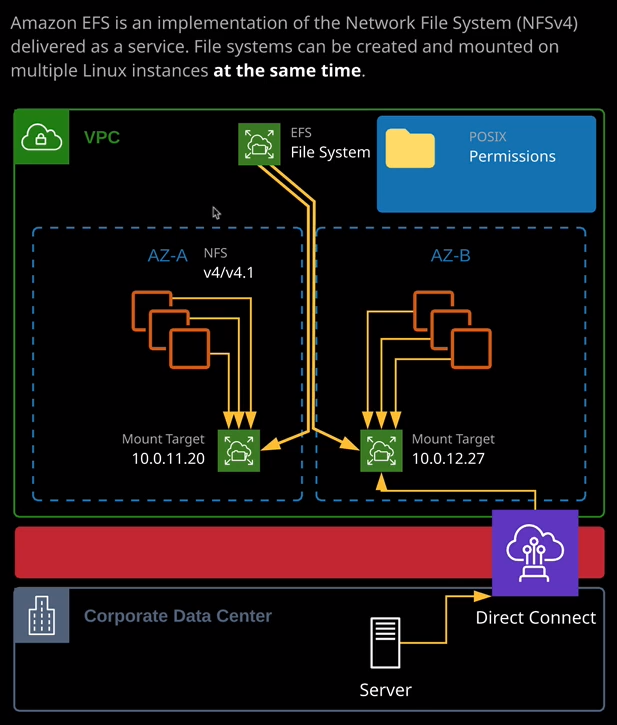
Welcome back and welcome to the network file systems topic of the course where I'm going to be reviewing all of the different file system products available within AWS.

Now what I will be doing with this topic of the course is expanding it as different file system products make it into the exam. But at this point, the only topic that I'm going to cover his **Amazon EFS or Elastic File System, which is an implementation of the NFS or network file system version four**.

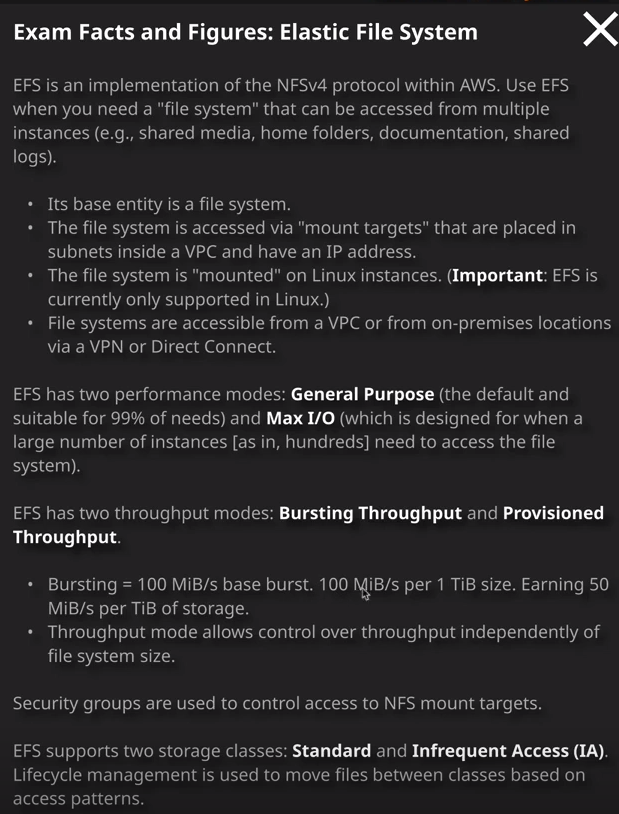


Now, **NFS allows you to deploy a file system into AWS, which could be mounted on multiple Linux instances at the same time.** Up until this point, the only storage products that you've had exposure to have been EBS and S3. Now I want to demonstrate why EFS is different than both of those. To do that, I'm going to do a quick demonstration, so I'm going to start by going to the EC2 console, and I'm going to create a key pair that I'll be using to connect to EC2 instances in this lesson and I'm going to call this key pair EFS and when I create it, it's going to download it to my downloads folder. Once I've done that, I'm going to move across to the CloudFormation console and apply a CloudFormation template that creates a basic VPC together with public and application subnets that I can use to demonstrate EFS. So I'm going to create a stack, going to select, upload a template, going to pick the vpc.yaml template file that's inside this lessons folder. So storage and content delivery network file systems and then EFS. So I'm going to select this template click on "Next" to keep things simple, I'll call EFS and the only parameter that I'll need to select is to pick the key pair that have just created. So EFS click "Next" "Next" again, acknowledge that I might be creating some IAM resources and then finally hit "Create Stack." Now, at this point, I'm just going to go ahead and pause the video until the stack is finished creating. So remember earlier in the course when **I talked about EBS or Elastic Block Store, that was a file system. It's a block file system, and it's the type of file system where you can only have it connected to one instance at a time. So if we look at all of these individual instances, or if we go to the volumes view, we can see that we've got three system volumes created, and each one of these is mapped or attached to a particular EC2 instance. Now I can create a volume, and I can attach it to an EC2 instance, I can store some files on that volume, detach it from the EC2 instance, and then attach it to another instance as long as it's in the same availability zone but the crucial part about EBS and block storage in general is that it can only be attached to a single EC2 instance at a time. It's essentially directly attached storage. It does operate over the network, but it is block storage, and so it can only be attached to one instance at a time.**

**One of the benefits of EFS is that it's a network based multi user file system, so it can be connected to multiple EC2 instances at the same time, and I want to demonstrate exactly how that works.** So the first thing I'll do in order to demonstrate this is I need to connect to these EC2 instances. So I'm going to do that. I'm going to go to the dashboard. I'm going to go to running instances. The first thing I'll do is right click on the Bastion host and hit "Connect" because I'll need to change the permissions on the SSH key. So in a copy of this command into my clipboard and move to my terminal. Now, if you are using a Windows operating system, you can safely skip this step. You won't need to do it. If you're using a Linux or Linux-based operating system such as Linux or Mac OS, you'll need to perform this step. So I'm going to go into the folder where I've downloaded this PEM key and change the permissions using this command. Now that I've done that, I want to copy the full connection string to get into this Bastion host into my clipboard and move back across to my terminal. I'm going to paste this command in, but I'm going to make one small change. I'm going to go to the start of the command and add hyphen uppercase A. Now, this is because I want to pass through authentication. I want to allow me to connect to this Bastion host and then connect into one of the application instances using the same PEM key, but without having to copy it onto the Bastion host itself. Again if you're using a Windows based operating system, this process is slightly different, and I'll make sure I include links in the lesson description that explain exactly how to do it but if you're using a Linux or Linux based operating system such as Linux or any flavors of Linux or Mac OS, then you can safely follow these instructions. So I'm going to press enter to connect to this instance. If it's the first time you've done it, you may be asked to confirm its authenticity. I don't need to do that. So now I'm connected to the Bastion host. I'm going to clear the screen. Next I'm going to go back to my console and I'm going to get the connection command for the application A instance. Now all I want is the username and the IP address. I'm going to copy that part into my clipboard, move back across to my terminal, and do SSH and then paste that in. Now, I don't need to specify a PEM key at this point because I used hyphen uppercase A to connect into this Bastion host so it's going to use the same efs.pemkey that's on my local machine. So I'm going to connect through to this instance.

Now at this point, I'm going to implement a simple version of what's on this diagram. So I'm going to create an EFS file system. The basic configuration of EFS is a file system. So first we need to configure a file system. So I'll go to services and then go to EFS. I'm going to create an EFS file system as the first step, and I want to make sure that I select the right VPC. **So the first thing straight away to realize is that EFS is tied to a specific VPC.** So I'm going to select VPC1, which is VPC that was created by the CloudFormation template that I applied at the start of this lesson. Now, as well as creating the file system in order **to allow access to the file system, we need to create what are known as mount targets. Now these are actually network interfaces that live in a particular subnet inside a VPC, and it's these targets that your EC2 instances will connect to using the NFS four or NFS 4.1 protocols.** **Now you can create as many of these mount targets as you want and the more you create, the more resilient you'll have. The one limitation is that you can only create one in each availability zone.**

So if you see what's on my screen now because I've only got this VPC configured for two availability zones I'll only have the option of creating it inside two availability zones. So I have to pick a subnet in each of these availability zones. So I'm going to pick the application A subnet and the application B subnet. Now, these are where my EC2 instances are located in, but you don't need to pick the same subnets. You just need to make sure that you got one in each of the availability zones that you want to use. Now, **the security of these mount targets is determined using security groups, so you'll have security groups associated with each of these mount targets. and it's these security groups that will control exactly what traffic can get in to these mount targets. Now, by default, it uses the default security group of the VPC.** I'm going to delete that because I don't want to use that. What I'm going to pick is the application instance security group, the same security group that's associated with the application instances because it's simpler to configure that way and I'll show why momentarily. So I'm going to select the application security group of both of these. Once I've done that I'm going to click on "Next Step." Now, there's a lot about EFS that you don't need to know that an associate level but I want to draw your attention to the specific important points.



So you can create an EFS file system in one of two different throughput modes bursting or provisioned. Now deciding between these two is important, so I'm going to go to the exam hints. **So with bursting, you get a certain level of performance based on the size of the file system you are creating. So with bursting, you get at least 100 MiB/s of base burst and a further 100 MiB/s per TiB of size. So it's important to understand, with bursting you're linking the size of the file system to its performance. With provisioned mode, sometimes known as throughput mode. You're able to specify these things separately so you can configure a certain amount of throughput independent of the size of the file system.** For this demonstration, though, I'm going to pick bursting. It is the default, and it's what I would recommend for most circumstances. **You're also able to pick between general purpose and max IO. Again general purpose is the default, and it's suitable for 99% of needs. You generally select max IO when you're using a large number of instances somewhere in the region of hundreds in order to get a benefit from this. So 99% of cases you would pick general purpose it is again the default, and it's suitable for most situations.**

Now, **EFS file systems also support encryption at rest and if you do select this option then as with other AWS services that use encryption, it uses KMS that I'm going to talk about elsewhere in the course.** I either have already talked about it, or I will be depending on the order that you take in these lessons, but it does use a KMS master key. So if you do select to use encryption at rest you'll need to pick the appropriate master key to use. For this demonstration though, I'm not going to pick encryption at rest, so I'll leave that unselected. You don't need to worry about any of the other options. So I'm going to go ahead and pick next step. Okay, at this point, I'm going to review all of the options. I haven't configured anything in availability zone 2c, because I don't have any resources in this particular availability zone. Obviously, I've got the Sydney region selected because it's my local region. If you are following along with this in your own environment, I would recommend picking North Virginia. Certainly if you're using any at the Linux Academy resources always pick North Virginia, but just to keep things fairly performant I've made sure that I've selected Sydney and it was also leftover from one of the previous lessons. So in this particular case, there are only three availability zones in Sydney, and I'm only using two of them. So I'm going to leave everything else selected as it is and go ahead and create the file system. Now note you don't have to specify an initial size for the file system. It essentially grows as you put data on it and remember, because I've picked bursting, I only get the base performance initially and I'll only get performance above that as I add data to this file system. Now, I will need to give that a couple of seconds to create. What's essentially happening **is it creates the file system relatively quickly but then obviously these mount targets will need to be created and properly configured inside the VPC.** So I'll need these to change away from this creating state before I'm able to use it. Now, accessing the file system is pretty easy. Obviously**, when you create a file system, you're given a DNS name for this file system.** So that's what this address is here and I'll need to configure this inside any EC2 instances that I want to access this file system. An important thing to understand, though from an exam perspective, **is that you can access this file system either from the local VPC, across VPC peers**, which you have learned about elsewhere in the course, but **you are able to access EFS file systems also across Direct Connect connections. So anything outside the VPC assuming you have appropriate private network connectivity can also access thes file systems.** So I'll refresh this a couple of times just to see if we're in a ready state. Obviously, we've already got one of these mount targets available. That's the one that's in availability zone 2b in Sydney. Okay, so refreshed that again, everything is up an running. We've got both of these mount targets in an available state.