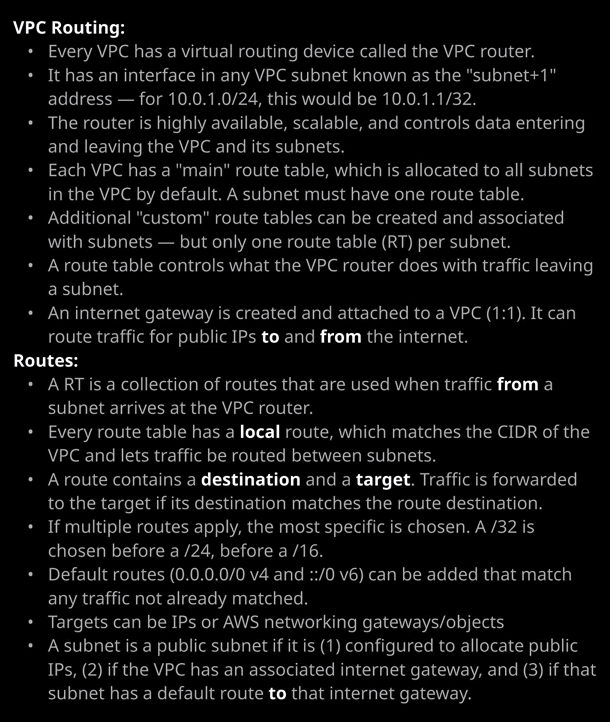
Routing inside a VPC

So I'll cover routing, the internet gateway, and the differences between public and private subnets.  
**So inside every VPC is the VPC router. It's a highly available network device, which is largely hidden from your view so you won't see it mentioned anywhere inside the AWS console. The VPC router is what moved traffic between the subnets in your VPC as well as to and from the internet** as I'll demonstrate in this lesson. **The VPC router is also used in conjunction with other services to provide access to and from other VPCs as well as other networks such as those are on premise or in a corporate data center** and I'll talk about that in much more detail later in the course when I cover hybrid networking, **the VPC router runs as the network plus one address in every subnet, so it's one of the reserved addresses.** **It's always available and it scales to whatever data rate is required in your VPC.** So essentially it's something that you don't need to worry about.

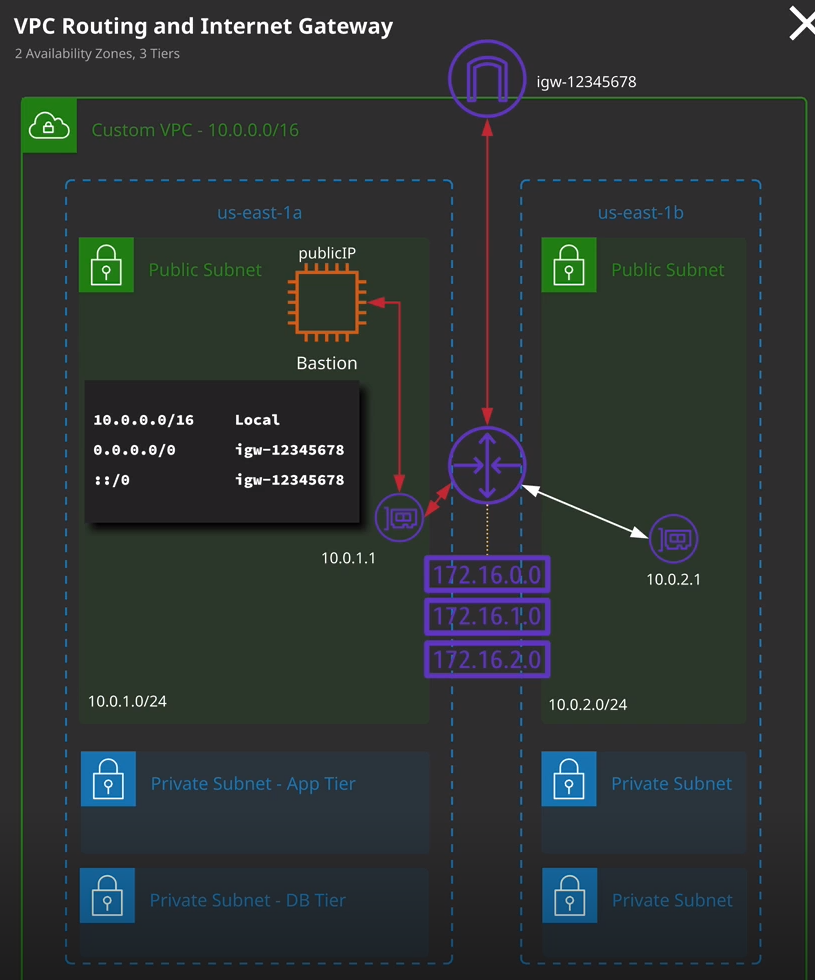


Now the **VPC router is actually controlled by using route tables and route tables are containers for one or more routes**. **When you create the VPC, it creates a main route table for that specific VPC.** So you'll see that I've got two route tables listed in my console and if a drag aside this column, one of them is for the VPC demo VPC. **So for every VPC that you create in AWS, you get one main route table. Now this route table is associated with any subnet that you create inside that VPC unless you create a custom route table and explicitly associate it with one or more subnets. A route table controls what the VPC router does with traffic leaving the subnet. So essentially route tables control the VPC router.**

To illustrate how route tables work, I actually want to show you and so to do that, I'm going to talk about public and private subnets. Now if I go back to subnets we're looking specifically at these top six subnets. These are the ones that I created inside the VPC demo VPC and some that's actually come in one of two types public and private. By default, every subnet that you create inside a custom VPC is a private subnet, so resources inside a private subnet they get private IP addresses, and they have no internet access. Now, if I look at the route tables for these subnets, so I open up route tables and I pick the main route table for the VPC demo VPC and then I go to the route tab. **There is a single route and that's known as the local route. It's essentially what the VPC router uses to pass traffic between subnets in the VPC.** You'll notice **that it's got a destination which matches the CIDR range for the VPC, and the target is local and that means that all of these subnets are connected to the VPC router, and it can be used to swap traffic between these different subnets.** Now this local route is always there. If I click on edit routes, I can't make any changes. It always takes priority, and it's something that's present in every single route table. **Route tables are essentially lists of routes. When the VPC router receives traffic, it checks the destination IP address of the traffic and attempts to map it to one of these routes that play to that subnet. When it finds one, it selects it, and then, based on what the target is, it forwards that traffic onto that destination, the exception being if it's local, then it knows that it can handle itself so the VPC routed can handle local traffic directly.**

**Now it is possible for a route table to have multiple routes for the same destination and if that's the case, it picks the one with the most specific prefix.** So the higher this value, the more specific it is and the higher priority it is when the VPC router comes to selecting it. So the VPC router can only ever select one route, and it's looking for the one that's most specific. Now this is the local route. You can't do anything with this one. It's always going to be there. It's always going to be set to local, and you're never going to override it but you will have other routes and they can have different prefixes. So when it comes to the exam **it's always the highest prefix that matters.** So /32 for example, means a single IP. So if you have a route for a /32 even if you also have a route for the network that that IP is in the /32 will always take priority. A /0 here means every IP address. Remember earlier in the course where I talked about 0.0.0.0/0 being representative of all IPs? Well, that's what that means, and I'll show you how that looks in a second.

Now, a subnet can be made public, and making a subnet public is actually a multiple step process. There are three steps that we need to follow in order to allow resources inside a subnet to be public resources. So we make a subnet public by doing these three steps**, the first step is to make the subnets allocate public IPs by default when any resources are placed into them**. To do that, we need to select the subnet, click on actions, modify auto assign IP settings, and then enable that subnet so that it automatically assigns that IP version four address to any resources that are created in it. So I'll do that for this subnet and then I'll select the subnet in availability zone B and do the same process. So I'm making it so that both of these public subnets allocate public IPs by default. Now this is the way that subnets in a default VPC are configured by default. So if I select one of the subnets inside the default VPC, modify the auto assign IP settings, that is switched on by default. So all that does is if you deploy a resource into it then by default, it does get a public IP. You can override it, and you don't actually need to set it at a subnet level when you're deploying, for example, and EC2 instance into a subnet, you can explicitly say that you want a public IP assigned, but it is simpler for any public subnets just to change the default to assign an IP address of public IP address.



Now **the second step that we need to do is to allocate what's known as an internet gateway.** Now resources with IP version four addresses in the VPC they always get private IP addressing. Using IP version four EC2 instances, for example, never get public IP addresses configured on the actual instance. So if you ever log on to an EC2 to instance and look at the networking configuration, it will never have a public version four IP address. When we assigned public IP addressing to something, we're just creating a record of that association. So, instance X has a private IP on its network interface and that network interface has an associated public IP. Now **it's the internet gateway that actually handles this process, and internet gateway is something that handles communication to and from the public internet, and it works with both IP version four or IP version six** but for now, I'm focusing on version four. So the second step in making these public subnets publicly accessible is to create an internet gateway. So I'm going to do that. I'm to go to internet gateways, and there's already one created for the default VPC. This is one of those steps that's automatically done for you for the default VPC, but I'm going to create one for my custom VPC. So I'll create an internet gateway. I'll need to name it. So I'm going to call it IGW for internet gateway, and then I'll call it VPC Demo, which is the same name as the VPC I created. So I'll set that and hit create. Now when you create an internet gateway, you don't initially attach it to a VPC. You create an internet gateway, and it sits here in a detached state, waiting for you to attach it to a specific VPC. So that's what I'll do a next. I'll click on actions, attach to VPC, and then select my VPC demo VPC, and click attach. Once I've done that, it shows is attached, and then it's linked to my VPC demo VPC. It's really important that you understand that **an internet gateway is a highly available product by design. You don't need to worry about the performance or the failure an IGW or an Internet gateway is created it's attached to one VPC that's important to understand for the exam. An internet gateway can only be attached to one VPC and a VPC can only have a single internet gateway attached because it's highly available by design, you don't need to worry about multiple internet gateways for different availability zones. You simply create an internet gateway and you attach it to the VPC and then it's made available across all the availability zones for that VPC.**

Now, when the internet gateway receives any traffic from an EC2 instance, if that EC2 instance has an allocated public IP then it adjusts those packets. What it does is **it replaces the private source IP address on the packet with the associated public IP address of that instance.** So when these packets go from this private instance through the internet gateway it adjusts the packet, replacing it with the public IP address that's associated and then it forwards it through to the public internet. So any destination IP address that receives that packet, we'll see the source IP address as this public IP but the public IP is not stored on that instance. **Essentially what this internet gateway is doing is it's performing a process called static network address translation. It's known a static NAT or S NAT.** It's why instances can use public IPs and that traffic appears to be coming from that public IP without ever being configured on the instance itself. Essentially, **the internet gateway is translating the private IP address to a public IP** address. It's a 1 to 1 translation, and that's known **a static network address translation.** So if you ever get any questions in the exam talking about networking issues and an answer is apply the public IP address to the network interface on the operating system. Now you know that answer is wrong. **In AWS, no internal product ever has a public IP not on the operating system. It's the internet gateway that translates between the private IP and the associated public IP and that goes both ways. So when this instance sends data out and then whatever the destination is wants to respond, it will be responding to this instance's public IP when the internet gateway receives those packets, it will translate from that public IP to the private IP of the resource that's associated and then it'll forward the traffic through to this instance.** So it works in both ways, essentially to an external perspective, that instance does have a public IP but internally, it's only got the private one configured, and it's important to understand the distinction.

So now that I've configured these public subnets to allocate public IP addresses and I've created an associated an internet gateway with this VPC. **The last step is to add route**. So we need to tell the VPC router to forward any IP traffic destined for the internet through with this internet gateway. So to do that, we need to create a route table. Now it is always best practice not to update these main route tables. I don't like doing it because I like to maintain these to be clean, and you can always switch back to them in an emergency. I always like to create a brand new custom route table, so I'll do that. I'll create a route table. I'm recalling rt-public, and every route table is associated with the VPC, so I'm going to select VPC demo and hit Create. Once I've done that I'll go ahead and close. I'll select this new route table and again you can see that it's got this local route already added, but I want to add another one. So I'm going to click on edit routes and I'm going to add an additional route. Now, remember the designation that I mentioned in a previous lesson, which refers to all IP addresses? Essentially, what I want to occur is whenever instances that are in these public subnets want to talk out to the public internet if we don't have any other routes, any other directives for this VPC router I want it's defaults to be to send it out to the internet, so we're going to create what's known as a default route. So we're going to use this designation, which is 0.0.0.0/0. So this represents any IP version four addresses and then for target I want to click on the dropdown, select internet gateway, and pick the internet gateway that I just created. Once I've done that, I can go ahead and click on save. Now looking now it's got two routes. If the destination was in the 10.0.0.0 network because this is more specific because this value of /16 is more than this value of zero. The VPC router would always prefer this route. So even though I've got this default route, this will only take effect for anything that isn't in 10.0.0.0/16. So, in effect, anything that's not inside this VPC will be sent to the internet gateway. **Now that the route table is ready, I want to associate it with one or more subnets, and these will be the public subnets inside my VPC.** So I'll select the route table, make sure it's the correct one, click on subnet associations, edit subnet associations, and I will want to select both the public A and the public B subnets. So public A public B and then click on save and that's it. That's all of the steps that I needed to configure to give these public subnets connectivity.

So what's going to happen is **any time I launch any resources into these public subnets because of the configuration that we did on the subnet itself, it's going to be given an associated public IP then I've got this route table that's associated with these subnets. So with public A and public B so that means that any traffic that leaves this subnet and hits the VPC router this route table is going to be used. Now the routes on it mean for any traffic that's destined for the VPC itself that's going to stay local. For anything else, it's going to go through to the internet gateway. When the internet gateway receives the packets, it's going to translate from the private IP of any of these resources to the public IP and forward it on to its destination and the same process will happen in reverse.**

Now, before I do that, there's one final thing that I want to talk about and that's the difference between **static routes and dynamic routes**. The routes that I've added to this route table so far are static routes, they are known **as static routes because I've explicitly added them and they don't change but there's also something known as route propagation**. So if I click on the route propagation tab and edit propagation, if I had a virtual private gateway that was associated to this VPC that I could elect to propagate any routes that it learned onto this particular route table. It's a way that I can dynamically populate new routes that are learned by the virtual private gateway, which is used for VPNs and automatically populate a route table. Now, certain types of networking products that you'll learn about later in the course, such as **VPNs and Direct Connect, can dynamically learn routes using BGP. Remember BGP is Border Gateway Protocol, as I talked about that in a previous lesson. It's essentially a protocol that allows us to advertise and learn routes from other routers. So if you've got a VPN or Direct Connect that support BGP and you integrate those with your VPC then you can enable this route propagation to automatically add those routes to these route tables.** Just save you some time and you don't need to do it manually. Now I have not configured this route, which is double colon forward slash zero ::/0 because this is an IP version six default route it's equivalent to this 0.0.0.0 but version six of IP.