Introduction

Amazon Virtual Private Cloud, or VPCs, as they're affectionately known. A VPC lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. These resources can be public facing so they have access to the internet, or private with no internet access, usually for backend services like databases or application servers. The public and private grouping of resources are known as subnets and they are ranges of IP addresses in your VPC.

Now, in our coffee shop, we have different employees and one is a cashier. They take customers' orders and thus we want customers to interact with them, so we put them in what we call a public subnet. Hence they can talk to the customers or the internet. But for our baristas, we want them to focus on making coffee and not interact with customers directly, so we put them in a private subnet.

Connectivity to AWS

A VPC, or Virtual Private Cloud, is essentially your own private network in AWS. A VPC allows you to define your private IP range for your AWS resources, and you place things like EC2 instances and ELBs inside of your VPC.

Now you don't just go throw your resources into a VPC and move on. You place them into different subnets. Subnets are chunks of IP addresses in your VPC that allow you to group resources together. Subnets, along with networking rules we will cover later, control whether resources are either publicly or privately available. There are actually ways you can control what traffic gets into your VPC at all. What I mean by this is, for some VPCs, you might have internet-facing resources that the public should be able to reach, like a public website, for example.

However, in other scenarios, you might have resources that you only want to be reachable if someone is logged into your private network. This might be internal services, like an HR application or a backend database.

Public facing resources: -

First, let's talk about public-facing resources. In order to allow traffic from the public internet to flow into and out of your VPC, you must attach what is called an internet gateway, or IGW, to your VPC.

An internet gateway is like a doorway that is open to the public. Think of our coffee shop. Without a front door, the customers couldn't get in and order their coffee, so we install a front door and the people can then go in and out of that door when coming and going from our shop. The front door in this example is like an internet gateway. Without it, no one can reach the resources placed inside of your VPC.

Private resources: -

let's talk about a VPC with all internal private resources. We don't want just anyone from anywhere to be able to reach these resources. So we don't want an internet gateway attached to our VPC. Instead, we want a private gateway that only allows people in if they are coming from an approved network, not the public internet. This private doorway is called a virtual private gateway, and it allows you to create a VPN connection between a private network, like your on-premises data center or internal corporate network to your VPC.

To relate this back to the coffee shop, this would be like having a private bus route going from my building to the coffee shop. If I want to go get coffee, I first must badge into the building, thus authenticating my identity, and then I can take the secret bus route to the internal coffee shop that only people from my building can use. So if you want to establish an encrypted VPN connection to your private internal AWS resources, you would need to attach a virtual private gateway to your VPC.

Now the problem with our super secret bus route is that it still uses the open road. It's susceptible to traffic jams and slowdowns caused by the rest of the world going about their business. The same thing is true for VPN connections. They are private and they are encrypted, but they still use a regular internet connection that has bandwidth that is being shared by many people using the internet.

The point being is you still want a private connection, but you want it to be dedicated and shared with no one else. You want the lowest amount of latency possible with the highest amount of security possible.

With AWS, you can achieve that using what is called AWS Direct Connect. Direct Connect allows you to establish a completely private, dedicated fiber connection from your data center to AWS. You work with a Direct Connect partner in your area to establish this connection, because like my magic doorway, AWS Direct Connect provides a physical line that connects your network to your AWS VPC. This can help you meet high regulatory and compliance needs, as well as sidestep any potential bandwidth issues. It's also important to note that one VPC might have multiple types of gateways attached for multiple types of resources all residing in the same VPC, just in different subnets.

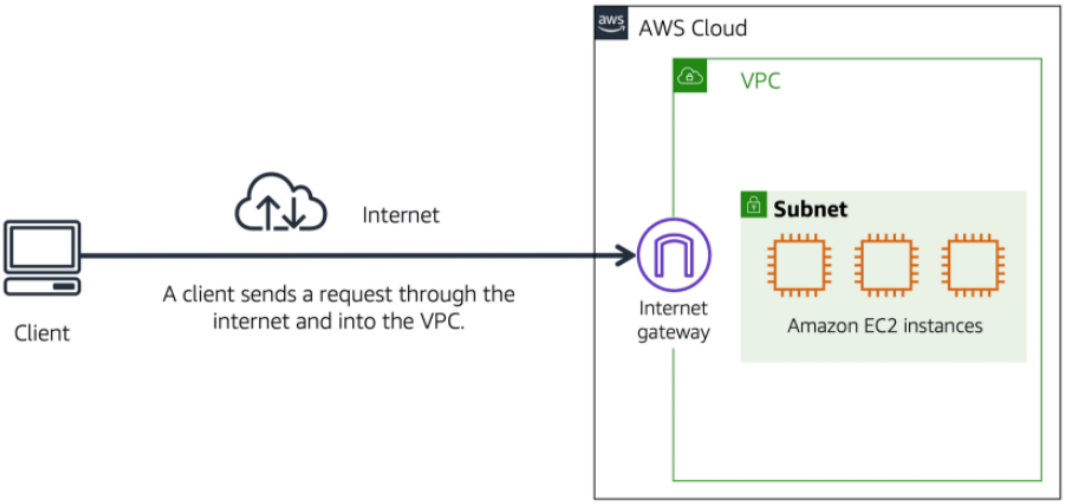
Amazon Virtual Private Cloud (Amazon VPC)

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

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

Internet gateway

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



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

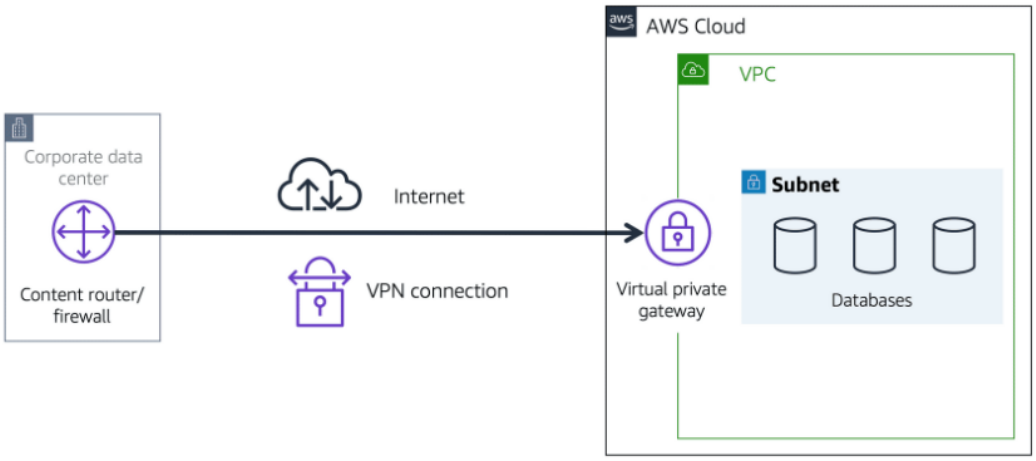
Virtual private gateway

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.



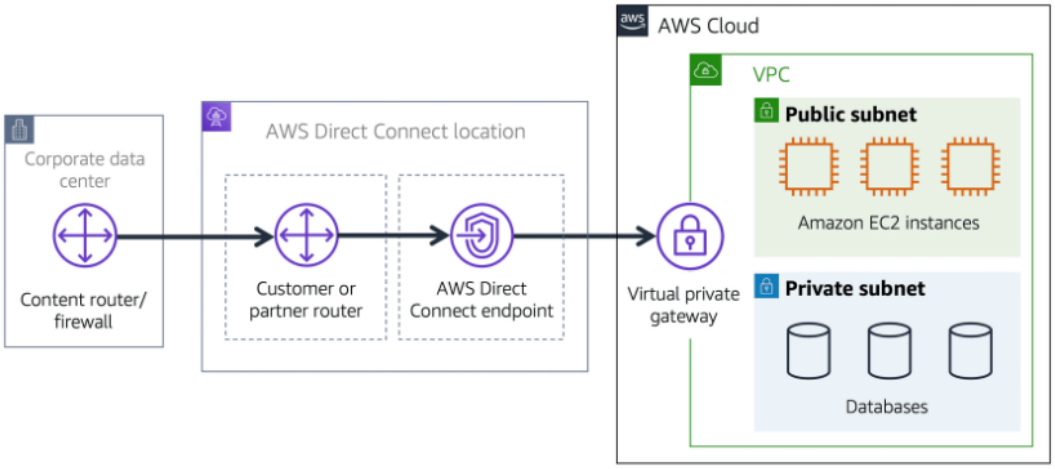
A virtual private gateway enables you to establish a virtual private network (VPN) connection between your VPC and a private network, such as an on-premises data center or internal corporate network. A virtual private gateway allows traffic into the VPC only if it is coming from an approved network.

AWS Direct Connect

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

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

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



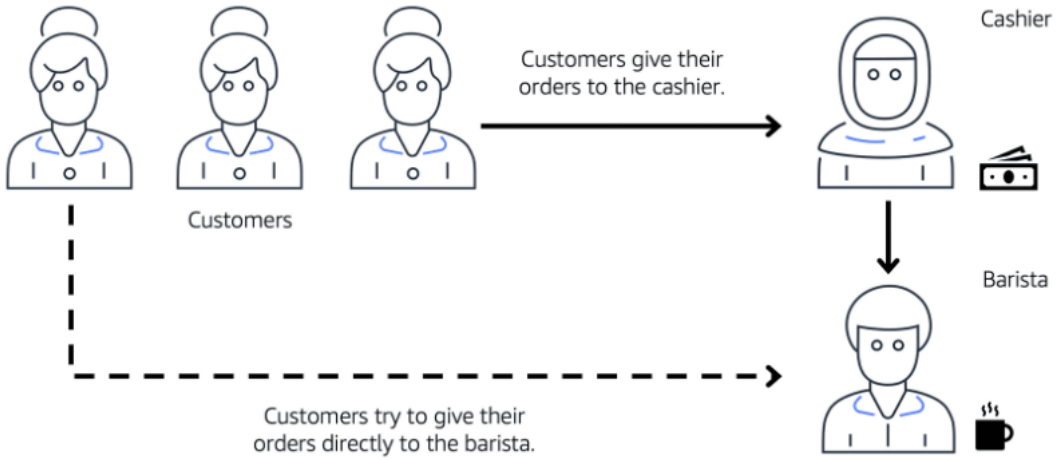
The private connection that AWS Direct Connect provides helps you to reduce network costs and increase the amount of bandwidth that can travel through your network.

Subnets and network access control lists

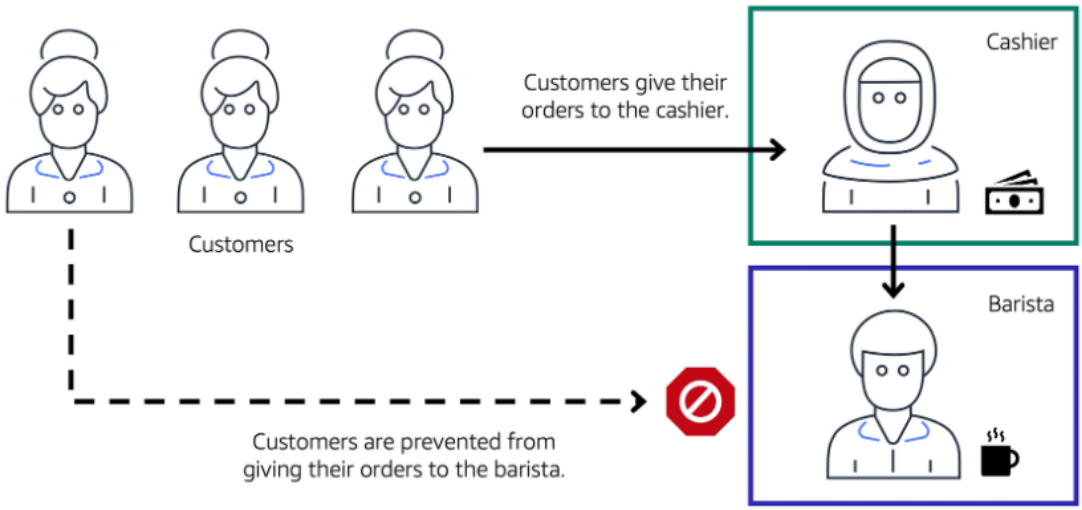
To learn more about the role of subnets within a VPC, review the following example from the coffee shop.

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

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



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

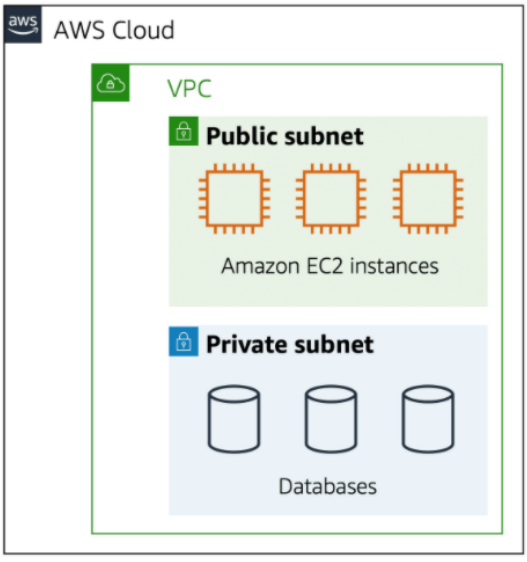


This is similar to how you can use AWS networking services to isolate resources and determine exactly how network traffic flows.

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

Subnets

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



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

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

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

Network traffic in a VPC

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

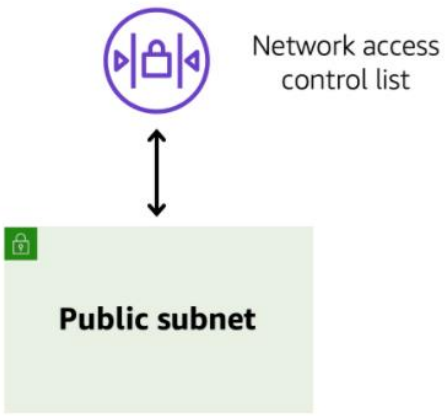
It enters into a VPC through an internet gateway. Before a packet can enter into a subnet or exit from a subnet, it checks for permissions. These permissions indicate who sent the packet and how the packet is trying to communicate with the resources in a subnet.

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

Network access control lists (ACLs)

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

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



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

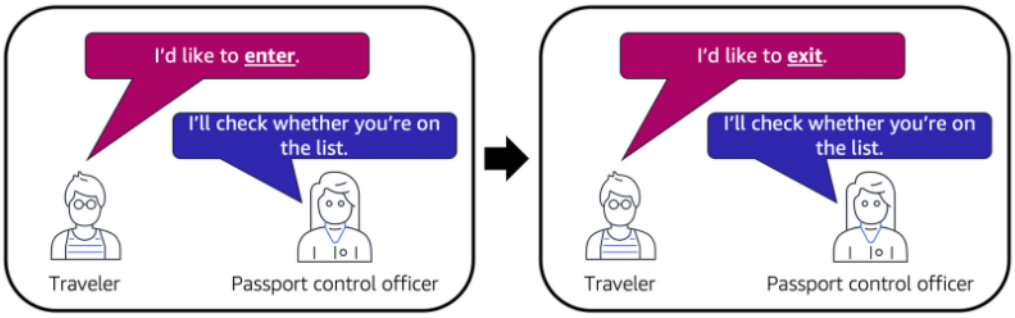
By default, your account's default network ACL allows all inbound and outbound traffic, but you can modify it by adding your own rules. For custom network ACLs, all inbound and outbound traffic is denied until you add rules to specify which traffic to allow. Additionally, all network ACLs have an explicit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

Stateless packet filtering

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

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

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



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

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

Security groups

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

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

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

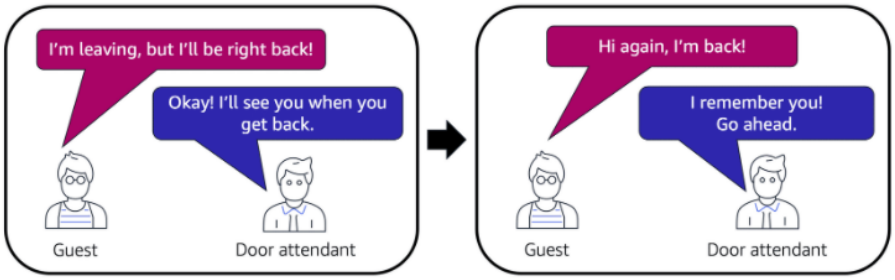
If you have multiple Amazon EC2 instances within a subnet, you can associate them with the same security group or use different security groups for each instance.

Stateful packet filtering

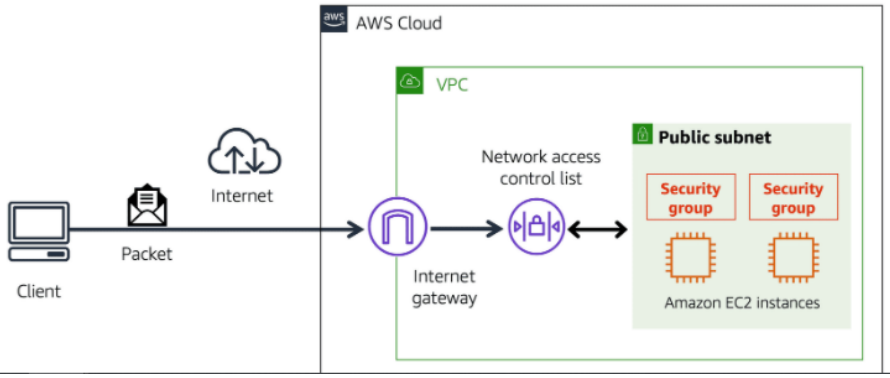
Security groups perform **stateful** packet filtering. They remember previous decisions made for incoming packets.

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

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



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

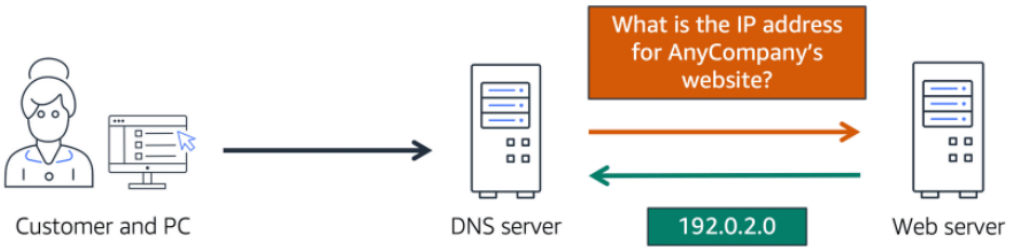


Global Networking

Domain Name System (DNS)

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

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



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

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

Amazon Route 53

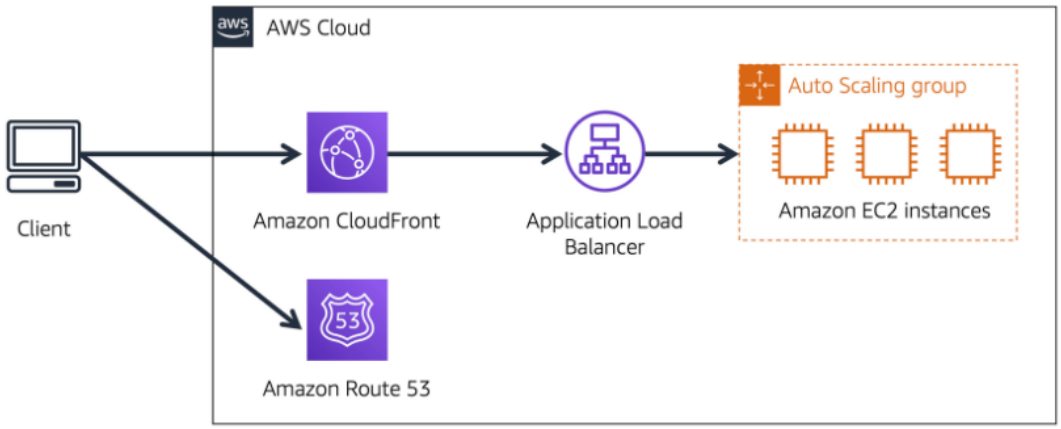
Amazon Route 53 is a DNS web service. It gives developers and businesses a reliable way to route end users to internet applications hosted in AWS.

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

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

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

Example: How Amazon Route 53 and Amazon CloudFront deliver content



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

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