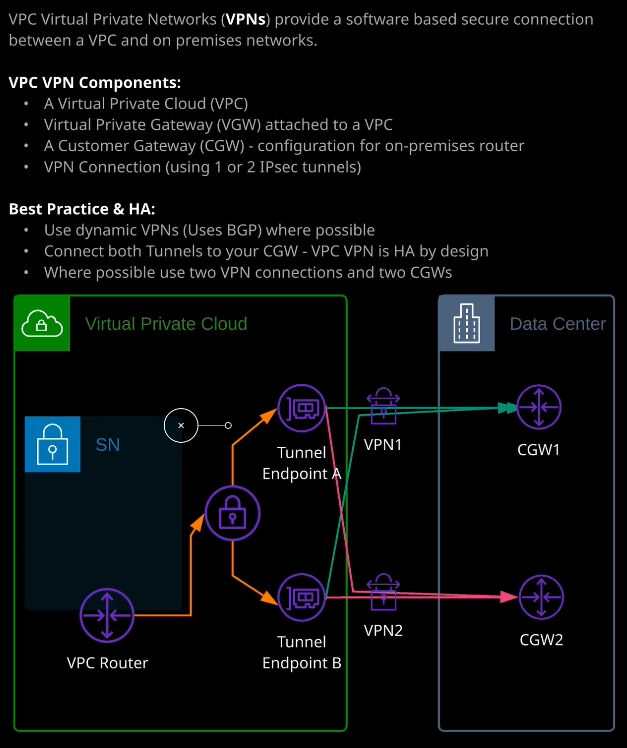
connectivity options between on premises networks and your VPCs. In this lesson, I'm going to be talking about **AWS VPN which is also known as hardware VPNs**, and it's **a product set that allows you to configure a hardware based, highly available VPN connection between your VPC and on premises networks and VPNs use the public internet as a transit mechanism between your on premises networks and your VPCs.**

Now **AWS VPN provide a fully encrypted transit path across the internet from wherever your premises are located to your VPCs**. **Now the VPN tunnels operate over IP version four and allow you to access any of your remote VPC networks from on premises networks and vice versa**. In this lesson, I'm going to step you through the architecture of AWS VPNs as well as compare and contrast situations where you might use a VPN compared to other AWS connection technologies such as Direct Connect, because that's always something that tends to come up in the exam as well as understanding the architecture of VPNs, you also need to understand when and where to use them.   


So let's get started and first I want to step through the different components that make up a VPN connection between your on premises networks and the VPC and for a demonstration, I'm going to connect a VPC that I've got running in my production AWS environment to my home network. So in order to do that, we're going to go to VPC. Now the first component that I want to introduce is what's known as a **customer gateway. Now, a customer gateway represents a physical piece of hardware at the customer side. It's generally a home, business, or enterprise router, and it's a router that's capable of IP sec VPN connectivity using either static or dynamic routing** and I'll step you through what that means very shortly. Now the **customer gateway object inside AWS is a logical representation of the hardware that lives on premise**. So first we need to configure one or more customer gateways. Now you'll generally **create a customer gateway object for every single customer gateway router that you physically have on premise.** So I'm going to do that for my home internet connection. I'm going to call the customer gateway Home. I'm going to elect to pick static routing, and I'll explain what that means exactly in a second and then I'll need to put in my static IP address for my home internet connection. So I'll enter that and then click on Create Gateway. **Now for customer gateways, you need to tell AWS whether you need to use static routing and static routing is the simplest form of routing within AWS VPNs. It's essentially the routing method where you simply have to tell either side of the VPN connection what subnets are available at the remote end of the connection**. So at my customer gateway side I need to add the VPC CIDR within my VPC and then the VPC side I need to provide AWS with details of my home network IP address range. If you **utilize dynamic customer gateways, then you're able to exchange that information dynamically over BGP, which is a routing protocol. So if you do utilize dynamic VPNs and both sides of the relationship can exchange IP addressing information for what subnets are at both sides of the VPN tunnels**.

So generally, if you're rolling this out in production, you'll try to aim to use dynamic VPNs and if you do use dynamic VPNs, you'll need to provide a BGP autonomous system number or ASN and that's essentially a unique identifier for the BGP router at either side of the relationship. Now, if you don't have a BGP ASN allocated to your network, then you can use a private ASN and the private ASNs occupy the range of 64512 to 65534. So now that we've got the customer gateway configured, the next step is to create a virtual private gateway and the **virtual private gateway is a gateway entity, much like an internet gateway and a NAT gateway and an egress only gateway**, it's something that the VPC router is aware of and the VPC router via route tables can direct traffic towards I'm going to go ahead and create a virtual private gateway. I'm going to call this prod because it's my production account and then I'll click on create virtual private gateway. Now once this is created you've got to attach it to a specific VPC. So I'm going to select VPC1 and say attach. Now that process can take a number of minutes. **A virtual private gateway can only be attached to a single VPC but a virtual private gateway can act as the termination point for many different VPN connections. So VPN connections occur between the customer gateways and the virtual private gateway. So you need to create virtual private gateway, attach it to a VPC, and then the virtual private gateway acts as the endpoint for these VPN tunnels** and I'll explain what I mean by that very shortly. So now that this virtual private gateway is created and attached to the VPC, I need to go ahead and create the **site to site VPN connections**. Now, this is the **logical entity that links the virtual private gateway and the customer gateway** and you've got the option of a number of different configurations for these VPN connections. The first type of connection architecture and the one that I'm going to demonstrate in this demonstration is where you've got a **single tunnel between one of the tunnel endpoints of this virtual private gateway and a single customer gateway, and that doesn't provide any form of high availability. If this customer gateway fails, or if the tunnel endpoint that you're connected to fails, then you lose that VPN connection**. So it's not something that you want to try to aim for if you're utilizing VPNs in production. This is the architecture that I'm going to utilize to demonstrate the technology because I'm going to connect it between my home VPN router and a single tunnel endpoint but in production, you try to use something a little bit more resilient. **The second type of architecture is full resilience on the AWS side and this is where you create two tunnels from a single customer gateway. So you're connecting to a tunnel endpoints, and these total endpoints are located in different availability zones. So you've always got the ability to tolerate failure of a single availability zone at the AWS side** and then the third architecture is **full, high availability, and that's where you're creating VPN connections between two different customer gateways and that VPN connection itself has two tunnel to both of the tunnel endpoints as part of the virtual private gateway. So in this particular case, you can tolerate a failure of either of the two customer gateways and either of the two VPN tunnel endpoints. So in this particular case, you've got what's known as full resilience or full high availability** and you'll need to know that architecture for the exam because I've seen the question pop up many times where you're given a partially resilient VPN architecture and you're asked how you can make it fully highly available.

Now **using these dual tunnel designs does require that you're using dynamic VPNs so you need to be using BGP across these VPN connections**, so keep that in mind when you're looking at these exam questions, **you need to make sure you're using dynamic VPNs and you have connections to two different customer gateways on two different IP addresses so what you'll have is one virtual private gateway object, because that is highly available by design. You'll have two customer gateways representing two physical customer routers on premise, and then you'll create two VPN connections, the first between customer gateway one and that will have two tunnels one to each of the endpoints and you'll create a second VPN connection, which will be from customer gateway two and that will also have a single tunnel to each of the tunnel endpoints**. So in that particular case you'll have a total of **four active tunnels and you'll be running a fully resilient architecture**.

So now that you know the connection architecture I'm going to go ahead and create a VPN connection, I'm going to call it Prod-home to indicate that it's going between a prod environment and the home environment. I'll select a virtual private gateway. I'll use an existing customer gateway. I'll select static routing and provide the CIDR range of my home network. I'll accept the rest of the defaults and click on create VPN connection. Now while that's creating, I'm going to quickly hop over to EC2 and I'm going to create a test EC2 instance, I'm using the same VPC architecture as I've used throughout this section of the course, so I'm going to launch a small Linux instance it'll be in VPC1. I'm going to use subnet A3 because that's one of my private only subnets. I'll add storage, configure a security group, but this time I'm going to add a rule and allow all ICMP traffic from any source. That'll make it nice and easy to ping test once this instance is launched. So I launch it, choose an existing key pair, and allow that to create. While that is creating, I'm going to get the private IP address and switch over to my terminal. I'm going to start off a ping test and as expected, that's not running because I need to do further configuration to get this VPN connection up and running. Now I need to go back to VPC. All right, so the VPN connection has been created and the next thing we'll need to do is to download the configuration for this VPN connection. Now, AWS provide the password and authentication information and it could obtain that by going to download configuration and then I always change it to generic, and I'll click download to download that configuration file. Now the configuration file looks something like this. You'll have preshared key information. You'll be given all of the configuration settings that you'll need to set up the VPN and that's for both tunnels you'll be provided with the unique configuration for tunnel one and for tunnel two and because I'm creating a static VPN connection, I'll need all of this information in addition to the subnet information on both sides. So I'm going to go ahead and create the configuration on my home router for this connection. So I'll need to copy a preshared key. That's essentially the password or the authentication information so that both sides of the VPN connection both endpoints can authenticate the validity of the other party. So I'm going to switch over to my home network configuration. The configuration of this side depends on your home network. I'm using an a ubiquity system, so I'll need to go ahead and create a site to site VPN, select manual IP SEC. I'll need to add the remote subnets and I'm using the remote subnet VPC1. So that's 10.10.0.0/16. I'll select advanced options, and then I need to make sure that that configuration matches what's in here. Specifically, I already know that I need to change the Diffie-Hellman group configurations I'll need to go back, change that to group two and turn off dynamic routing. I'll need to make sure I got a preshared key, enter that on my router, and then I'll lead the peer IP address, which is the IP address of the virtual private gateway endpoint. So the endpoint of tunnel one so I'll need to go back and get that and this is the virtual private gateway IP address of the remote end of this VPN connection, at least from the perspective of my router. So I'll enter that. I'll also need the IP address of my local one IP. So enter that as well. I'll make sure I've got all this information correct, and then I'll give the connection of name and click on save. I'll also need to add a route because I'm using static routing so I'll create a new route. I'll call it aws-prod I'll need to enter the remote CIDR range of my VPC that's 10.10.10.0.0/16, distance one, interface, and then the AWS VPN. I'll go ahead and say that the rest of the configuration will occur from the AWS side. So I'll just do a refresh the state is now changed to available and so at some point, I should see one of these tunnels go from a downstate to an active or upstate. While I'm waiting for that, I'm going to go to route tables. I'll pick the route table of the subnet that the instance is in that I created moments ago. So go to routes, edit route, and then I add a static route for my home network. The destination will be virtual private gateway and I'll enter my virtual private gateway ID and click Save. Now I could enable route propagation on this subnet so if I go ahead and do that. So I'll propagate routes as well. Go back to routes and you'll know that I've got two routes on this route table for the same destination. One that's marked has propagated and one that's not marked is propagated, and that's important. I'll talk about those shortly. I'll go back to site to site VPN connections and just check on the status of these tunnels. There's often a lag between when the tunnel actually comes up when the status is updated in this console. So I'm just going to verify by switching to my command line and there we go, the ping is now working. So I've got full connectivity between my home network and AWS. So at some point, I want to keep refreshing. One of these tunnels will move into upstate. **It's just that there is a lag time between when it occurs on when it's updated in the UI.** Now you can see how quickly it wants to set up this VPN connection between my home network and AWS and that such a one of the benefits of utilizing VPN connections rather than Direct Connect.

As a professional solutions architect, you need to be aware of when to use VPN rather than Direct Connect and it comes down to a number of key points. **So VPN connections are quick to set up that can be done in minutes rather than the potentially weeks or months that it takes to install a Direct Connect**. **They're cheap, especially for sporadic or low usage.** **There's a per hour cost for running an active VPN connection as well as a data charge for any outgoing data**. **Now that data charge is higher than using a Direct Connect, so VPNs are generally economical when you've got lower requirements or lower data transfer requirements. The performance of VPN connections is generally limited by your customer gateway equipment. VPNs do use encryption end to end, and there is a compute performance overhead for using that encrypted tunnel**. So generally **the performance of VPN connections is limited both by the CPU capability of your customer router end to end internet performance. The performance of a VPN is fairly variable because it does use the internet for transit** **so you can experience latency and throughput issues, and both of those combined to control how much performance you can expect out of the VPN connection but generally, using a VPN connection versus a Direct Connect is cheaper and quicker to set up for all but the highest load**. Now, as mentioned at the start of the lesson.

You've got **three high availability profiles. You've got no high availability, so that's a single tunnel between a customer gateway and a virtual private gateway. You can utilize high availability AWS end so used to tools and then full high availability, which uses two customer gateways and the two tunnels on each of those.** Now I pointed out earlier how I had two routes on the route table once enabled route propagation and it's useful, I think, to understand how routing works and how it's influenced by VPN connections. So routing in general and route tables, **local route always takes priority**, even if you've got other routes that should, in theory, be more preferred than the local route, the local route, always takes priority. So for this 10.10.0.0/16 address in my case that is always preferred. The general rule is that in route tables any routes with the longest prefix so the highest value here are preferred. **If there are two routes with equal preference, like I have here then any static routes so non propagated routes such as the one I'm highlighting take preference**.   
So **static routes generally take preference over nonstatic routes so propagated routes**. So the priority for route tables overall is that **local is always the highest priority**, either most preferred. After that, any **static route tables in the route table are always next preference**. After that, **we've got any routes learned from a Direct Connect connection over BGP**. So **dynamically learned routes over a Direct Connect connection. Direct Connect is always preferred over VPN, and then we've got any statically configured VPN route.** So **anything that's propagated in but it's learned from a static VPN, and then the last level of preference is any VPN routes that have propagated in that have been learned from BGP**. So BGP VPNs are always the lowest preference, the idea being if you've got any statically configured VPN connections than they're there for a reason and Direct Connect is always preferred because of the connection reliability and higher performance. So you need to be aware of that for the exam. **So it's always local route first and then longest prefix and if that's equal, static routes always win and Direct Connect is preferred over VPN. So it goes Direct Connect, BGP, static VPN, and then BGP VPN** and that's basically what I wanted to cover in this lesson, I just wanted to give you an architectural overview of how hardware VPN works, together with the high availability architectures that you can use. In the next lesson, I'm going to be talking about Direct Connect and Direct Connect is obviously the physical private connection between your on premises network and AWS. So if you do have any queries or feedback about any of the topics I've covered and lesson or the wider course, then please do reach out. You can do so directly for your email. Connect with me on social media, use the Linux Academy discussion area, or my preference is that you join me on the Linux Academy Community Slack, and you can find that Slack that linuxacademy.com. If you don't have any questions or queries, go ahead and mark this lesson as complete and join me in the next where I'll be talking about Direct Connect.