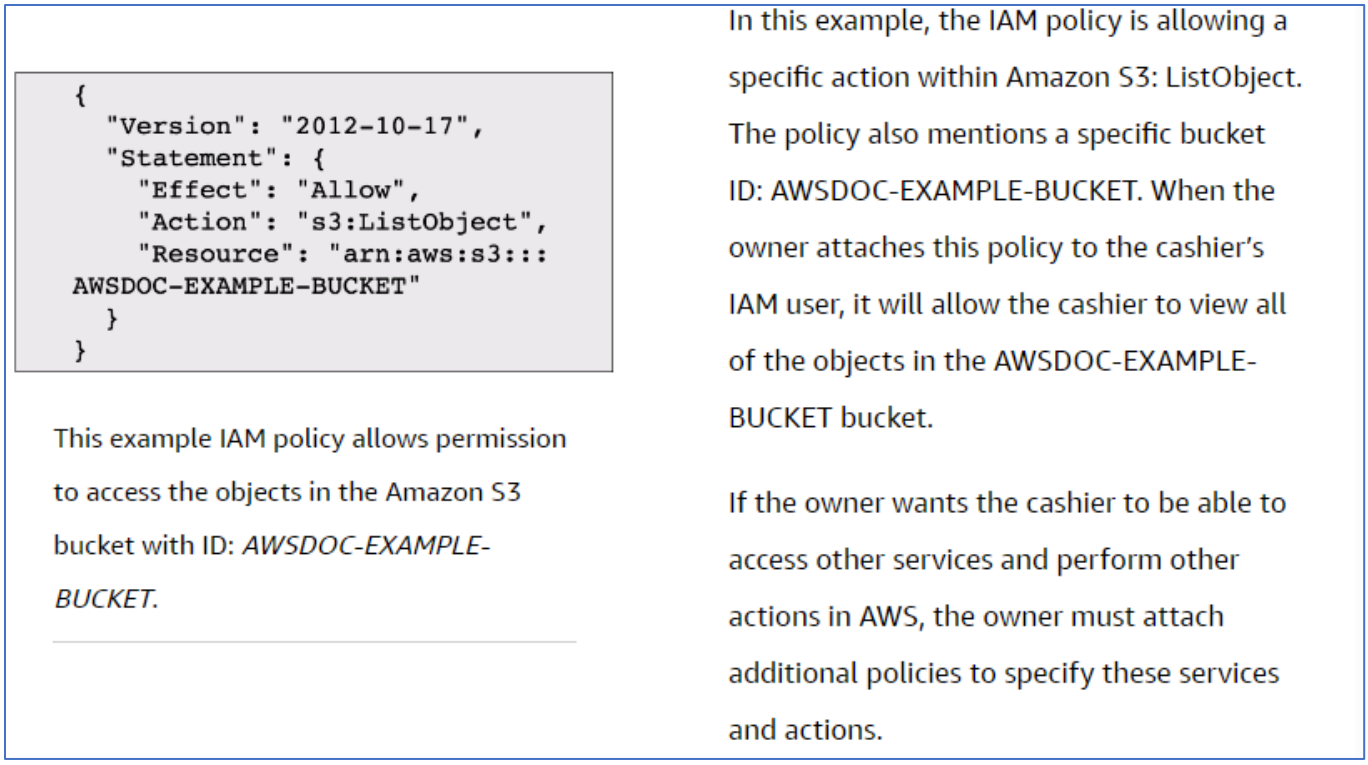
By default, when you create a new IAM user in AWS, it has no permissions associated with it. To allow the IAM user to perform specific actions in AWS, such as launching an Amazon EC2 instance or creating an Amazon S3 bucket, you must grant the IAM user the necessary permissions.

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.

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.

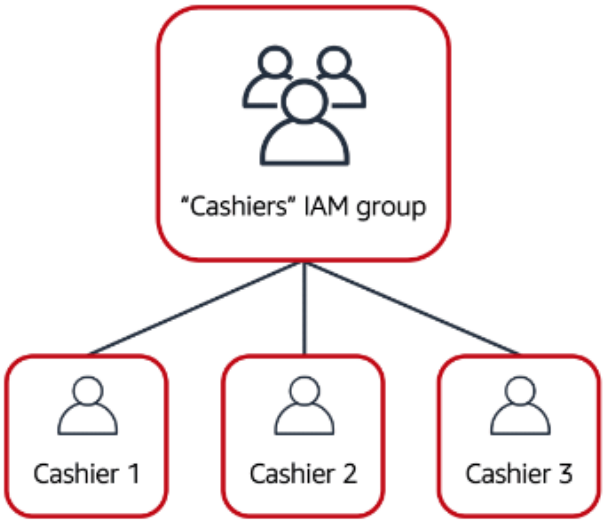


Now, suppose that the coffee shop has hired a few more cashiers. Instead of assigning permissions to each individual IAM user, the owner places the users into an **IAM group**.

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

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.

Multi-factor authentication

multi-factor authentication.

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

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: 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.

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.

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.

1. **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).

2. **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.

AWS Artifact provides access to AWS security and compliance documents, such as AWS ISO certifications, Payment Card Industry (PCI) reports, and Service Organization Control (SOC) reports. To learn more about the available compliance reports, visit AWS Compliance Programs.

Customer Compliance Center

The **Customer Compliance Center** contains resources to help you learn more about AWS compliance.

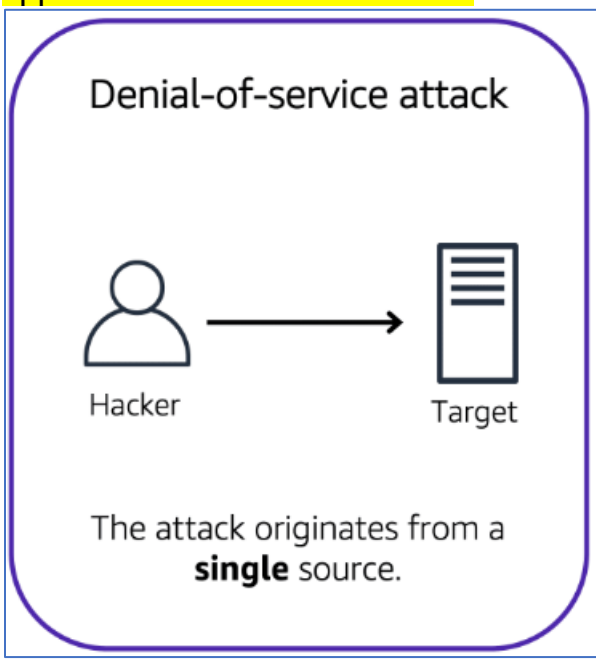
In the Customer Compliance Center, you can read customer compliance stories to discover how companies in regulated industries have solved various compliance, governance, and audit challenges.

You can also access compliance whitepapers and documentation on topics such as:

- AWS answers to key compliance questions
- An overview of AWS risk and compliance
- An auditing security checklist

Additionally, the Customer Compliance Center includes an auditor learning path. This learning path is designed for individuals in auditing, compliance, and legal roles who want to learn more about how their internal operations can demonstrate compliance using the AWS Cloud.

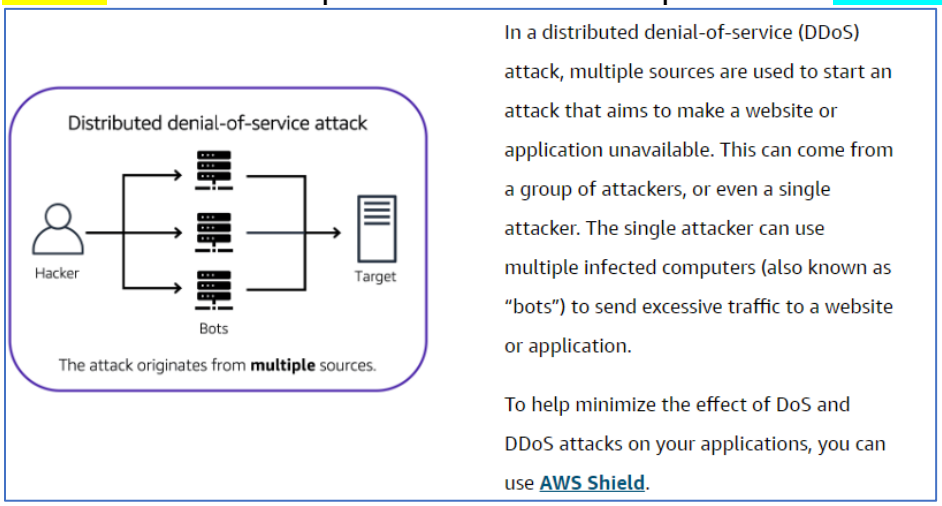
Denial-of-Service Attacks:
A **denial-of-service (DoS) attack** is a deliberate attempt to make a website or application unavailable to users.



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:
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**.
AWS Shield: https://aws.amazon.com/shield

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



1. **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 attacks.
   - 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.

2. **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:

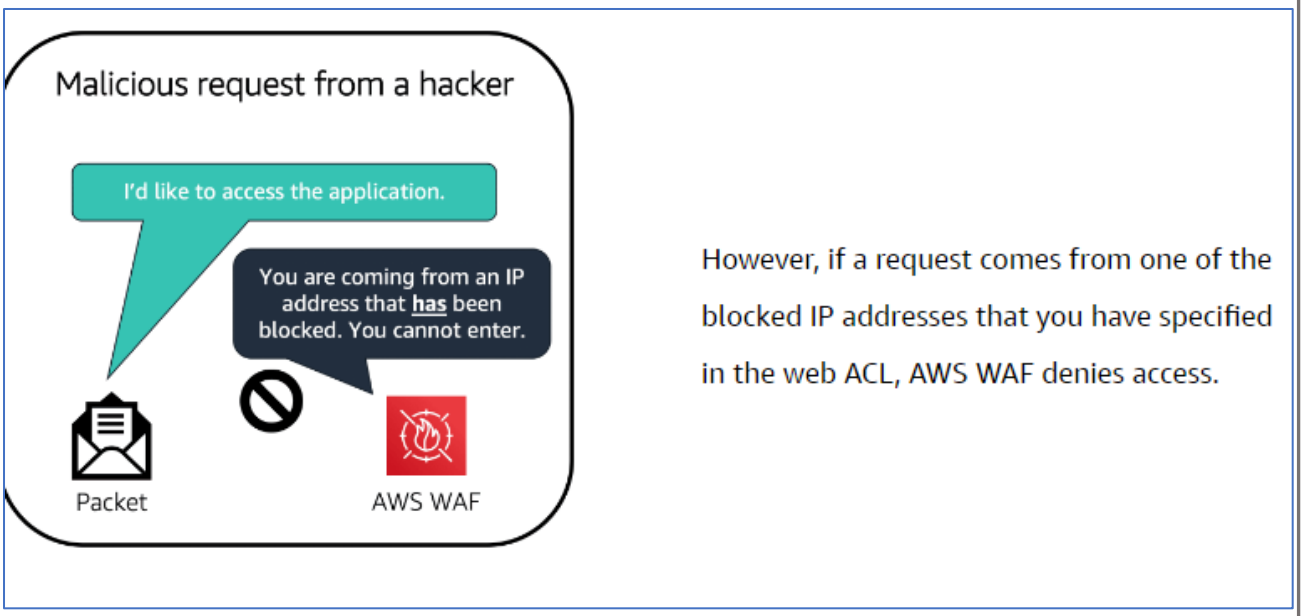
- **AWS Key Management Service (AWS KMS)** You must ensure that your applications' data is secure while in storage **(encryption at rest)** and while it is transmitted, known as **encryption in transit**.
- **AWS Key Management Service (AWS KMS)** enables you to perform encryption operations through 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** 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.
- 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 does not come from one of the blocked IP addresses, it allows access to the application.

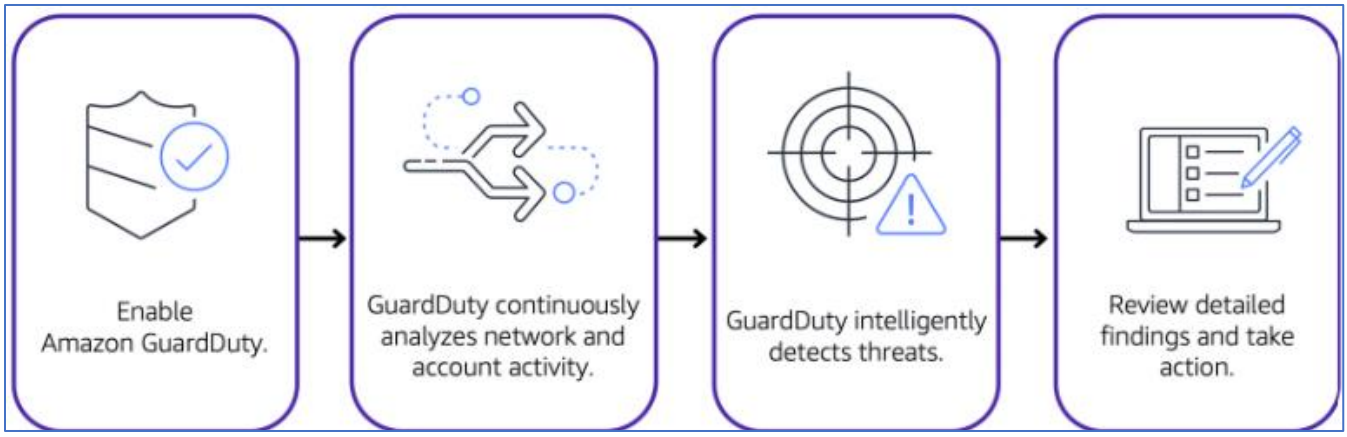


-
- However, if a request comes from one of the blocked IP addresses that you have specified in the web ACL, AWS WAF denies access.

- **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** 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.

Additional 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
- Security best practices in IAM

Module 7: Monitoring and Analytics

Idea of observing systems, collecting metrics, evaluating those metrics over time, and then using them to make decisions or take actions, is what we call monitoring.

It's important to set up monitoring in the cloud. With the elastic nature of AWS services that dynamically scale up and down, you'll want to keep a close pulse on your AWS resources to ensure that your systems are running as expected.

1. 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.

- **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.

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 know who, when, or which method created the user.

1. 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 know 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.

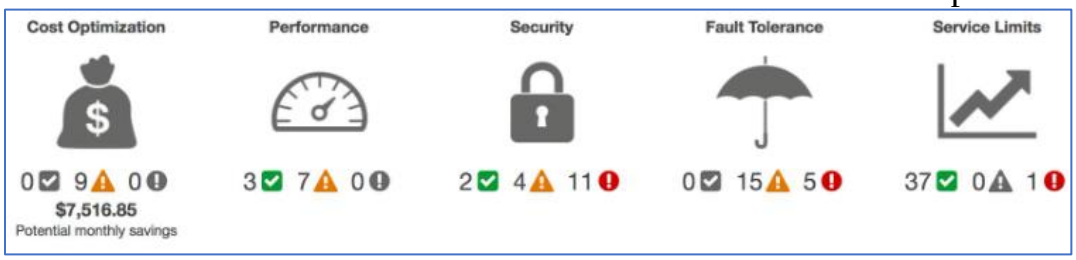
CloudTrail Insights

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.

2. 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.



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**.

Additional resources:

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

Module 8: Pricing and Support

- Describe AWS pricing and support models.
- Describe the AWS Free Tier.
- Describe key benefits of AWS Organizations and consolidated billing.
- Explain the benefits of AWS Budgets.
- Explain the benefits of AWS Cost Explorer.
- Explain the primary benefits of the AWS Pricing Calculator.
- Distinguish between the various AWS Support Plans.
- Describe the benefits of AWS Marketplace.

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.

1. **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.

2. **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.

3. **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:

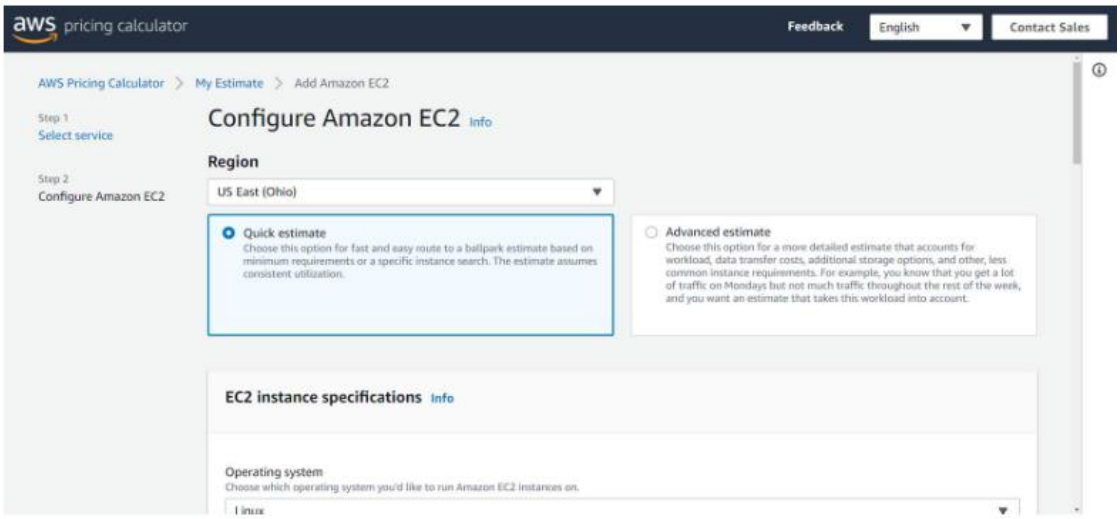
AWS offers a range of cloud computing services with pay-as-you-go pricing.

- **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:

1. **AWS Lambda:**

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**.

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

2. **Amazon EC2: 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.

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 (ELB) 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

3. Amazon S3:

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.

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:

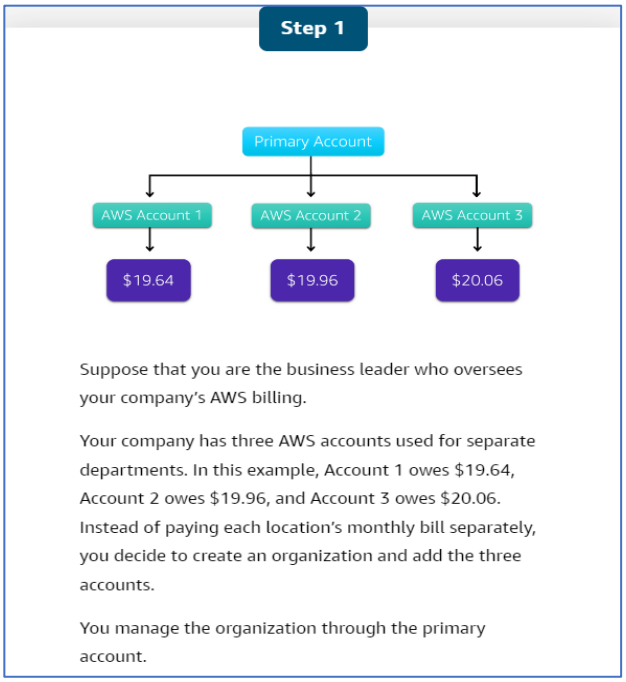
In an earlier module, you learned about AWS Organizations, a service that enables you to manage multiple AWS accounts from a central location. AWS Organizations also provides the option for **consolidated billing**.

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.

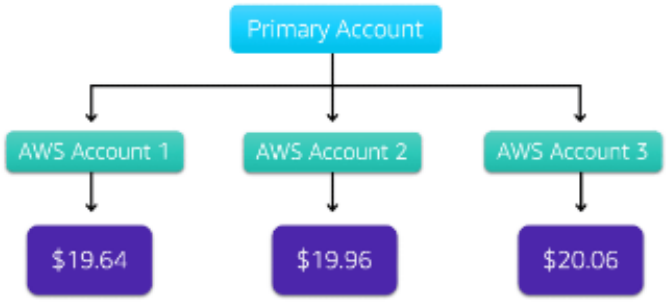
On your monthly bill, you can review itemized charges incurred by each account. This enables you to have greater transparency into your organization's accounts while still maintaining the convenience of receiving a single monthly bill.

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.

To review and example of consolidated billing, choose the arrow buttons to display the four steps.



Step 2



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

Continuing the example, 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. In this case, the primary account incurred $14.14. 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). The total charged to the paying account, including the primary account and accounts one through three, is $73.80.

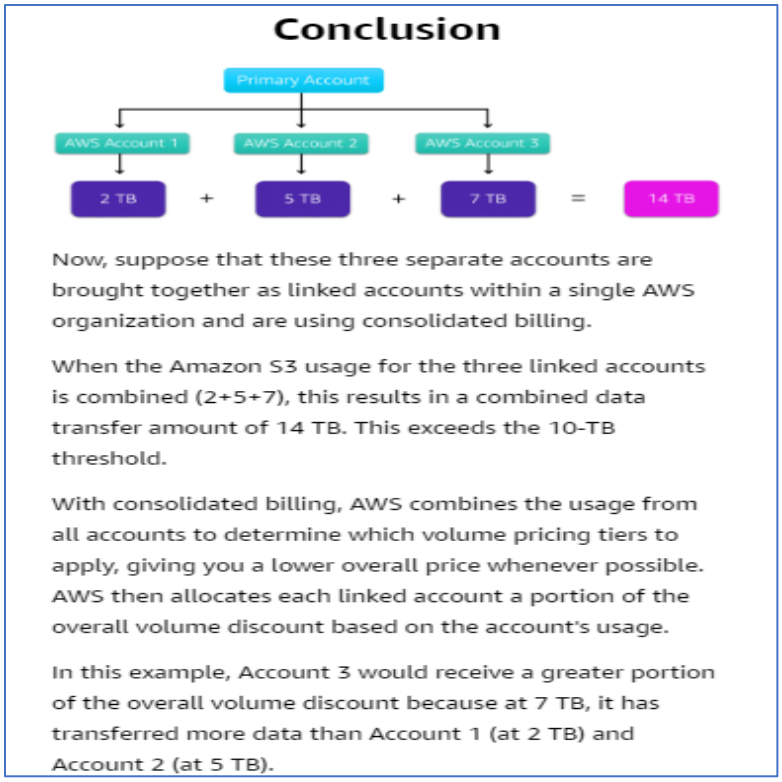
Consolidated billing also enables you to share volume pricing discounts across accounts.

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.



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.

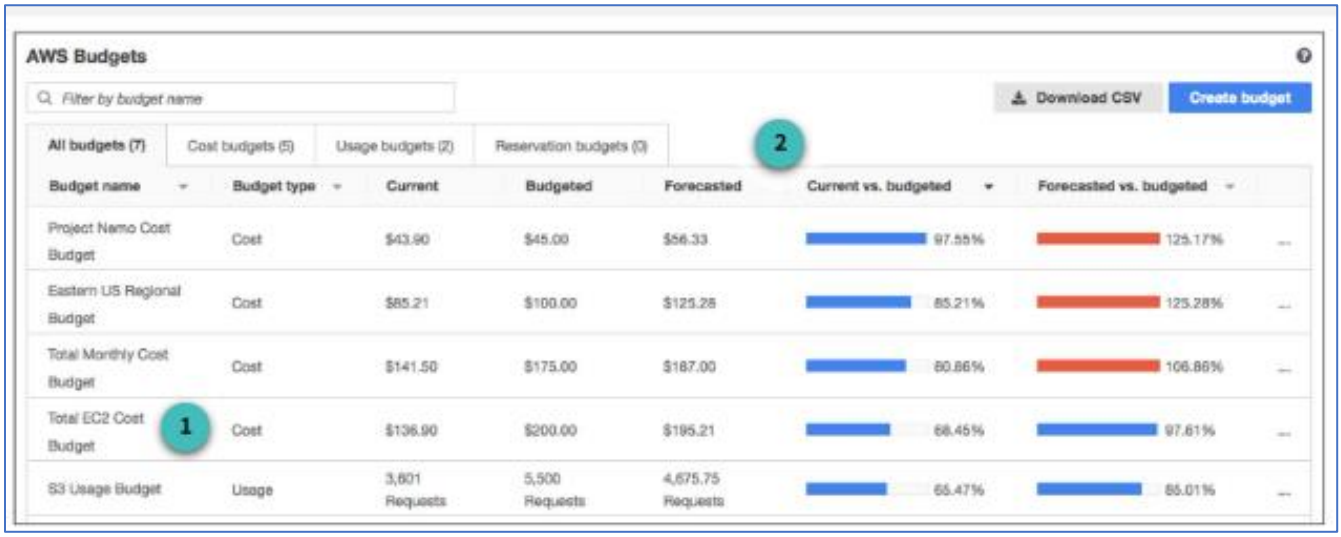
In AWS Budgets, you can also set custom alerts when your usage exceeds (or is forecasted to exceed) the budgeted amount.

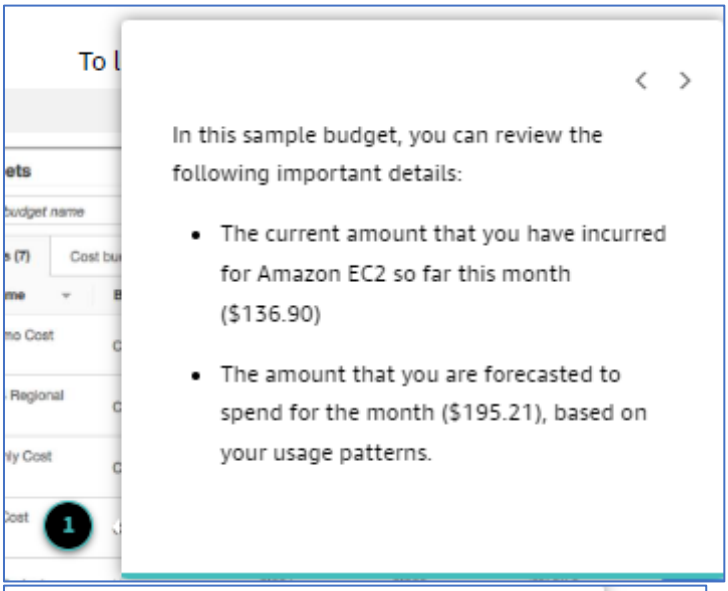
Example: AWS Budgets

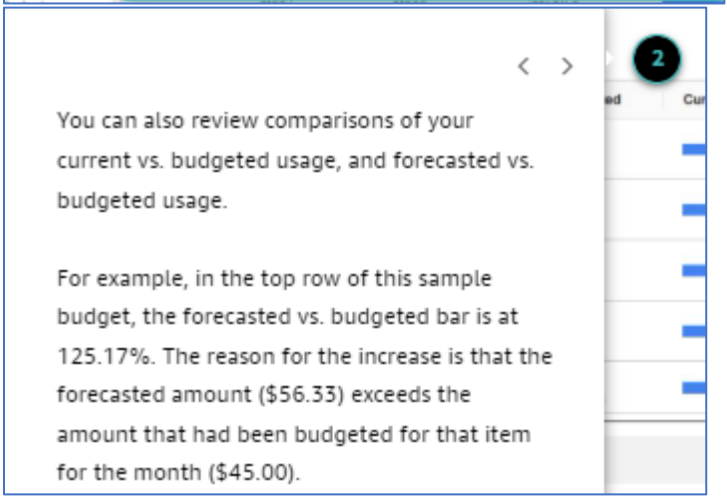
Suppose that you have set a budget for Amazon EC2. You want to ensure that your company's usage of Amazon EC2 does not exceed $200 for the month.

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.

To learn more about AWS Budgets, choose each of the numbered markers.



- 

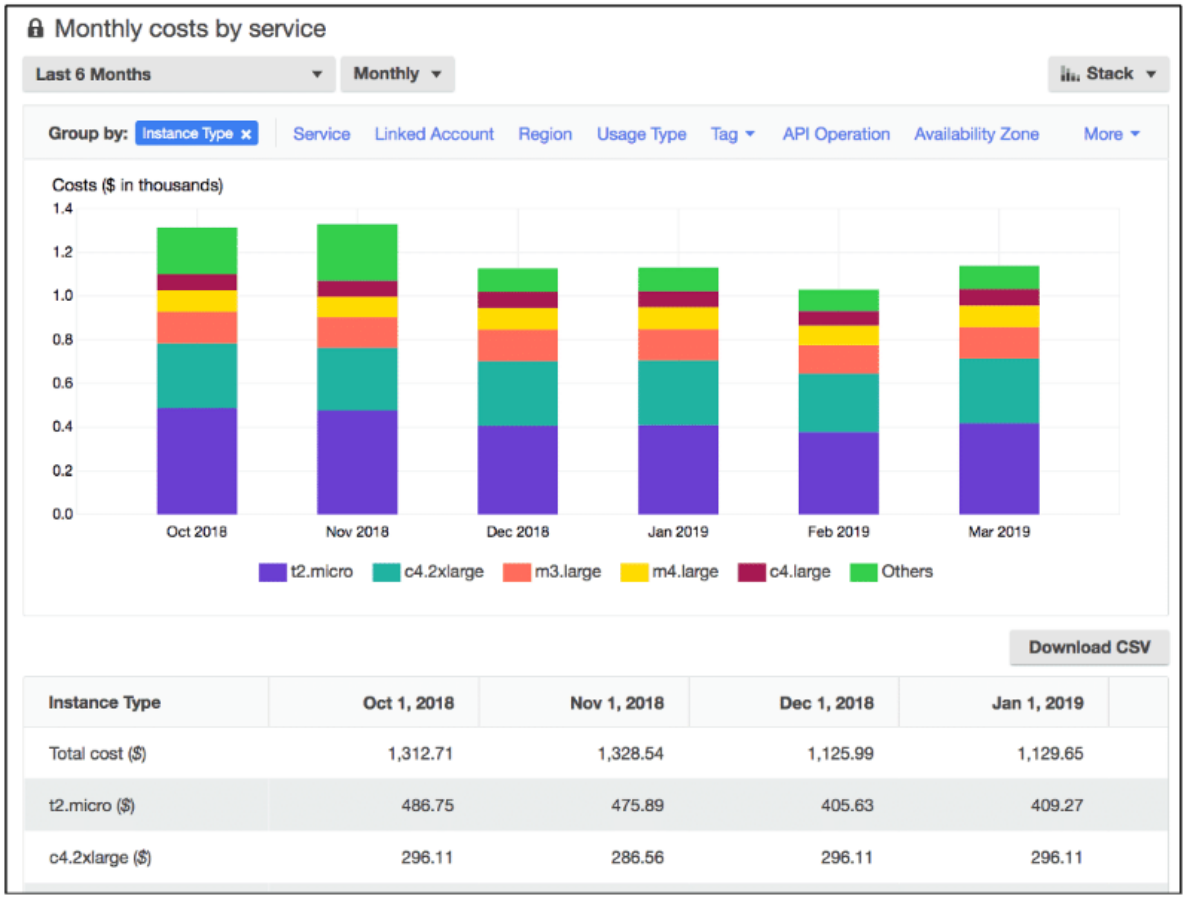
- 

AWS Cost Explorer:

AWS Cost Explorer is a tool that lets you 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 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, and 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 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 TAM.

- 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.

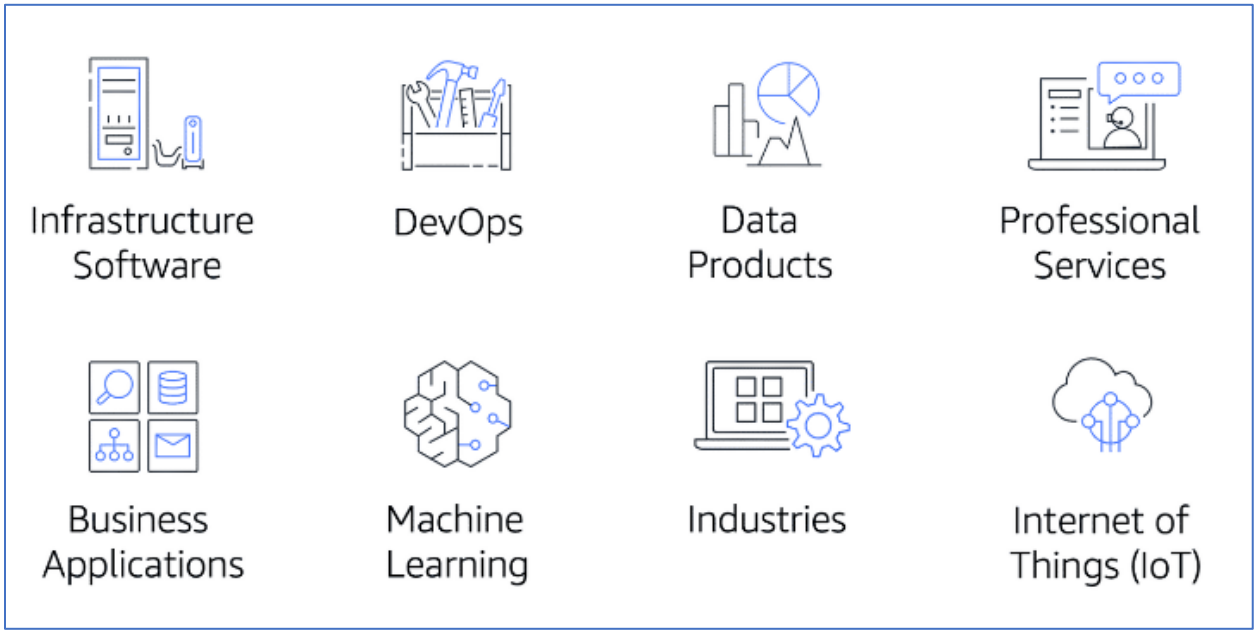
AWS Marketplace:

AWS Marketplace 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 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.

In Module 8, you learned about the following concepts:: Three types of offers included in the AWS Free Tier: 12 months free, Always free, and Trials

- Benefits of consolidated billing in AWS Organizations
- Tools for planning, estimating, and reviewing AWS costs
- Differences between the five AWS Support plans: Basic, Developer, Business, Enterprise On-Ramp, and Enterprise
- How to discover software in AWS Marketplace

Additional resources

- AWS Pricing, AWS Free Tier, AWS Cost Management, Whitepaper: How AWS Pricing Works, Whitepaper: Introduction to AWS Economics, AWS Support, AWS Knowledge Center

Module 9: Migration and Innovation:

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** 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** 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** 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 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** 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** 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** 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**, 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.

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

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** is a small, rugged, and secure edge computing and data transfer device.
  It features 2 CPUs, 4 GB of memory, and up to 14 TB of usable storage.

- **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.

  - Storage: 80 TB of hard disk drive (HDD) capacity for block volumes and Amazon S3 compatible object storage, and 1 TB of SATA solid state drive (SSD) for block volumes.
  - Compute: 40 vCPUs, and 80 GiB of memory to support Amazon EC2 sbe1 instances (equivalent to C5).

- **Snowball Edge Compute Optimized** provides powerful computing resources for use cases such as machine learning, full motion video analysis, analytics, and local computing stacks.
  - Storage: 80-TB usable HDD capacity for Amazon S3 compatible object storage or Amazon EBS compatible block volumes and 28 TB of usable NVMe SSD capacity for Amazon EBS compatible block volumes.
  - Compute: 104 vCPUs, 416 GiB of memory, and an optional NVIDIA Tesla V100 GPU. Devices run Amazon EC2 sbe-c and sbe-g instances, which are equivalent to C5, M5a, G3, and P3 instances.

- **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.

Innovation 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 application:** 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.

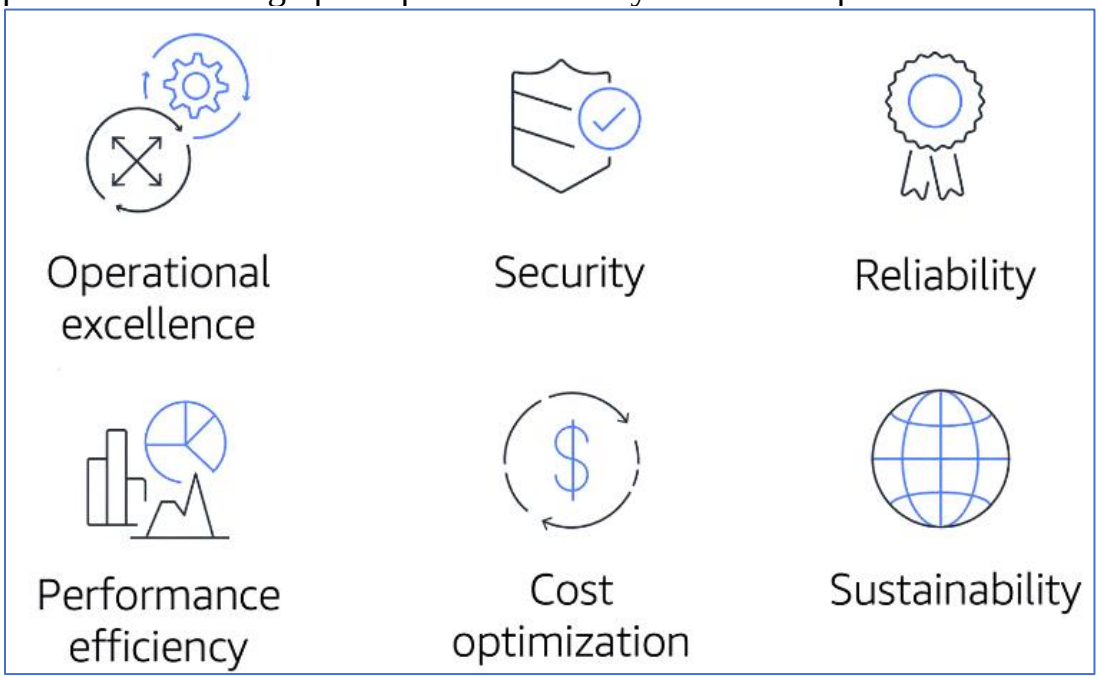
➔ AWS offers Amazon SageMaker to remove the difficult work from the process and empower you to build, train, and deploy ML models quickly.

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

Additional 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 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** 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** 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** 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** 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** 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.

Additional resources

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

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

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.

Domain	Percentage of exam
Domain 1: Cloud Concepts	24%
Domain 2: Security and Compliance	30%
Domain 3: Technology	34%
Domain 4: Billing and Pricing	12%
Total	100%

Exam details

The AWS Certified Cloud Practitioner exam consists of **65 questions** to be completed in **90 minutes**. The minimum passing score is **700** (the maximum score is 1,000).

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.

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

Course outline

Module 1: Get to know the exam

- Introduction to AWS Certified Cloud Practitioner (CLF-C02)
- Overview: AWS Certified Cloud Practitioner (CLF-C02)
- Exam guide: AWS Certified Cloud Practitioner (CLF-C02)

Module 2: Get to know exam-style questions

- Introduction to exam-style questions
- Overview and Instructions: Official Practice Question Set
- AWS Certified Cloud Practitioner Official Practice Question Set (CLF-C02 - English)

Module 3: Learn about exam topics

- AWS training recommendations
- Whitepapers

Module 4: Prepare for the exam

Domain 1: Cloud Concepts

- Lesson 1: Introduction to Cloud Concepts
- Lesson 2: Define the benefits of the AWS Cloud
- Lesson 3: Identify design principles of the AWS Cloud
- Lesson 4: Understand the benefits of the strategies for migration to the AWS Cloud
- Lesson 5: Understand the concepts of cloud economics
- Lesson 6: Walkthrough question 1
- Lesson 7: Walkthrough question 2
- Additional resources

Domain 2: Security and Compliance

- Lesson 1: Introduction to Security and Compliance
- Lesson 2: Understand the Shared Responsibility Model
- Lesson 3: Understand Cloud security, governance, and compliance concepts
- Lesson 4: Identify AWS access management capabilities
- Lesson 5: Identify components and resources for security
- Lesson 6: Walkthrough question 3

- Lesson 7: Walkthrough question 4
- Additional resources

Domain 3: Cloud Technology and Services

- Lesson 1: Introduction to Cloud Technology and Services
- Lesson 2: Define methods of deploying and operating in the AWS Cloud
- Lesson 3: Define the AWS global infrastructure
- Lesson 4: Identify AWS compute resources
- Lesson 5: Identify AWS database resources
- Lesson 6: Identify AWS Network resources
- Lesson 7: Identify AWS storage resources
- Lesson 8: Identify AWS artificial intelligence and machine learning services and analytics services
- Lesson 9: Identify services from other in-scope AWS service categories
- Lesson 10: Walkthrough question 5
- Lesson 11: Walkthrough question 6
- Additional resources

Domain 4: Billing, Pricing, and Support

- Lesson 1: Introduction to Billing, Pricing, and Support
- Lesson 2: Compare AWS pricing models
- Lesson 3: Understand resources for billing, budget, and cost management
- Lesson 4: Identify AWS technical resources and AWS Support options
- Lesson 5: Walkthrough question 7
- Lesson 6: Walkthrough question 8
- Additional resources

Module 5: Register for the exam

- Register for the exam
- What to Expect: Taking an Online-Proctored Exam

AWS Cloud Practitioner-Exam Prep coutse transcripts: AWS Skill Builder

Domain 1: Cloud Concepts
Domain1: Introduction:
Let's get started with Domain 1 covering cloud concepts such as what is AWS, the benefits of AWS, design principles, migration, and economics.
Domain 1: Cloud concepts is broken into four task statements that we will discuss over the next few lessons.

- Task statement 1.1: Define the benefits of the AWS Cloud.
- Task statement 1.2: Identify design principles of the AWS Cloud.
- Task statement 1.3: Understand the benefits of and strategies for migration to the AWS Cloud.
- And task statement 1.4: Understand concepts of cloud economics.

Before the exam, know the seven migration strategies.

- Retire is for applications you want to decommission or retire.
- Retain is for applications that you want to keep in the source environment or applications that are not ready to migrate.
- Rehost, also known as lift and shift are for applications to migrate without making any changes to the application.
- Relocate is for a large number of servers that are made up of one or more applications.
- Repurchase is also known as drop and shop and is for applications with a different version or product and provides more value than the existing infrastructure.
- Replatform is also known as lift, tinker, and shift and is for applications that need some level of optimization in order to operate efficiently or take advantage of AWS capabilities.
- Refactor or re-architect is for applications that you want to migrate to AWS and take full advantage of the cloud-native features to improve agility, performance, and scalability.

The question reads: "**Which AWS Cloud architecture design principle supports the distribution of workloads across multiple Availability Zones?**"
Reading this question, can you identify any keywords or phrases? And also, what exactly is the question asking? A few keywords I see are design principles and distribution across multiple Availability Zones.
Now that we have examined the question, identified some keywords, and reviewed the requirements, let's explore the responses.

- Option A, Implement automation.

- Option B, Design for agility.
- Option C, Design for failure.
- And Option D, Implement elasticity.

Pause the video if you need more time. Okay, let's evaluate the options.

Option A is incorrect. You can use automation services, such as AWS CloudFormation, to deploy resources into one or more Availability Zones. However, the implementation of automation is not directly tied to, or limited to, the distribution of workloads across multiple Availability Zones.
Option B is incorrect. When you design for agility, you can provision resources more quickly. Agility is not related to the number of Availability Zones.
Option C is to design for failure. AWS recommends that you distribute workloads across multiple Availability Zones. This distribution will ensure continuous availability of your application, even if the application is unavailable in one single Availability Zone. This makes option C a good candidate for the correct answer, but let's look at the rest of the responses.
Option D is also incorrect. Elasticity is the ability to activate resources as you need them and return resources when you no longer need them. Elasticity is not related to the number of Availability Zones.
Option C is the correct answer. That's all for this question. Be sure to take note of any knowledge gaps that you may have identified while exploring this question. And let's get started with our second walkthrough question.

Walkthrough question 2

–

The question reads, "**A system administrator is reviewing a group of servers that were found during a portfolio discovery. All servers are migrating to AWS. The servers have no current owner. There is very little traffic to the servers. Which migration strategy should the system administrator suggest for these servers?"**

Reading this question, can you identify any keywords or phrases and exactly what this question is asking? A few keywords I see are servers, portfolio discovery, migrating to AWS, and little traffic. Also, the final ask is looking for a migration strategy for the found servers.
Now that we've examined the question, identified keywords, and reviewed the requirements, let's explore the responses.

- Option A, rehost
- Option B, replatform
- Option C, retain
- Option D, retire

Pause the video if you need more time. Okay, let's evaluate the options.
Option A is incorrect. To re-host a server means to move the server to the AWS Cloud without making any changes to the server. Because the servers are rarely used, the servers can be shut down to save on cost.
Option B is incorrect. To re-platform a server means to move the server to the AWS Cloud and make changes to optimize the server. Because the servers are rarely used, the servers can be shut down to save on cost.
Option C is also incorrect. To retain a server means to keep the server in the current environment without moving the server to the AWS Cloud. Because the servers are rarely used and all servers are being migrated to AWS, the servers must be shut down.
So that makes option D correct. To retire a server means to decommission or remove the server from the environment. Because the servers are rarely used, the servers should be retired. The cost to move and keep the servers running may be more than the productivity that the servers provide. By retiring the servers, you can realize savings upfront by no longer having to pay for those servers.
That's all for this question. Be sure to take note of any knowledge gaps that you may have identified while exploring this question, and let's get started with domain 2, security and compliance.

Domain 2: Security and Compliance

- Task statement 2.1: understand the AWS shared responsibility model.
- Task statement 2.2: understand AWS Cloud security, governance, and compliance concepts.
- Task statement 2.3: identify AWS access management capabilities
- Task statement 2.4: identify components and resources for security.

Let's get started with our third walkthrough question, which is from task statement 2.1, understand the AWS shared responsibility model.
The question reads, **which task is the responsibility of the customer according to the AWS shared responsibility model?**
Reading this question, can you identify any keywords or phrases and also exactly what the question is asking? A few keywords I see are the shared responsibility model and the responsibility of the customer.
Now that we've examined the question, identified keywords, and reviewed the requirements, let's explore the responses.

- Option A, install patches on an Amazon RDS database instance.
- Option B, patch the operating system of Amazon RDS database instances.

- Option C, determine which services have access to an Amazon DynamoDB table.
- And Option D, patch the Amazon VPC Network devices.

Pause the video if you need more time. Okay, let's evaluate the options.

Option A is incorrect. AWS provides Amazon RDS as a service. AWS manages patches for the Amazon RDS engine. The customer can choose a time window to apply patches.
Option B is incorrect. AWS provides Amazon RDS as a service. The customer is not responsible for patches to the operating system.
Option C is to determine which services have access to the table. The user determines access permissions between services within the cloud. This makes option C a good candidate for the correct answer but let's look at the rest of the responses.
Option D is incorrect. AWS manages Network devices that provide AWS Network services. AWS also installs patches.
So that makes option C the correct answer. That's all for this question.
Be sure to take note of any knowledge gaps that you may have identified while exploring this question, and let's get started with our fourth walkthrough question.
Walkthrough question 4
–.
The question reads: "**A company has an application server that runs on an Amazon EC2 instance. The application server needs to access contents within a private Amazon S3 bucket. What is the recommended approach to meet this requirement?"**
Reading this question, can you identify any keywords or phrases and exactly what the question is asking? A few keywords I see are application server, EC2 instance, and access is needed to a private S3 bucket.
Now that we've examined the question, identified keywords, and reviewed the requirements, let's explore the responses.

- Option A: create an IAM role with the appropriate permissions. Associate the role with the EC2 instance.
- Option B: configure a VPC peering connection to allow private communication between the EC2 instance and the S3 bucket.
- Option C: create a shared access key. Configure the EC2 instance to use the hardcoded key.
- And option D: configure the application to read and access key from a secured source.

Pause the video if you need more time. Okay, let's evaluate the options.
Option A is to create an IAM role. IAM roles are temporary credentials that expire. IAM roles are more secure than long-term access keys, because they reduce risk if credentials are accidentally exposed. This makes option A a good candidate for the correct answer, but let's look at the rest of the responses.
Option B is incorrect. A VPC peering connection is a Network connection between two VPCs that enables you to route traffic between them by using private IP version 4 addresses or IP version 6 addresses. VPC peering connections cannot connect a VPC and an Amazon S3 bucket.
Option C is incorrect. The creation of a shared key is not a best practice, because this reduces the value of auditing AWS resource access. It is not a best practice to embed access keys into the application code, because application code can can become compromised.
Option D is also incorrect. It is a best practice to use IAM roles instead of long-term access keys. Long-term access keys, such as those associated with IAM users and AWS account root users remain valid until you manually revoke them. However, temporary security credentials that are obtained through IAM roles and other features of AWS security token service expire after a short period of time, using temporary security credentials to help reduce the risk in case credentials are accidentally exposed.
This makes option A the correct answer. That's all for this question.
Be sure to take note of any knowledge gaps that you may have identified while exploring this question and let's get started with domain 3, cloud technology and services.

Domain 3: Cloud Technology and Services

- Task statement 3.1: Define methods of deploying and operating in the AWS Cloud.
- Task statement 3.2: Define the AWS global infrastructure.
- Task statement 3.3: Identify AWS compute services.
- Task statement 3.4: Identify AWS database services.
- Task statement 3.5: Identify AWS Network services.
- Task statement 3.6: Identify AWS storage services.
- Task statement 3.7: Identify AWS artificial and machine learning services and analytics services.
- And task statement 3.8: Identify services from other in-scope AWS service categories.

Let's get started with our fifth walkthrough question which is from task statement 3.1: define methods of deploying and operating in the AWS Cloud.
The question reads**: A company wants a dedicated private connection to the AWS Cloud from its on-premises operation. Which AWS service or feature will provide this connection?**
Well, let's see if you were paying attention and explore the responses.

- Option A, AWS VPN
- Option B, AWS PrivateLink
- Option C, VPC endpoint
- And option D, AWS Direct Connect.

Pause the video if you need more time. Okay, let's evaluate the options.
Option A is incorrect. AWS VPN establishes a secure connections between your on-premises Network, remote offices, client devices, and the AWS global Network. AWS VPN is not a dedicated connection.

Option B is incorrect. You use PrivateLink when you want to use services offered by another VPC securely within the AWS Network. With PrivateLink, all Network traffic stays on the global AWS backbone and never traverses the public internet. PrivateLink does not connect to on-premises operations.

Option C is incorrect. A VPC endpoint enables private connections between your VPC and supported AWS services and VPC endpoint services powered by PrivateLink. A VPC endpoint does not connect to on-premises operations.

So that makes option D correct. Direct Connect provides a dedicated private connection from your on premises to the AWS Cloud. Direct Connect is an alternative to using the internet to access AWS services.

That's all for this question.

Be sure to take note of any knowledge gaps that you may have identified while exploring this question.

Let's get started with our sixth walkthrough question.
Walkthrough question 6
–
Let's get started with our sixth walkthrough question which is from task statement 3.2 define the AWS global infrastructure.

The question reads, **which aspect of AWS infrastructure provides global deployment of compute and storage? Reading this question, can you identify any keywords or phrases and also exactly what the question is asking?**
A few keywords I see are global deployment of compute and storage.
Now that we've examined the question, identified the keywords and reviewed the requirements, let's explore the responses.

- Option A is mulitple Availability Zones in an AWS Region.
- Option B is multiple AWS Regions.
- Option C is tags.
- Option D is resource groups.

Option A is incorrect. Availability Zones are one or more discrete data centers with redundant power Networking and connectivity in a Region. When infrastructure is deployed across multiple Availability Zones, you can achieve a highly available deployment within a geographical location of the Region. However, this solution does not provide global deployments.

Option B is multiple AWS Regions. A Region is a physical location where there are clusters of AWS data centers. AWS offers many different Regions where you can deploy infrastructure around the world. With the use of multiple Regions, you can achieve a global deployment of compute, storage, and databases. This makes option B a good candidate for the correct key but let's look at the other responses.
Option C is incorrect. Tags are metadata that you can associate with your AWS resources. Tags are user-defined data in the form of key value pairs. You can use tags to manage, identify, organize, and search for, and filter resources. Tags do not provide global deployments of applications and solutions.
Option D is also incorrect. AWS resource groups is a service that you can use to manage and automate tasks on many resources at the same time, resources in AWS are entities such as an Amazon EC2 instance and Amazon S3 buckets. With resource groups, you can filter resources based on tags or AWS CloudFormation stacks and then perform an action against a group of resources. You do not use resource groups to deploy AWS resources globally.
That makes option B the correct answer.
That's all for this question. Be sure to take note of any knowledge gaps that you may have identified while exploring this question, and let's get started with Domain 4. Billing, pricing, and support.

Domain 4: Billing, Pricing, and Support

- Task statement 4.1, compare AWS pricing models
- Task statement 4.2, understand resources for billing, budget, and cost management
- Task statement 4.3, identify AWS technical resources for AWS support options.

Let's get started with our seventh walkthrough question, which is from task statement 4.1: Compare AWS pricing models.
The question reads, "**A company must meet compliance and software licensing requirements that state a workload must be hosted on a physical server. Which Amazon EC2 instance pricing option will meet these requirements?"**
Reading this question, can you identify any keywords or phrases and exactly what the question is asking? A few keywords I see are compliance and software requirements and physical server.
Now that we've examined the question, identified key words, and reviewed the requirements, let's explore the responses.

- Option A, Dedicated Hosts
- Option B, Dedicated Instances
- Option C, Spot Instances
- And Option D, Reserved Instances.

Pause the video if you need more time. Okay, let's evaluate the options.
Option A is Dedicated Hosts. An EC2 Dedicated Host is a physical server with EC2 instance capacity that is fully dedicated to your use. This makes Option A a good candidate for the correct answer, but let's look at the rest of the responses.
Option B is incorrect. Dedicated Instances are EC2 instances that run in a VPC on hardware that is dedicated to a single customer. Other instances for that customer can be hosted on the same hardware.
Option C is incorrect. With Spot Instances, you can take advantage of unused EC2 capacity in AWS. Spot Instances are available at up to a 90% discount compared to on-demand instance pricing.
And Option D is also incorrect. Reserved Instances provide you with significant savings on your EC2 costs compared to on-demand instance pricing.
However, Reserved Instances are not hosted on a physical server, so that makes Option A correct.
That's all for this question.
Be sure to take note of any knowledge gaps that you may have identified while exploring this question, and let's get started with our eighth walkthrough question.
Walkthrough question 8
–

Let's get started with our eighth walkthrough question which is from task statement 4.2: Understand resources for billing, budget, and cost management.
The question reads, **what is an advantage of consolidated billing on AWS?**
Reading this question, can you identify any keywords or phrases and exactly what the question is asking? A few keywords I see are consolidated billing.
Consolidated billing is another keyword for an AWS service. Do you remember which one?
Now that we've examined the question, identified keywords, and reviewed the requirements, let's explore the responses.

- Option A, volume pricing qualification.
- Option B, shared access permissions.
- Option C, multiple bills for each account.
- And option D, elimination of the need to tag resources.

Pause the video if you need more time. Okay, let's evaluate the options.
Option A is volume pricing discounts. Consolidated billing is a feature of AWS Organizations and a keyword. You can combine the usage across all accounts in your organization to share volume pricing discounts, reserved instance discounts, and saving plans. This solution provides a lower cost compared to the use of individual standalone accounts. This makes option A a good candidate for the correct answer, but let's look at all of the responses.
Option B is incorrect. Shared access permissions is a feature of roles that are developed in AWS IAM. This solution is not related to consolidated billing.
Option C is incorrect. The goal of consolidated billing is to have one bill for multiple accounts.
And option D is incorrect. In consolidated billing, you can apply tags that represent business categories. This functionality helps you organize your cost across multiple services within consolidated billing.
So that makes option A the correct answer.
That's all for this question. Be sure to take note of any knowledge gaps that you may have identified while exploring this question, and let's wrap up this course in our next lesson.