Okay so to test this out, I'm actually going to move across to the EC2 console, and I'm going to create a private EC2 instance, I'm going to launch instance, select Amazon Linux 2, I'll pick my usual T3.micro, and configure instance details. I want to use my custom VPC so VPC demo. I'm going to place this instance into the application tier in availability zone A so that's subnet app and then because I'm placing it in here. It won't be allocated with a public IP address and that's what we want. We don't want it accessible from the public internet, so I'm going to go to next add storage. I'll accept the defaults. I'll go to next tags. I'll add a tag. The key will be name and then the value will be app-a for app server A. Now I'm going to configure a security group. I'm going to call the security group App server. I'll have the same for description, but I'm going to remove this default SSH rule. In its place, I'm going to add until the rule and click on the dro down and pick SSH but from the source, I'm going to type SG and pick the security group of the Bastion host that I created in the previous lesson. So just focus on this a minute because it's important. What I'm doing is **I'm not allowing any other connectivity from anywhere else apart from the Bastion security group. So in theory, I should be able to connect into the Bastion host, and then once I'm in the Bastion host.** I can connect into the EC2 instance, but I won't be able to checked into this instance from anywhere else and that's exactly what I want. I'm going to click review and lunch, go to launch. I'll use the same VPC demo key pair. So I'll acknowledge that I'm still in possession of that. I'll go to launch instances. Okay, so this is finished, so I'll be able to view instances, and I've got my application instance and it's got it's private IP. Now to perform this demo what I want to do is to be able to connect into the Bastion host and then into this private application instance and I want to do so without having to copy across my PEM file. So I don't want to leave any SHH keys on this Bastion host I want to forward them through. So in order to get this working I'm going to move to my terminal and, I'm going to run shh-add vpcdemo.pem. Now, because I'm running MacOS I'm going to use an uppercase K on that command and press enter. So that's added my identity to SSH in and then once I've done that I'm going to go back to the console, right click on the Bastion, and get the connection string. So I'm going to copy all of that into my clipboard. Move back to the terminal, paste that in, but then to get it so I don't need to copy my PEM key to the Bastion host. I'm going to put minus -A as a command line switch at the start. So that's shh -A and then specify the PEM key to use and then the DNS name of the EC2 instance. So I'm going to press enter, verify the authenticity, and that will connect me through to the Bastion host. So this is the Bastion host. Now I'm at the Bastion host I'm going to go back to the console, right click on the app server, and hit connect and this will give me the connection string to connect through to the application server. Now, I'm not going to use this full command because I don't need to use this PEM key. So what I'm going to do is I'm going to copy the username and then IP address components. I'm going to copy that into my clipboard, move back to my terminal, type SSH and paste that component in and press enter. I'll need to verify the authenticity and what that does is essentially use this PEM key that I've got my local machine via the Bastion host to connect into this application server and now I'm on this application server. So you don't need to worry about how this works in detail. I'll make sure include a link in the lesson description that explains this SSH forwarding that I just did but just be aware for now that I am now connected to the application instance through the Bastion host. What I'll do next is I'll try and ping 1.1.1.1 which is internet IP address and note that I can't do that because this instance doesn't have a public IP and so it can't use the internet gateway to get to the public internet. What's going to fix that is this NAT gateway. So one thing NAT gateway to allow this instance, which is the application instance I just configured to be able to access the public internet. To finish off the configuration, I'm going to go back to VPC and I'm going to need to create a new route table just like I did for the internet gateway. So I'm going to create the route table but remember, because an internet gateway is highly available by design, **I only needed a single route table to cover the entire VPC. So recall both public subnets used a single public route table.** If I just cancel that for a second, this route table provided access to the internet gateway for all of the public subnets but because a NAT gateway is not highly available by design, I would need a single NAT gateway for every availability zone and so, in theory, I would need a single route table for each availability zone. So I'm going to go ahead and create the first route table to create route table and I'm going to call it rt-private-a. So the route table for private subnets in availability zone a and I'll need to select the VPC. So VPC demo and hit create. Now I'll be associating this route table with any of the private subnets inside availability one zone A. So I'm going to do that. I'm going to go to subnet associations, and it's on that association, and I'll be selecting the DB subnet in availability zone A as well as the application subnet that's also in availability zone A. So I'll need so select both of those and click on save. If I go back to the instance, it's still not working and that's logical because I haven't added the route. Now what I need to do is add a default route so that whenever either of these instances make a request to the public internet, it goes via this NAT gateway, so essentially anything that isn't matched with an existing route will be sent to NAT gateway as the default gateway, so I need to add a default route so click on edit routes. Now, just like with the internet gateway, I'll use this notation so 0.0.0.0/0 to denote any IP address. I'll click on the target, drop down, select NAT gateway, pick NAT gateway, and click on save. At that point, if I switch back over to this terminal, you'll see that the ping is now working. **So because I've got this NAT gateway set up as the default gateway, then the packets from this private instance this app server they're going to this NAT gateway, the IPs are being translated. It's moving up to the internet gateway, and then it's able to access the public internet as well as receive response traffic because the NAT gateway understand sessions, it's a session level product. So because this outgoing connection works the response for the outgoing connection is also allow back in but any connections initiated from the public internet will not be allowed to get to this private instance so that's the functionality that NAT gateways provide.**

Now to complete this configuration I would also need to create a second NAT gateway. So I create NAT gateway. This time it would be in the public subnet in availability zone B. I need another elastic IP. I'd create the NAT gateway. Again that would take a few moments to create. So I need to create another route table. This time I call it rt-private-b for availability zone B, select VPC demo again, I'd hit create, and then for this private route table go to subnet associations and edit those associations this time selecting the DB subnet in availability zone B as well as the application subnet in that same availability zone B, hit save, and then once the NAT gateway is finished creating I'd need its ID. So it ends in C94 I need to go back to the route table, select the private route table, and hit the route and then add a new default route so 0.0.0.0/0. This time I'd select NAT gateways but then I'd need the second NAT gateway. So this is C94. So I'd select that it saved routes and that would set up this fully highly available architecture.

So now **I've got two NAT gateways one in each availability zone, two route tables applied to the subnets in those availability zones, and then a default route pointing at that Nat gateway and that means if either of the availability zones inside my VPC failed then instances in the other availability zone would continue to have outgoing internet access.** So if I go back to this instance, just exit out of this ping, and then do a sudo yum update, that will be up to go out to the public internet and do a package update of this operating system.

Now in terms of the performance **a NAT Gateway initially supports up to five gigabits of bandwidth and can scale up to 45**. **If you need more then you can distribute the workload by splitting your resources into multiple subnets inside an availability zone and then specify that each of those goes to a separate NAT gateway, so you can scale them yourself by deploying more and more NAT gateways if you do require more than 45 gigabits of bandwidth.** If you do go above that you can start to see connection errors, but that is extreme usage. **So in most cases you can be better at using in NAT gateway rather than the old style of NAT instance.** Now there is a page that include in the description for this lesson, which does give you a good comparison between the features for a NAT gateway and a NAT instance, so **NAT gateways tend to be the preferred option**.

**There are very few situations where I would recommend the use of a NAT instance. The only real scenario is if you can't deploy a Bastion host and you want to utilize a NAT instance to provide similar functionality then you can combine those use cases but unless you absolutely need to save costs, then you're always better having a separate Bastion host and a NAT gateway and allow them each to play to their strengths. It's always better. A Bastion host can be secured and can offer incoming secure access, and a NAT gateway will scale to whatever load requirement you place on it whereas a NAT instance is always going to be limited.**

Now if you do have any exam questions where you do see NAT instances mentioned I don't expect that you will do but a question that you might face is that your **NAT instance is not working and if you do face that question, it's often that you haven't enabled this source and destination check. So by default an EC2 instance will only accept traffic for which it is the source or the destination. So with a NAT instance, you do need to change this and disable it so that it can receive packets that it is neither the source or the destination** for but I would be very surprised if you do see that question on any of the new versions of this exam. It tends to be a question that's only featured on the older versions, but that's everything that I wanted to cover in this lesson about NAT gateways. Again, it's a crucial part of the AWS networking product set. It gives private resources inside of VPC access to the public internet in an outgoing only way. So at this point, go ahead and mark this lesson as complete, and when you're ready, I'll see you in the next.