**VPC PEERING**

VPC peering is a way to link multiple VPCs together. I'm going to talk about VPC endpoints which is a way that private VPCs can access public endpoints such as S3. I'll talk about how IP version six can be utilized within the VPC. I'll cover egress-only internet gateways and then lastly, I'll talk about subnet and VPC sharing.

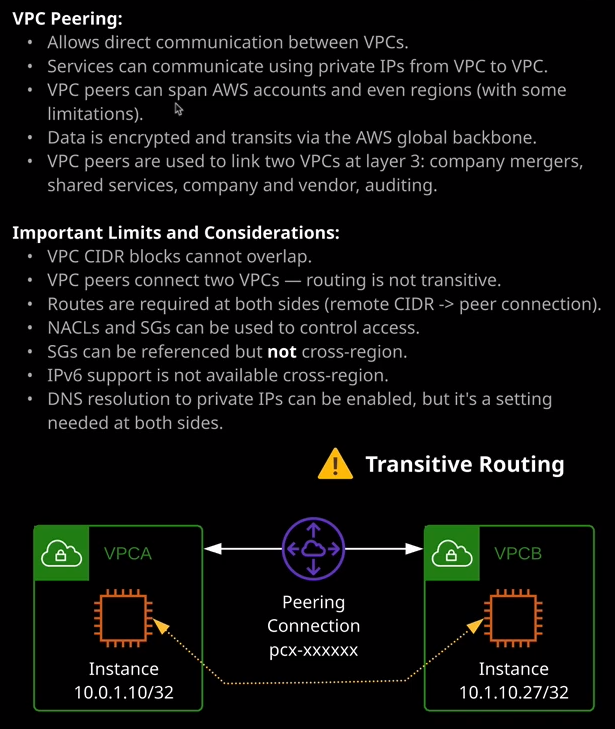
So let's get started on the first topic is going to be VPC peering now you just mentioned VPC peering is a way to link or connect two VPCs together without using any additional non AWS services. **It essentially allows direct communication between two isolated VPCs. Now once VPCs are connected together using VPC peers. Services in those VPCs can communicate using private IPs from the source VPC to the destination and vice versa. VPC peers can span AWS accounts and even regions, although that does have some limitations. Now between VPCs when you're using VPC peering the data is encrypted and if you're using VPC peering cross region then that data transits via the AWS global backbone and so it benefits from a much lower latency and higher performance than using the public internet.**

Now, before I demonstrate the implementation of VPC peers I think it's important, especially for the exam, to understand why you'd use VPC peering. So we know from earlier in the course that **VPCs by design are isolated networks. Unless you configure otherwise, there is no connectivity between VPCs and under most circumstances this is fine, but you do have some situations where you do want connectivity between multiple VPCs.** Now earlier in the course, you've learned how **you can use things like an internet gateway to make a VPC publicly accessible in both directions or using a NAT gateway for outbound only access.**

Later in the course, I'll also be teaching you how you can use virtual private networks as well as a service known as Direct Connect, which are both ways to link a VPC with your on premises network, either using a virtual connection or a physical one.

**Now VPC peers used when you want to link VPCs together. So VPCs their networks in AWS and when you want to link them together in a scaleable, highly performant way, then you use VPC peering.**   
Now some examples of why you might want to do this is if you've got shared services running in a single VPC, and you want to make that accessible to other VPCs maybe a database, an ID provider, or another bespoke business system. You might also want to connect your VPCs to a vendor or partner system to access an application that they provide, or you might want to give them access to your VPC and that might be during, for example, a security audit. You might also have a requirement to split an application up into multiple isolated networks to limit the blast radius in the event of any network based attacks. So there are lots of scenarios where separate, isolated VPCs makes sense but there are also equally just as many scenarios where you want to join VPCs together. In short, **VPC peering allows to normally isolated VPCs to communicate with each other at a network level.**

Now the way that VPC peering works is it uses **an object called a peering connection**. This is essentially a network gateway object similar to an internet gateway or a NAT gateway, but used to join two VPCs. Now an important concept to understand for the exam is that **a VPC peering connection is a link between two VPCs no more and no less.** On this diagram, we've got a link between VPC A and VPC B and we've got a single peering connection that connects each of those VPCs together. If you need to connect more VPCs then you need more peering connections. It's absolutely critical for the exam to understand that a VPC peering connection connects two VPCs if you want additional connectivity so more VPCs then you'll need more peering connections.



Now is probably easier at this stage rather than just talking about the theory to demonstrate VPC peering via a demo and that's what I'm going to do. So I'm going to move across to the AWS console and go into the VPC area of the console. Now what I have already done before I started this lesson is apply a CloudFormation template that's created two VPCs so VPC1 and VPC2. Now VPC1 actually mirrors the example VPC that we used in the previous topic of the course. So let me just bring that up to demonstrate. So VPC1 looks very much like the example in the previous topic of the course, so it's using the CIDR range of 10.0.0.0/16. It's operating in US East 1a. Inside two availability zones and then in each of those availability zones we've got a single public subnet and then two private subnets one for an application tier and one for a database tier. Now VPC two has been created using the 10.1.10.0/24 CIDR and it only has a single subnet inside it, and what the CloudFormation temple has also done is created a number of instances inside this VPC. First we got the bastion host that you'll remember from the previous topic in the course, and that's the one that I'll be connecting into for this demonstration. We've also got the application instance, and if you remember from the previous topic, this is an application that's running in a private subnet then we've also got app instance two which is an instance that's running in the second VPC known as VPC B or VPC 2. So you'll see these referenced as either VPC A and B or VPC 1 and 2 but the structure is exactly the same. Now, creating a VPC peering connection is actually a two step process. The first step is to create the VPC peer and the second is to accept it. One side is the requester and the other is the accepter. Now, with this demonstration, both of these are going to be within the same account but it could be two different accounts, two different regions or a combination of those. So while this process will occur inside the same account, so I will be both the requester and the accepter. It could be a completely separate process, but I'm going to demonstrate how this works. So first I'm going to create the VPC peer I'm going to request it. Now you do that from the peering connections item under Virtual Private Cloud. So I'll click on Peering Connections and I'll create a peering connection. Now the information that you need to provide is a name or a name tag for the peering connection and what I always do is use the two names of the individual VPC. So in this case, I could do VPC 1 to VPC 2 or it could also do as on this diagram VPC A to VPC B. You'll need to select the request of VPC. So this is the VPC inside your account that you want to peer from so on this diagram it would be VPC A. In this account, it would be VPC 1. So this VPC is the one that's requesting the connection. It's initiating it on this architecture, it's the one on the left. You also need to provide the other VPC. The remote end of the connection the VPC which has the accepter. Now you can select my account or another account or this region or another region I'll talk about the differences between those later in this lesson, but for now I'm going to select VPC 2 which is the VPC B or the right side of this architecture diagram. Now, when you select it, what you'll notice is you'll see the CIDR ranges for each of the VPCs in the peering connection. Now this is important to understand for the exam because **you cannot create a VPC peer between two VPCs with either a complete or partial overlap of these CIDR ranges. So in this case, 10.0.0.0/16 and 10.10.0/24 do not overlap but if they did overlap, if I use the same CIDR range for both I would not be able to create the VPC peer** and that's important if you remember to start the course I stated that whenever possible, I would always use completely separate ranges for any VPCs that I create and that goes for both inside AWS inside other cloud environments, such as Azure or Google, as well as on premises networks get used to using completely separate CIDR ranges. It just makes things easier. Now once we've set that up with set requester and an accepter VPC I could go ahead and create the peering connection. The information for the connection will be confirmed, but this is only one side of it. This simply means that we requested the VPC peer. We've initiated it. The next step is to accept that. Now **because this is in the same account I'll be doing the same process from the same location but if I was creating this to a separate account, then I'd need to be in that remote account and accept that. So I need to be the root user or a user with permissions to accept this connection. If it was in a different region that need to change the region on this dropdown and accept it from the other region but because it's in the same region and same account, I can simply click on Actions and accept the request.** I'll need to confirm it and once I've done that, the VPC peering connection will be active between the two VPCs.