Now there are other steps that are needed and to demonstrate that I'm going to go ahead and connect to the Bastion host. So I'm going to right click and hit connect to get the connection string. I'm using an SSH key that I've created earlier in the course, so I could just go ahead and copy this into my clipboard and then move over to my terminal. I'm going to paste this command in, but what I'm also going to do is use a hyphen uppercase A because I want to use SSH authentication forwarding. I want to be able to connect to this Bastion host and then hop in to other EC2 instances without having to copy the SSH key and I've demonstrated this else around the course. So go ahead and connect to this instance. I'll accept the authenticity and there we got now connected to this Bastion hosts. So I'm connected to an instance inside VPC A also known as VPC 1 inside this example. Next, I'm going to select app instance too and copy the private IP address into my clipboard. What I want to do is move across to my terminal and attempt to ping this IP address. Now, you might expect that to be operational now that we've got this VPC peering connection that's active but that's not the case because there are a number of other steps that we need to do. We've now got the peering connection in place, so we've got the network connectivity but just as with other gateway objects inside AWS, we need to configure the routing and that's the next step. So I'm going to move back across to the VPC console and I'm going to go to route tables. I've got this route table created and this is the one that's in VPC 1. So this is the VPC that's left of this diagram. Also known on this diagram as VPC A what I'm going to need to do just as when I created a route for the internet gateway. I want to create a route so that any traffic that's destined for this other VPC get sent to this VPC peer because the VPC peer is simply a network gateway object. So to do that, I'm going to select the public route table. Now, if I was going to demonstrate this from any other instances inside the VPC I would need to edit each of the appropriate route table. So, for example, routable private A and routable private B but for this demonstration to keep things simple, I'm just going to want to test the connectivity from the Bastion host, and that's in the public subnet using this public route table. So I'm going to select the routes tab, and I'm going to edit routes. Now we can see that we've already got the local route, which is for the CIDR range of this VPC. We've also got a default route pointing at the internet gateway, which is the reason why I can connect to this Bastion host but I'm going to add a more specific route. Remember, the more specific the route, the more priority it gets in the route table. Some going to add the CIDR range for VPC 2 or VPC B and the CIDR range for that is 10.1.10.0/24. Once I have got that CIDR range added I'm going to click on the target drop down, go to peering connections and the peering connection I've just created will be highlighted. So I select that and click on save. Now if I go back to my terminal, the ping is still not running. The reason for that is that whilst I have now got this route from the left VPC so VPC A or VPC 1 across to the right VPC I don't have the same route in the other direction. So there's no way that traffic can leave this VPC and make it back to this one. So I need to add the same route, but in reverse. So I go back to route tables. The CloudFormation template hasn't created a custom route table for VPC 2 so we've only got the default route table. So I'm going to select that. So it's the only one in the list that matches VPC 2. I go to edit routes and add a route and this time I want to add the CIDR range for this VPC. So this is 10.0.0/16. Once that's added, I'll click on the target dropdown I'll select peering connection I'll select a VPC peer. So that's the same VPC peering object as I just created. So it's one VPC peering connection object for both of the VPCs. Now that's added. I'll click on save routes, click on close, and go back to my terminal. Now, note we're still not seeing any connectivity between these two instances. So what we've done now is we've established the peering connection. We've established the routing, but we still need to allow the traffic to go from one instance to the other. There might be applicable network ACLs or there might be security groups that we need to edit to allow this traffic to flow. So that's going to be the next thing. So first I'm going to check out Network ACLs. The CloudFormation template is created nacl-public which is the one that's associated with the Bastion host at the A or the VPC 1 side. So first I'll check the rules on this NACL. All incoming traffic is allowed and all outgoing traffic is allowed. Next, I want to check the equivalent network ACL at the other side. So there's only one that's associated with VPC 2. So I'm going to select that one and again that's got the same inbound allow and outbound allow. So we now know, because we understand Network ACLs. There are no network ACL restrictions, which were impeding traffic flow between the left side and the right side. So the only thing that could be in play are the security groups so now I want to select the security group that's associated with the instance inside this second VPC. So this is this one. It's AdvancedVPC-2SG. So if you look at that security group and look inbound rules, we've got no inbound rule. So there's no traffic allowed into this instance, and that's the problem. Now we've got outbound rules as well, but they all look good. They allow any outbound traffic. So this is the problem. The reason that we're not getting any connectivity is because the security group around this instance on the right on this diagram isn't allowing any inbound traffic, so we need to correct that. So I'm going to go to edit rules and I'm going to add a rule. Now, just for this demonstration, I'm going to click on this dropdown and I'm going to add all ICMP traffic. So we just want ping traffic. So that's ICMP IPv4 and I'm also going to allow SSH connection. So I'm going to find SSH in the list and I'm going to click it. Now at this point, **we could add IP addresses or CIDR ranges representing this side of the architecture, so VPC A or VPC 1 That will be one way to do it but because security groups are capable of referencing logical AWS resources we can actually reference the Bastion security group so we can allow any connections into this instance from our Bastion host and we can do that because these VPCs are in the same region. If we create a VPC peer between two VPCs that are in the same region, then you can reference security groups. Now that same rule applies if they're in separate accounts but in the same region. If they're in separate regions then you cannot reference security groups, you can only use IP addresses or CIDR ranges.** So that's important to understand the **exam security group referencing can be cross account, but not cross region.**

Now to make this easy. I'm going to go back to instances. I'm going to select the Bastion host and I'm going to get the security group that's associated with that Bastion host. So I'm going to select security groups and I want to copy the security group ID of this Bastion security groups. I'll copy that into my clipboard, and I'm going to paste this into both these boxes. So this will allow any connections from the Bastion host security group into this instance and if I save both of those rules and click on close and move back to the terminal. Now, the ping is working because now we've got inbound connectivity from the Bastion host to this application instance and that's working across the VPC peering connection that I created earlier in this lesson. Now, to demonstrate that works when I also want to do is click on EC2 I want to go to the running instances, select the application instance, go to connect, and I'm going to get the username and IP address of this second instance tthe application instance in VPC 2 I'm not going to use the PEM key to authenticate because remember at the start of this lesson a connected using authentication forwarding. So I want to make use of the PEM key that's on my local machine. So I'm going to move back over to the terminal just exit out with this ping I'm going to use SSH and then hyphen uppercase A and I'm going to paste in the username and the IP address of this instance in VPC 2. I'll need to accept the authenticity which I'll do and then I'll be connected into this instance so now I'm connected into this instance, which is in VPC B, also known as VPC 2.

So far, you've learned that **you can create a VPC peering connection between two individual VPC. You have to create that peering connection. You have to add routes and you also have to configure any security allows either in network ACLs or security groups and once you've done all of that, you've got network connectivity between two VPCs.**

VPC peering is actually capable of some really interesting functionality as well as some limitations that you need to be aware of. I want to demonstrate this functionality right now. So remember earlier in the course I talked about how any public instance—**so any instance with the public IP address gets a public DNS name and how you ping this DNS name it actually resolves to the public IP address if you do it from your machine or any public internet machine but if you do it from inside the VPC it resolves to the private IP address.**

Well if I move across the terminal and ping this address, I also get the public IP address. Now, if you're communicating between these two VPCs and you want to do it using private IP addressing you don't want to resolve to the public IP address**. VPC peers are actually capable of doing this private IP address resolution**. To enable that, you need to go to the VPC peering connections so I'll select peering connections I'll go to actions and then edit DNS settings. Now, to enable this special DNS resolution feature, you need to tick both of these boxes. You need to allow DNS resolution at the requester side as well as they acceptor side, if you enable both of these and hit save and then close this down and to demonstrate this if I go back to the terminal and run this command again to ping the public DNS name of this instance of the Bastion host then I'll now get the private IP address and you might need to wait a couple of minutes for the DNS cache to clear before this resolves to the private IP address but after a number of minutes, it will start to resolve to the private IP address of the Bastion host. So that's one of the really cool features of VPC peering. **As long as you enable that option inside the peering connection, you're able to ping public DNS names and resolve those to private IP addresses and that means that once enabled any traffic between these two VPCs will be using private IP addresses and so won't be traveling over the public internet.**

Now, one final thing that I really needed you to be aware of the exam is something called **transitive routing.** In this example, I've created a VPC peer between two VPCs so VPC A and VPC B. In this example, I've got three VPCs and if I wanted to connectivity between all three of these VPCs such that A could talk to B, B could talk to C, and C could talk to A, and vice versa. Then it might be a logical to assume that I'd only need to create two VPC peers. So one between A and B and one between B and C and using that connectivity could happen from a through B and then to C but that's not the case that's known as transitive routing. VPC peering cannot do transitive routing and so, in this case, A could not talk to C and C could not talk to A. It's really critical to understand for the exam that for any connectivity between VPCs you need its own VPC peer. So in this example, we've got three VPCs and to give complete connectivity between all VPCs we would need a third VPC peering connecting A and C so that's known as transitive routing and VPC peers do not support that. So remember that for the exam. If you got two VPCs you need one VPC peer to connect those two. If you got three VPCs then you need three VPC peers connecting each individual pair of VPCs for a complete mesh network.

Important things for the exam: **you can't have overlapping CIDR ranges, peers can be used to connect two VPCs together in the same region of the same account, different regions of the same account, different AWS accounts, and even different regions in different AWS accounts. You need to configure the peering, the routing, the security so that's network ACLs and security groups and then you also need to enable DNS resolution if you want to do the public DNS to private IP address resolution.**