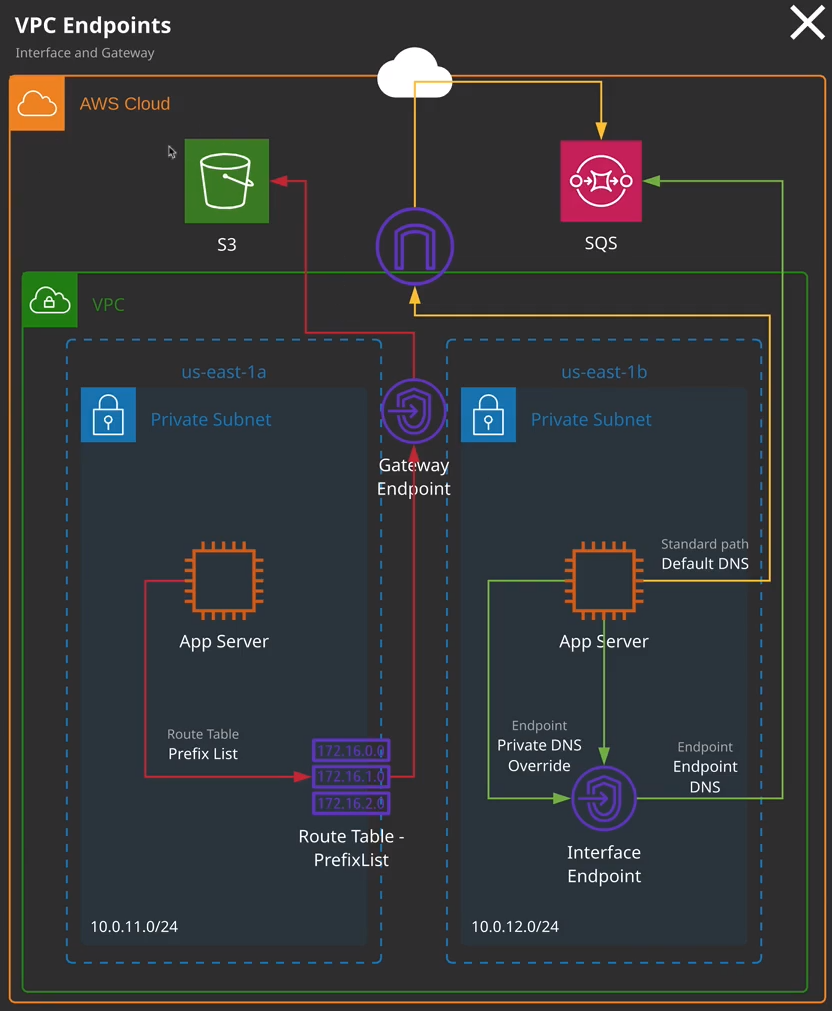
**VPC Endpoints**

Now VPC endpoints they're gateway objects that you create inside the VPC so they're much like NAT gateways and Internet gateways but they fulfill a very specific purpose.   
**They could be used to allow instances inside a VPC to connect to AWS public services,** for example S3 without the need for the VPC to have attached Internet Gateway, NAT Gateway, or be public. So essentially, if you want to access a service like S3 from a VPC and you're not allowed to use VPC endpoints then the instance or the service that you're accessing S3 from needs to be able to access the S3 public endpoint and that's an endpoint that's outside the VPC and so it needs to be able to access the internet or this public endpoint using either an internet gateway or a NAT gateway.



Well, let me show you how that works. If I move across to the terminal right now, I'm connected to the bastion host in VPC1. Now if I look at the endpoint architecture imagine that instead of just having these private subnets, we've actually got a public subnet on this diagram and this bastion host is sitting in this public subnet and it's got access to the public internet as well as these two public endpoints so S3 and SQS. Now, if I opened up the S3 console so I will open it up in new tab. I've got these two buckets inside this account, one of which is labeled www.associatecats.com now if I go across to my terminal and I run this command so AWS S3 ls then I'm able to perform a listing of S3 from this instance. Now I can do that for two reasons. The first is that the instance is able to access the public internet and public AWS endpoints using the internet gateway and the second is that this instance has an associated instance role with complete access to S3 so I could use this instance role to interact, to create buckets, delete buckets, upload and download objects. I've got access to S3 from this instance. Now what I'm going to do to demonstrate this is if I also go to the private instance that's also inside this VPC. So this application instant so app instance. If I just right click and hit Connect and I'm going to copy the username and IP address for this instance into my clipboard, move back to the terminal, I'm going on a SSH-[big]A and then this username and IP address and this is using this SSH authentication forwarding. So I don't need the PEM key on this instance. I'll accept the authenticity and I'll be connected through to this private instance. Now, this is an instance that has public internet connectivity and it has it via some NAT gateways so I can connect to the public internet and the public AWS endpoints in an outgoing only way using these NAT gateways and because of that and because this instance also has an attached instance role, I'll be able to do an AWS S3 ls and get the same S3 listing but what about if this instance did not have public internet connectivity and I can simulate that by moving across to the VPC console, and I'm going to edit the route table for the subnet that this instance is in and I'm going to remove the route for the NAT gateway. So I'm going to go to route tables and I know that this instance is in availability zone B. So I'm going to pick the private B route table. I'm going to click on routes, edit routes, and then I'm going to remove the whole route that goes to this NAT gateway. So I'll delete that and hit Save. Now if I go back to the terminal, and attempt to ping the public internet this time. Now, I can't do that because I don't have any method of accessing the internet or the public AWS endpoints. What this also means is I won't be able to do an AWS S3 ls. This will fail because this instance has no method of communicating with the public AWS endpoints. **So a requirement to be able to access any of the AWS public services is a method of communicating with those endpoints. Now those endpoints architecturally live inside the AWS cloud, but they live outside of your VPC. They live in a public address space and this goes for any public services so CloudWatch, S3, SQS, SNS, any of the public AWS services to communicate with them either over the public internet or from a VPC you need a method of accessing the public internet. Unless, of course, you use VPC endpoints** and that's what the topic of this lesson is going to be.

**VPC endpoints essentially provide a method of connecting to these public endpoints without having a publicly accessible instance or a public gateway so you can connect to it from any of these private instances by making use of one of two different types of VPC endpoints.** **Now the two different endpoint types that we've got are gateway endpoints which are used for accessing DynamoDB and S3 and then we've also got interface endpoints which could be used for everything else.**   
So, for example, SNS and SQS. Now picking the type of endpoint to use depends on the service and I want to demonstrate both of these and exactly how they work because they each operate in a slightly different way. So we're going to start with gateway endpoints because that's how we can access S3 from a private only instance or a private only VPC. Now you can access endpoints from inside the VPC console, virtual private cloud, and then select endpoints, Endpoints are created inside a specific VPC So I'm going to go ahead and click on Create Endpoint and you'll need to pick the specific service that you'll need to use. So note how we got a type column and most services use these interface endpoints but we've got DynamoDB that uses a gateway endpoint. If I scroll down further still, we got S3 which also uses a gateway endpoint. So I'm going to go ahead and select the one for S3 which will create a gateway endpoint. I'll also need to select the specific VPC. So it's VPC1 that this instance that I'm connected to in my terminal is created within so I'll select VPC1.

Now gateway endpoints use what are known as **prefix lists. Prefix lists are essentially specific things that get added to a route table and they are logical representation of the IP address ranges that a service uses so we'll be adding a prefix list to specific route tables that we want to be able to use this gateway endpoint** so if I look here we're able to select the route tables that we want to use with this gateway endpoint and if you hover over subnet you should be able to see which subnets are actually controlled by these specific route tables. So what I want to do is allow any of the database A or B as well as the application A or B subnets so the private subnets in this VPC to be able to use this gateway endpoint. So I'm going to select this route table, and then I'm going to select this route table. So this route table is actually used for the public subnets and because that has an internet gateway already which rise this public connectivity, I won't bother selecting that one. We could and it would override the use of the internet gateway because prefix lists are more specific types of routes than the 0000/0 that's used for the internet gateway. Now this route table at the top is the main route table for this VPC so I won't be selecting that one. I'll leave these two as selected so these control the private subnets inside this VPC and I'll click and create endpoint. Now before I hit create, it's worth noting that a gateway endpoint is capable of being associated with a policy. This policy can restrict what this gateway endpoint can be used for. The default is to allow full access. So anything inside the VPC can use this gateway endpoint but you can restrict it using a gateway endpoint policy that's a little beyond the scope of what you need for the associate level solutions architect exam but I do want you to be aware that gateway endpoints can have associated policies. For now I'll leave this as full access and I'll hit create endpoint. At this point, if I go to the route tables for this VPC so I'll hit route tables and then select the private A route table and then click on routes. Notice how we have this new item, **this prefix list. This represents all of the IP address ranges used in this region by a specific service in this case S3.** So what happens is that when anything inside any of the subnets where this route table is used attempt to access S3, they're directed at this VPC endpoint, which is just another gateway object. It's the same as the Internet gateway, the NAT gateway, and even VPC peers. They're all just logical gateway objects. So in this case, anything trying to access S3 will now go via this gateway endpoint. If I go back to my terminal and I attempt to ping an internet IP address, I still won't be able to do that because I've got no method of accessing the public internet but if I do an AWS S3 ls I will be able to access S3 via this gateway endpoint and **that's one of the benefits of using VPC endpoints that you can get access to public services without having the public internet connectivity.**

Now, as I mentioned **gateway endpoints are actually used using prefix list that are added to route tables. This happens automatically whenever you create a gateway endpoint you specify which route tables you want it to be used on, and it automatically adds these prefix lists.** Now **gateway endpoints, as I also mentioned, could be restricted via policies gateway endpoints are highly available across all availability zones in a region.** **So I only have to create one gateway endpoint to access S3 from any of the availability zones inside this VPC.** For the exam, remember all of those points as well as the fact that **gateway endpoints are used for DynamoDB and S3 everything else uses interface endpoints** and that's what I want to talk about next, before I do one last crucial point. **So gateway endpoints, they use routing so they have entries on a route table. Essentially, this prefix list just represents a group of CIDR ranges that are updated by AWS but it's using a routing system. So your route tables have these prefix listed. Traffic gets directed at a gateway endpoint and it uses routing. It does not use DNS at any point for gateway endpoints.** Now the reason I stress that is the interface endpoints which are used for everything else do use DNS and that's what I want to talk about now.