Amazon Virtual Private Cloud

Amazon Virtual Private Cloud 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). 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. 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. 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. 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. Subnets are chunks of IP addresses in your VPC that allow you to group resources together. 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.

Internet gateway

To allow public traffic from the internet to access your VPC, you attach an internet gateway (IGW) to the VPC. You can think of it as a hardened fortress where nothing goes in or out without explicit permission. You have a gateway on the VPC that only permits traffic in or out of 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

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

For example, 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. This means we want a private gateway that only allows people in if they are **coming from an approved network**, not the public internet.

To access private resources in a VPC, you can use a **virtual private gateway**, which is like a private doorway. It allows you to create a **VPN connection between a private network, like your on-premises data centre and internal corporate network to your VPC**.

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. Although VPN connections 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.

AWS Direct Connect

What if 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. **AWS Direct Connect** is a service that enables you to establish a **dedicated private connection between your data centre 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.

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**.

Subnets

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**.

Domain Name System (DNS)

We've been talking a lot about how you interact with your AWS infrastructure. But how do your customers interact with your AWS infrastructure?

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.

Think of DNS as a translation service. But instead of translating between languages, it translates website names into IP, or Internet Protocol, addresses that computers can read.

**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.

If we go further, Route 53 can **direct traffic to different endpoints** using several different routing policies, such as **latency-based routing, geolocation DNS, geoproximity, and weighted round robin**. If we take geolocation DNS, that means we direct traffic based on where the customer is located. So traffic coming from say North America is routed to the Oregon Region, and traffic in Ireland is routed to the Dublin Region, as an example.

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, so you can buy and manage your own domain names right on AWS. 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.

Speaking of websites, there is another service which can help speed up delivery of website assets to customers, **Amazon CloudFront**. In the previous module, you learned about Amazon CloudFront, a content delivery service. If you remember, we talked about Edge locations earlier in the course, these locations are serving content as close to customers as

possible, and one part of that, is the content delivery network, or CDN. For those who need reminding, a CDN is a network that helps to deliver edge content to users based on their geographic location.

The following example describes how Route 53 and Amazon CloudFront work together to deliver content to customers.

**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.