# AWS Cloud Practitioner Essentials

## Module 4: Networking

### 21. Module 4 Introduction

A purple square with a white cloud and a shield

Description automatically generated

Public Subnet: Customers just can interact with Cashier.

A screenshot of a video game

Description automatically generated

Private Subnet: Customers can’t reach the Barista.

A couple of men standing in a room

Description automatically generated

### 22. Connectivity to AWS

#### Amazon Virtual Private Cloud (Amazon VPC)

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

A networking service that you can use to establish boundaries around your AWS resources is [**Amazon Virtual Private Cloud (Amazon VPC)**(opens in a new tab)](https://aws.amazon.com/vpc/).

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

#### Internet gateway

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

A screenshot of a computer

Description automatically generated

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?

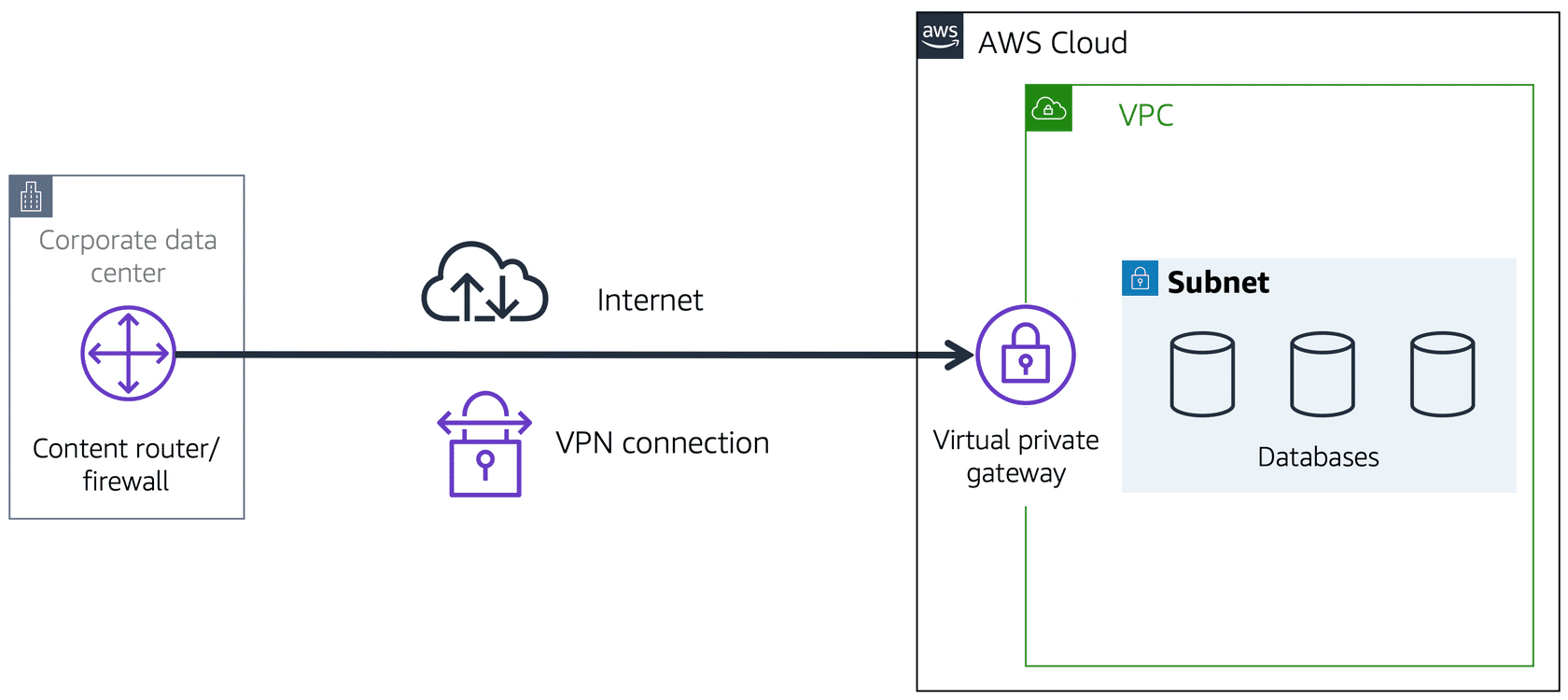
#### Virtual private gateway

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

[**AWS Direct Connect**(opens in a new tab)](https://aws.amazon.com/directconnect/) 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.

A diagram of a cloud server

Description automatically generated

A corporate data center routes network traffic to an AWS Direct Connect location. That traffic is then routed to a VPC through a virtual private gateway. All network traffic between the corporate data center and VPC flows through this dedicated private connection.

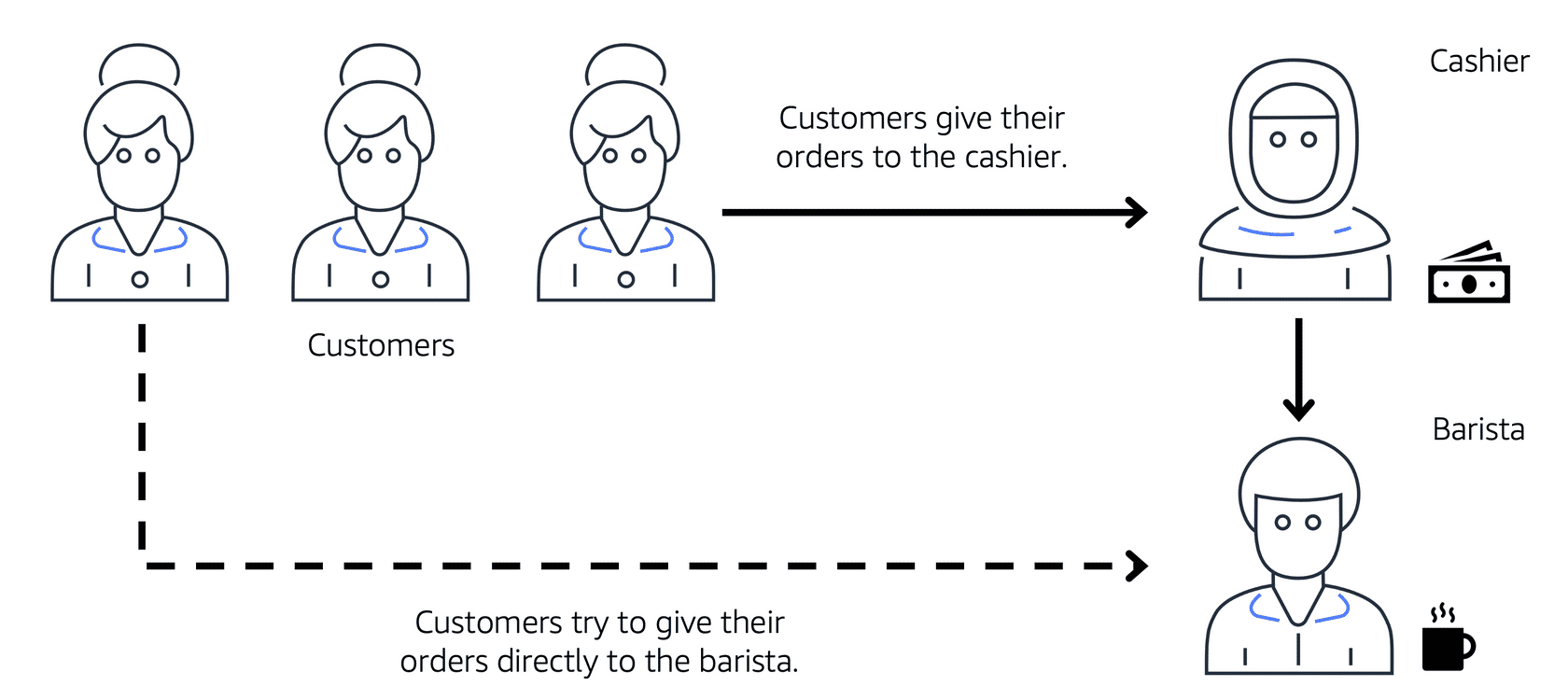
The private connection that AWS Direct Connect provides helps you to reduce network costs and increase the amount of bandwidth that can travel through your network.

### 23. Subnets and Network Access Control Lists

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

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

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



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

A diagram of a customer order

Description automatically generated

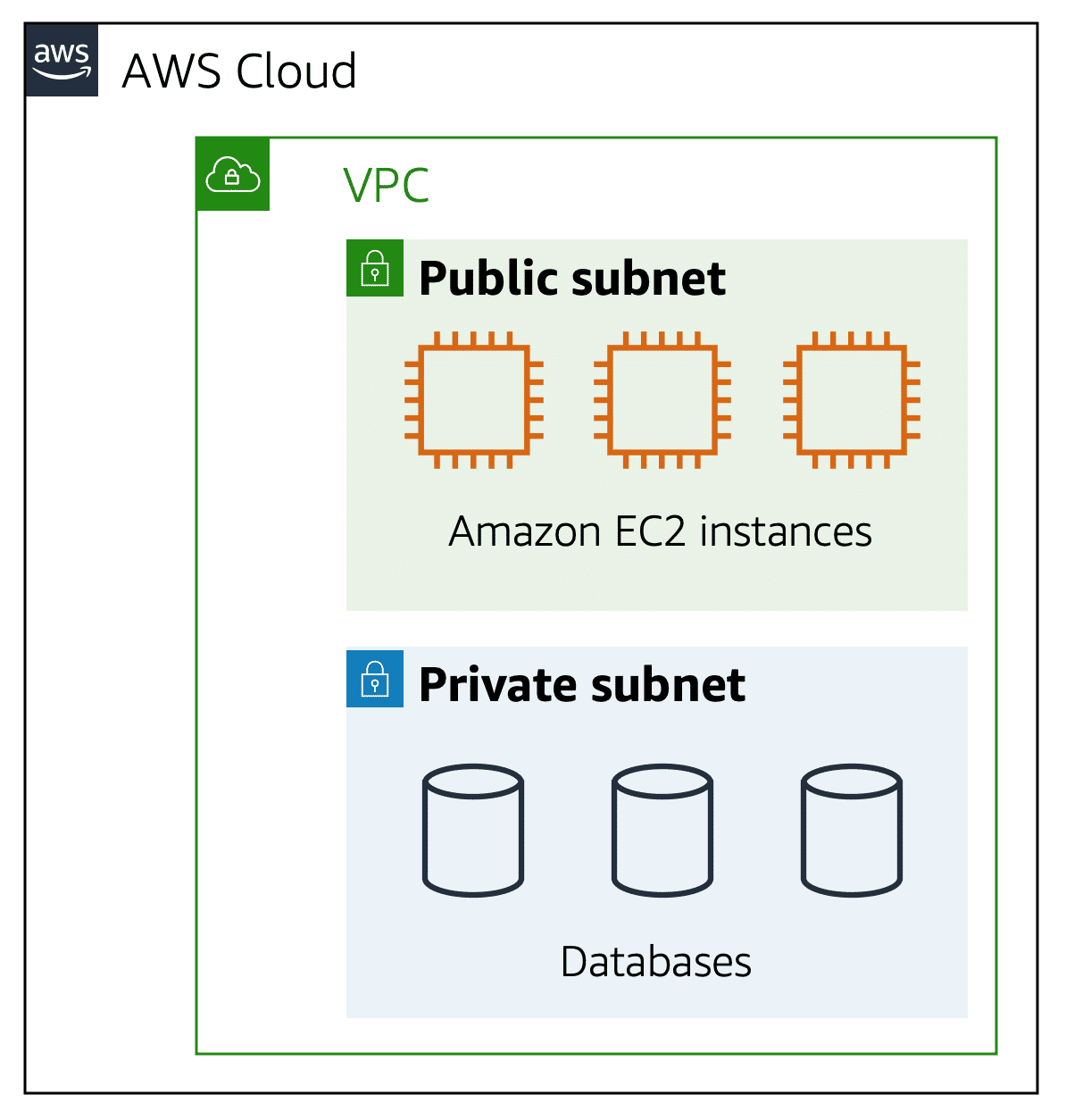
A cashier, a barista, and three customers in line. The icon for the first customer in line has an arrow pointing to cashier showing that the customer gives their order to the cashier. Then the cashier icon has an arrow pointing to barista icon showing that the cashier forwards the customer's order to the barista. The last customer in line tries to give their order directly to the barista, but they're blocked from doing so.

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

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

#### Subnets

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



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

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

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

#### Network traffic in a VPC

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

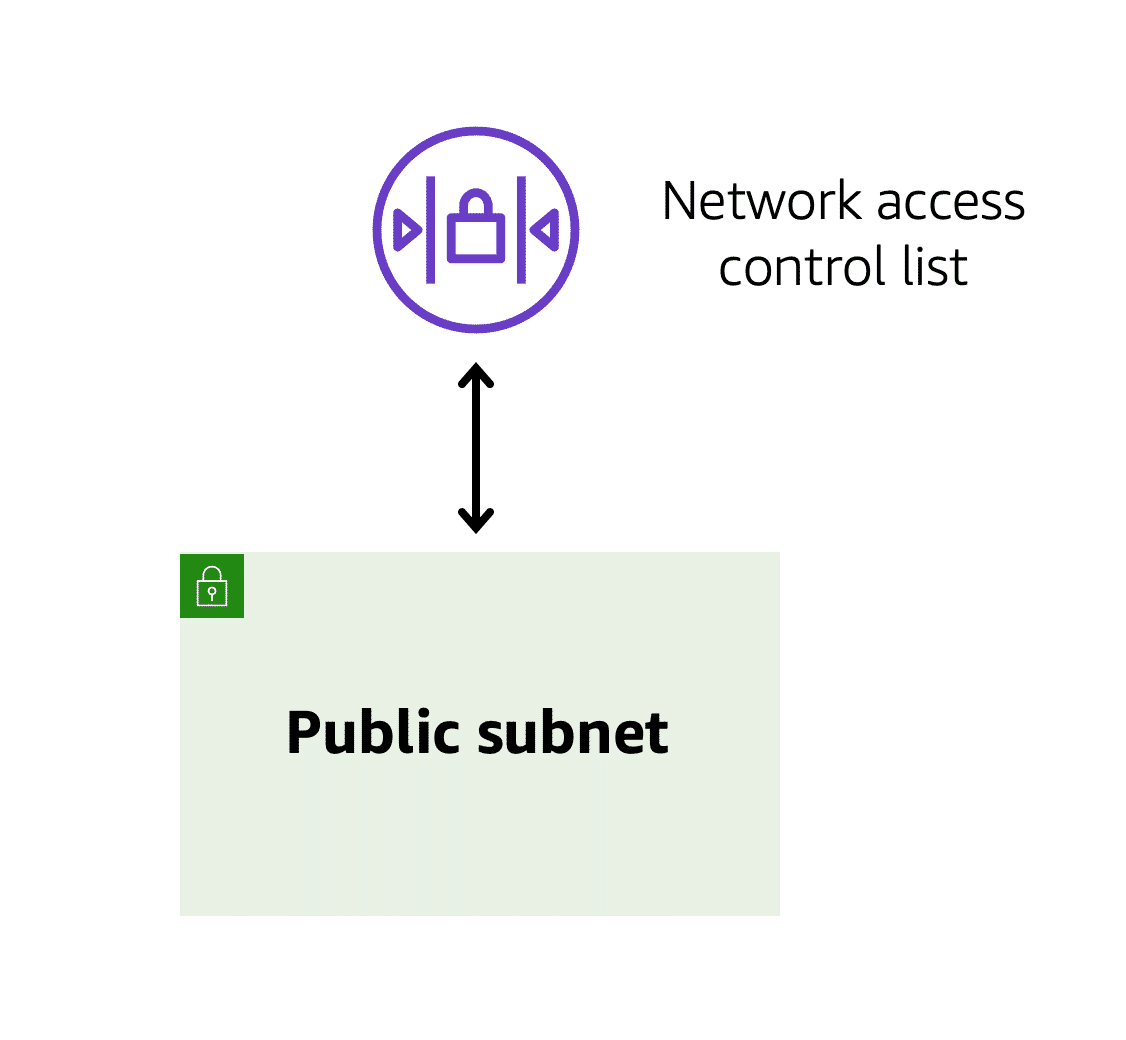
It enters into a VPC through an internet gateway. Before a packet can enter into a subnet or exit from a subnet, it checks for permissions. These permissions indicate who sent the packet and how the packet is trying to communicate with the resources in a subnet.

The VPC component that checks packet permissions for subnets is a [**network access control list (ACL)**(opens in a new tab)](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html).

#### Network ACLs

A network ACL is a virtual firewall that controls inbound and outbound traffic at the subnet level.

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



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

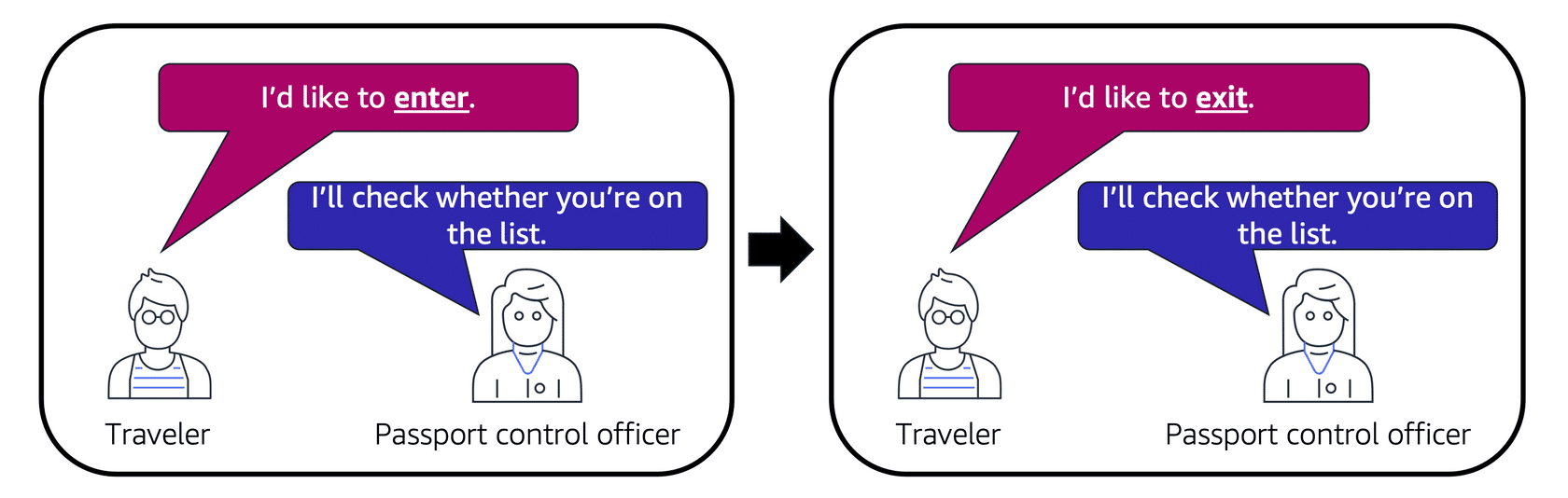
By default, your account’s default network ACL allows all inbound and outbound traffic, but you can modify it by adding your own rules. For custom network ACLs, all inbound and outbound traffic is denied until you add rules to specify which traffic to allow. Additionally, all network ACLs have an explicit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

#### Stateless packet filtering

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

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

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

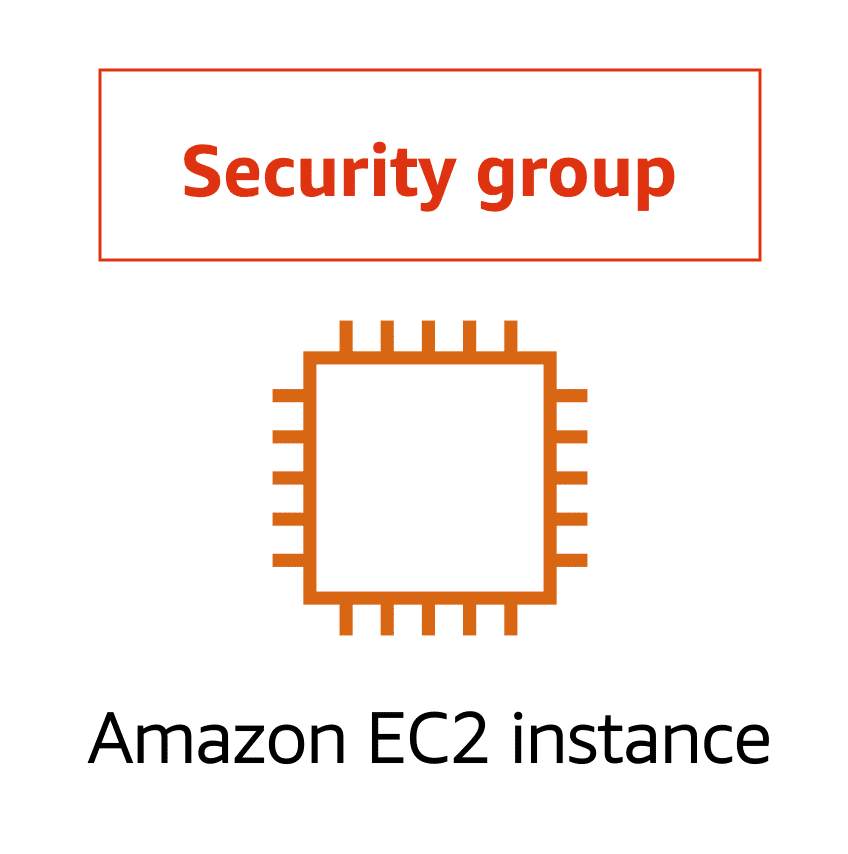


After a packet has entered a subnet, it must have its permissions evaluated for resources within the subnet, such as Amazon EC2 instances.

The VPC component that checks packet permissions for an Amazon EC2 instance is a [**security group**(opens in a new tab)](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html).

#### 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

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

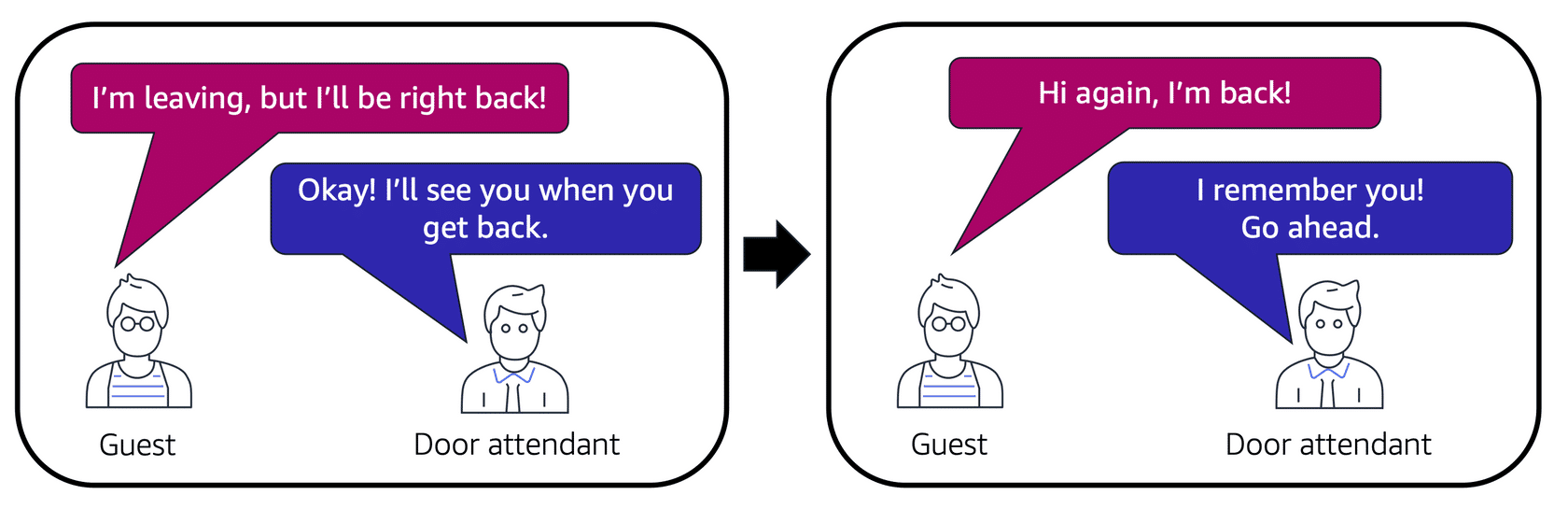
If you have multiple Amazon EC2 instances within the same VPC, you can associate them with the same security group or use different security groups for each instance.

#### Stateful packet filtering

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

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

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



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

A diagram of a network connection

Description automatically generated

A packet travels over the internet from a client, to the internet gateway and into the VPC. Then the pack goes through the network access control list and accesses the public subnet, where two EC2 instances are located.

Private subnet: Isolate databases containing customers' personal information.

Virtual private gateway : Create a VPN connection between the VPC and the internal corporate network.

Public subnet: Support the customer-facing website.

AWS Direct Connect: Establish a dedicated connection between the on-premises data center and the VPC.

### 24. 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.

A diagram of a computer

Description automatically generated

A client connects to a DNS resolver looking for a domain. The resolver forwards the request to the DNS server, which returns the IP address to the resolver.

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

* 1

1

When you enter the domain name into your browser, this request is sent to a customer DNS resolver.

* 2

2

The customer DNS resolver asks the company DNS server for the IP address that corresponds to AnyCompany’s website.

* 3

3

The company DNS server responds by providing the IP address for AnyCompany’s website, 192.0.2.0.

#### Amazon Route 53

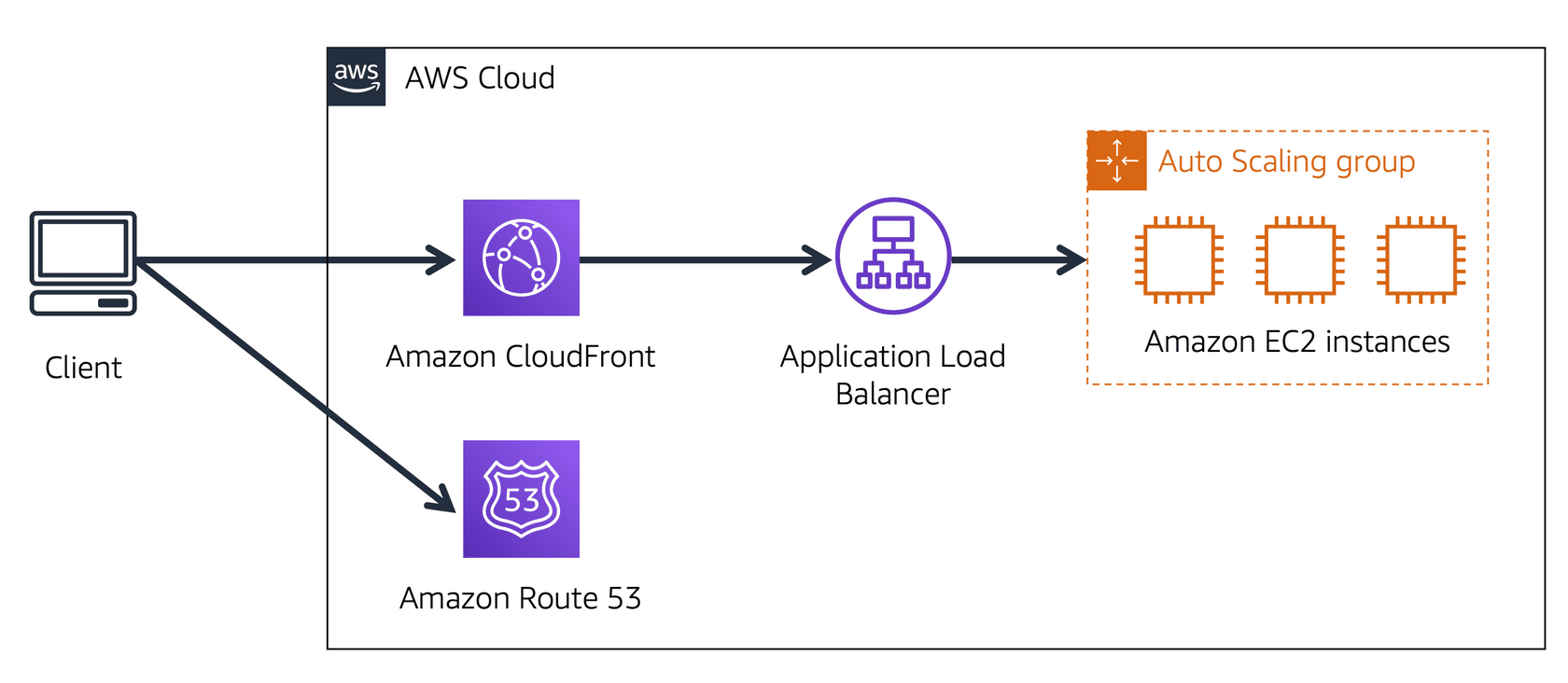
[**Amazon Route 53**(opens in a new tab)](https://aws.amazon.com/route53) is a DNS web service. It gives developers and businesses a reliable way to route end users to internet applications hosted in AWS.

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

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

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

#### Example: How Amazon Route 53 and Amazon CloudFront deliver content



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

* 1

1

A customer requests data from the application by going to AnyCompany’s website.

* 2

2

Amazon Route 53 uses DNS resolution to identify AnyCompany.com’s corresponding IP address, 192.0.2.0. This information is sent back to the customer.

* 3

3

The customer’s request is sent to the nearest edge location through Amazon CloudFront.

* 4

4

Amazon CloudFront connects to the Application Load Balancer, which sends the incoming packet to an Amazon EC2 instance.

#### Knowledge check

For guidance on navigating this question using the keyboard, expand the following keyboard instructions.

## Module 5: STORAGE AND DATABASES

### Introduction

### 28. Instance Stores and Amazon Elastic Block Store (Amazon EBS)

#### **Instance stores**

Block-level storage volumes behave like physical hard drives.

An [**instance store**(opens in a new tab)](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html) 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.

To review an example of how instance stores work, choose the arrow buttons to display each step.

A computer screen shot of a computer

Description automatically generatedA computer screen shot of a computer

Description automatically generated

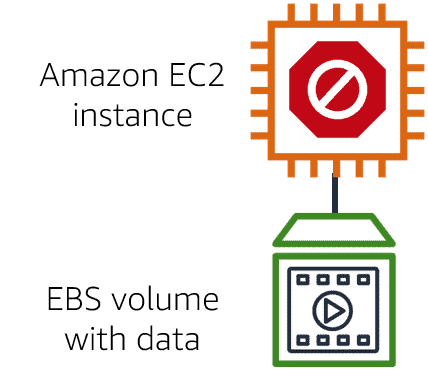
A diagram of a computer

Description automatically generated

[**Amazon Elastic Block Store (Amazon EBS)**(opens in a new tab)](https://aws.amazon.com/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

A screenshot of a computer

Description automatically generated

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**(opens in a new tab)](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html) 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.

Which of the following are characteristics of the Amazon EBS service? (Select TWO.)

* Best for data that requires retention
* Separate drives from the host computer of an EC2 instance

### 29. Amazon Simple Storage Service (Amazon S3)

#### Object storage

A computer screen shot of a key

Description automatically generated

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

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

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

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

[**Amazon Simple Storage Service (Amazon S3)**(opens in a new tab)](https://aws.amazon.com/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(opens in a new tab)](https://aws.amazon.com/s3/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 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.

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

–

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

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

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

–

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

Compared to 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.

**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.

**S3 Glacier Instant Retrieval**

–

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

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

**S3 Glacier Flexible Retrieval**

–

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

S3 Glacier Flexible Retrieval is a low-cost storage class that is ideal for data archiving. For example, you might use this storage class to store archived customer records or older photos and video files. You can retrieve your data from S3 Glacier Flexible Retrieval from 1 minute to 12 hours.

**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.

**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.

### 30. 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)**(opens in a new tab)](https://aws.amazon.com/efs/) is a scalable file system used with AWS Cloud services and on-premises resources. As you add and remove files, Amazon EFS grows and shrinks automatically. It can scale on demand to petabytes without disrupting applications.

#### **Comparing Amazon EBS and Amazon EFS**

Select each of the cards below to review a comparison of 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.

### 31. Amazon Relational Database Service (Amazon RDS)

**Relational databases**

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

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

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

Example of data in a relational database:

| **ID** | **Product name** | **Size** | **Price** |
| --- | --- | --- | --- |
| 1 | Medium roast ground coffee | 12 oz. | $5.30 |
| 2 | Dark roast ground coffee | 20 oz. | $9.27 |

**Amazon Relational Database Service**

[**Amazon Relational Database Service (Amazon RDS)**(opens in a new tab)](https://aws.amazon.com/rds/) is a service that enables you to run relational databases in the AWS Cloud.

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

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

**Amazon RDS database engines**

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

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

**Amazon Aurora**

[**Amazon Aurora**(opens in a new tab)](https://aws.amazon.com/rds/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.

### 32. Amazon DynamoDB

## Module 6: SECURITY

### 39. AWS Shared Responsibility Model

### 40. User Permissions and Access

#### **AWS Identity and Access Management (IAM)**

[**AWS Identity and Access Management (IAM)**(opens in a new tab)](https://aws.amazon.com/iam/) enables you to manage access to AWS services and resources securely.

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

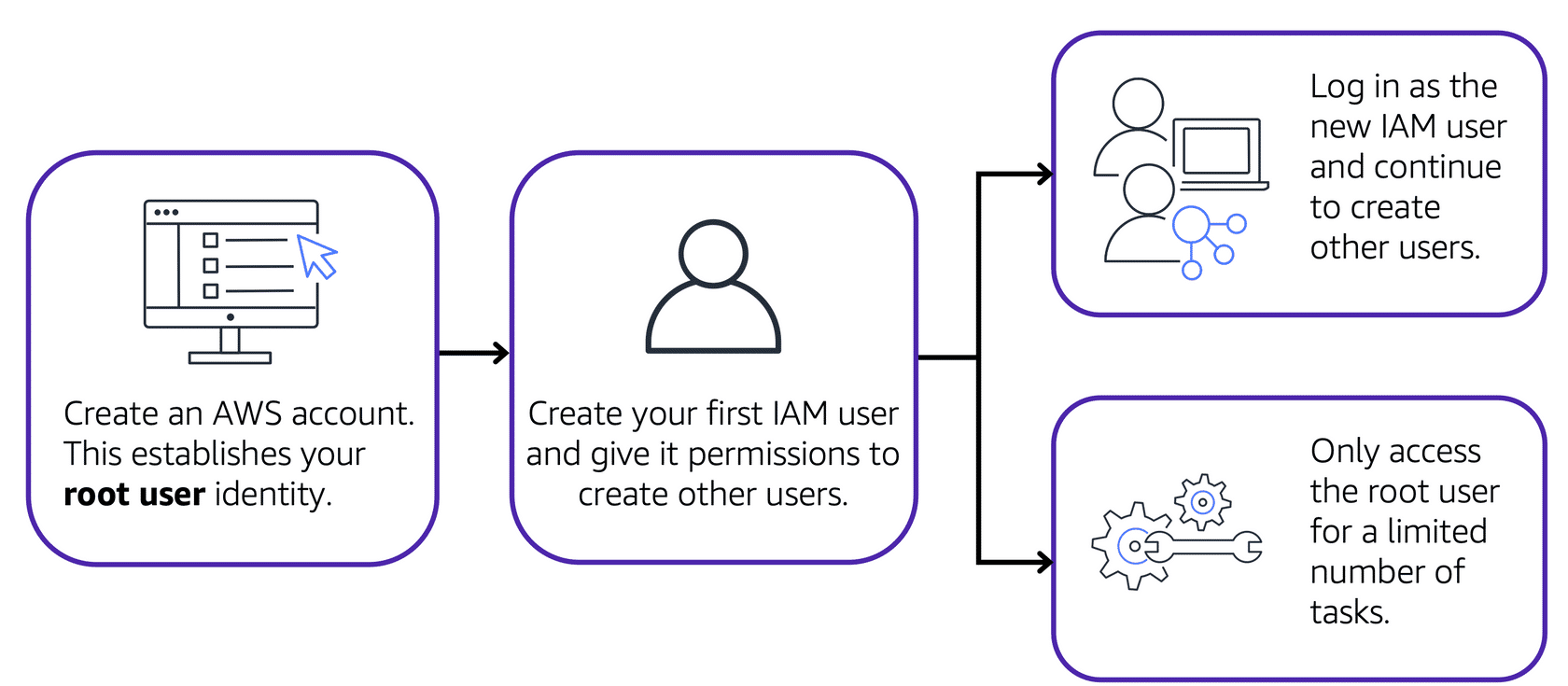
* IAM users, groups, and roles
* IAM policies
* Multi-factor authentication

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

#### **AWS account root user**

When you first create an AWS account, you begin with an identity known as the [**root user**(opens in a new tab)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html).

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



Best practice:

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

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

Then, continue to create other IAM users, and access those identities for performing regular tasks throughout AWS. Only use the root user when you need to perform a limited number of tasks that are only available to the root user. Examples of these tasks include changing your root user email address and changing your AWS support plan. For more information, see “Tasks that require root user credentials” in the [AWS Account Management Reference Guide(opens in a new tab)](https://docs.aws.amazon.com/accounts/latest/reference/root-user-tasks.html).

#### **IAM users**

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

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

Best practice:

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

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

#### **IAM policies**

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

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

Best practice:

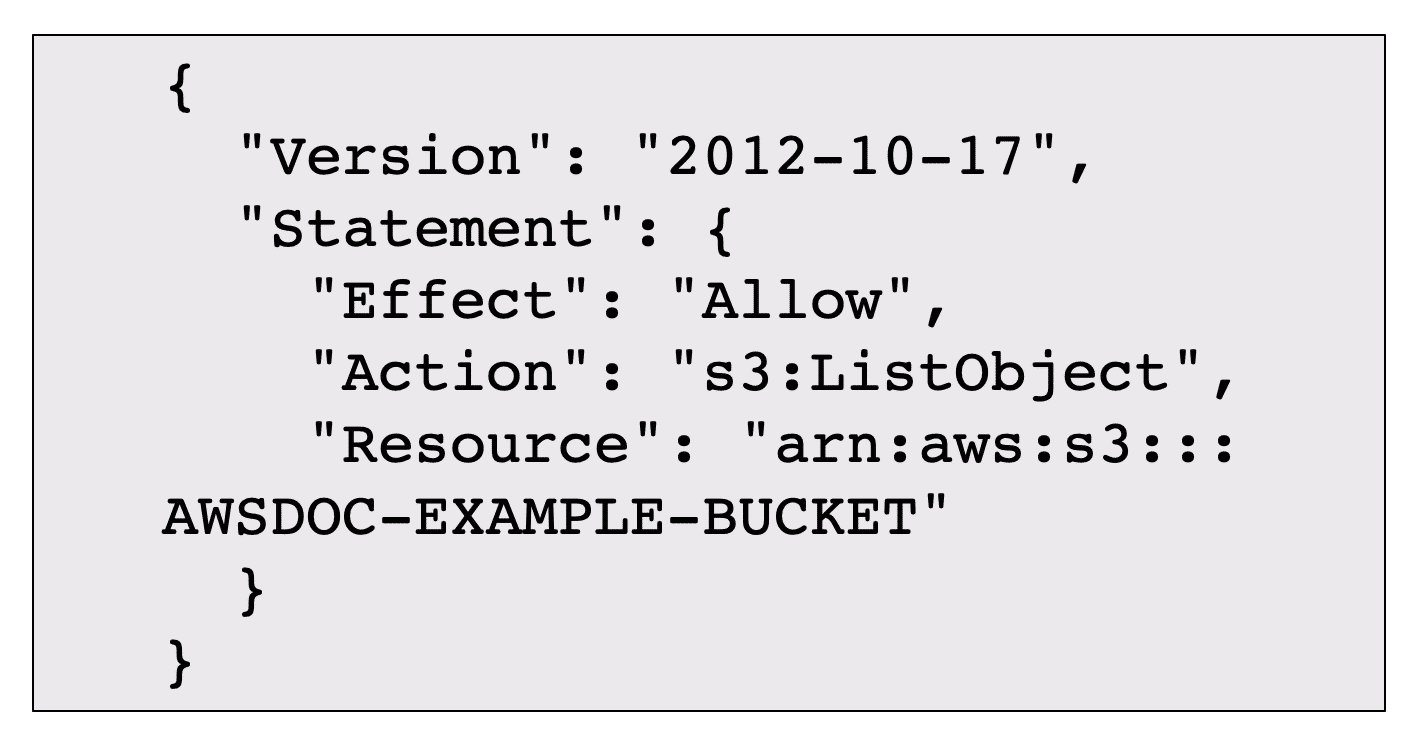
Follow the security principle of **least privilege** when granting permissions.

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

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

#### **Example: IAM policy**

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



This example IAM policy allows permission to access the objects in the Amazon S3 bucket with ID: AWSDOC-EXAMPLE-BUCKET.

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

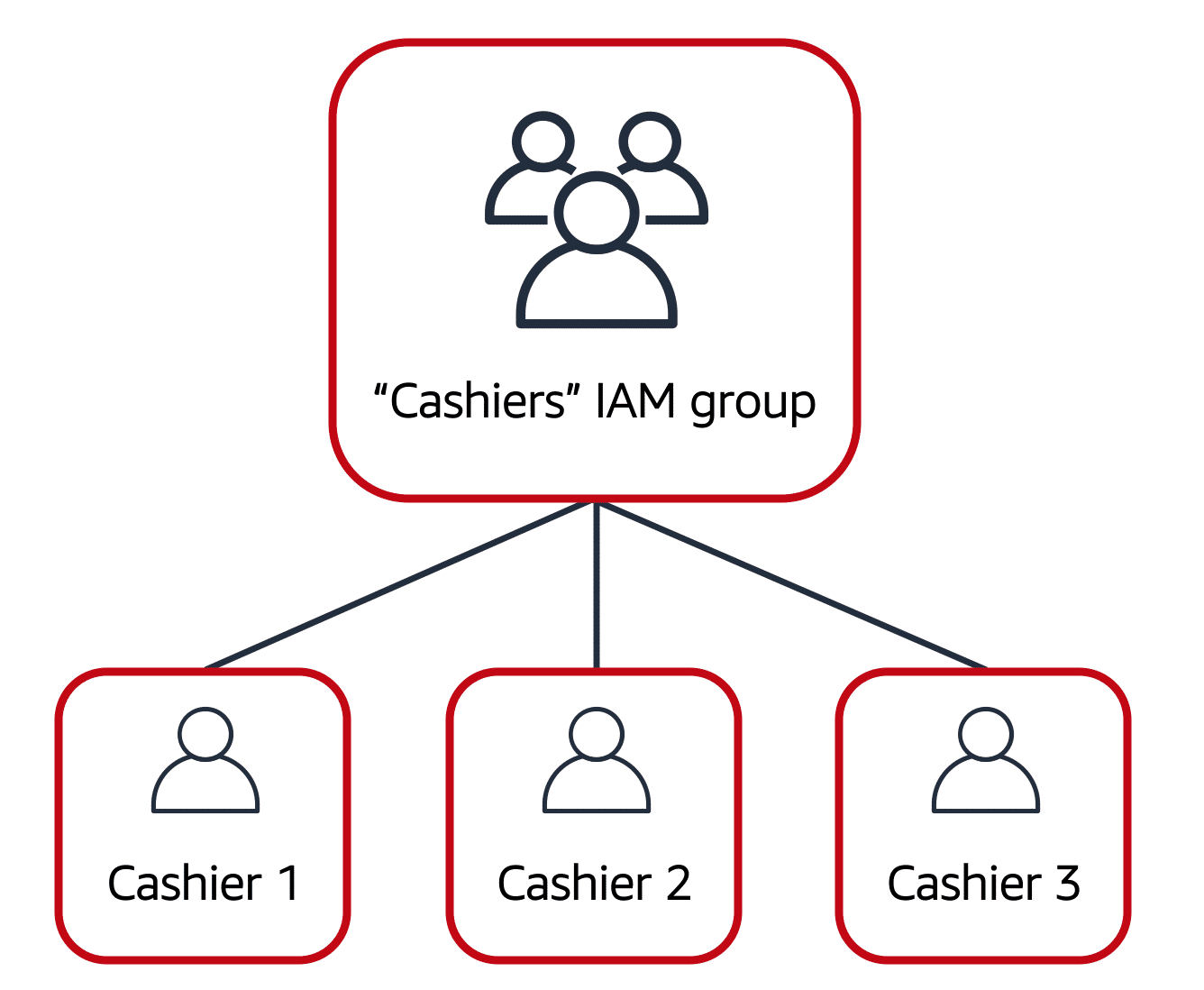
If the owner wants the cashier to be able to access other services and perform other actions in AWS, the owner must attach additional policies to specify these services and actions.

Now, suppose that the coffee shop has hired a few more cashiers. Instead of assigning permissions to each individual IAM user, the owner places the users into an [**IAM group**(opens in a new tab)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html).

#### **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**(opens in a new tab)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html).

#### **IAM roles**

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

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

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

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

Best practice:

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

To review an example of how IAM roles could be used in the coffee shop example, choose the arrow buttons to display each of the following two steps.

### 41. AWS Organizations