Okay so I've refreshed that again everything's open running. **We've got both of these mount targets in an available state. So the next step is actually mounting this file system on one or more EC2 instances.** So that's the next step. So I'm going to move across to my terminal and remember, I'm connected to the application instance in availability zone A. Now the first thing to do in order to mount this is to **install the Amazon EFS utilities** and I'm going to do that by running a sudo and then yum which is the package manager, install -y just automatically accept yes, and in then the package name that I need to install is amazon-efs-utils. Now you can actually do this without this utility**.**

**EFS uses the NFS protocol and so all you need is the ability to connect to NFS file systems, which is a standard inside Linux operating systems but by installing the EFS utilities, you get a little bit of tighter integration with the FS**. It's recommended that you do install these utilities. Now, if I do a df -k that will show you all of the file systems that are currently mounted on this Linux instance. The one in particular that we're currently operating on is the root file system and you can see that by this forward slash. So this is the one that I'm currently operating inside. What I want to do is create another folder which I'm going to use to mount this EFS file system. Now to do that, I'm going to sudo mkdir which creates the directory /mnt so that's going to make a mount folder that already exists. So what I'm going to do now is run that same command, but create a folder in there called EFS.

Now in order to quickly and easily mount an EFS file system because I've installed the EFS tools. What I can do then, is there a sudo and then mount, which is a standard Linux command. I'll specify -t, which specifies the type of file system and because I've installed the EFS tools I can use EFS as a file system. Now, what I'll need **to do next is get the file system ID.** That's one of the integrations that's provided by these EFS tools, so I can actually refer to this by its file system ID. So I'm going to go back to the AWS console and select that. So it's fs-963f75af, I'm going to copy that into my clipboard, move back to my terminal. I'm going to paste that in, specify :/ so I'm mounting this location on this file system and then I need to specify the local folder on the Linux instance to mount this into. So that's /mnt /efs and that's the folder that I created moments ago. It'll take a couple of seconds, but assuming the security group has been configured as required, what you'll find if I run a df -k is that this file system is now mounted in /mnt /efs and if I cd into that folder and then do a listing we'll see that there's no files, but it is a valid file system. Now what I'm going to do is go ahead and create a file so I'm going to call this file test.txt. Now by default only the root user has the ability to create files inside this file system. So I will need to do a sudo and then touch test.txt. So that will allow me to create it and again, I'll do a sudo and then nano, and I'll edit that text file and just say Hello World as just a message so that we know that this is the file that I've just created. Now **that's a file system that exists on one EC2 instance, and it's accessible over the network into the EFS product. So it's a network based file system**.

Now, if this was **EBS we would be limited to accessing this volume or this file system from one EC2 instance at a time. But that limitation doesn't exist for EFS** and I can demonstrate that. If I exit from application instance A, so I'm now back on the Bastion host, I'm going to move back across to my AWS console. This time I'm going to open a new tab to EC2 because I want to keep this one open. I'll go to my running instances. I'll locate application instance B so again, I need to make sure it's the running instance. I'll right click on it and hit "Connect" and again, I just want the username at the IP address. So I'm going to make sure I've got that copy that into my clipboard and move back across to my terminal. Once I'm there I'll SSH and then EC2-user at that IP address. I'll verify the authenticity by answering yes and I'm connected through now to application instance B. If I do a df -k you'll see that that file system is not mounted on this instance. So again I need to follow the same process so I'll do a sudo yum and then I'll install using -y the amazon-efs-utils package. I'll need to follow the same process and create a folder which will act as the mount point. So sudo mkdir/mnt/efs So I've created that and then I'll follow the same process again. So sudo mount to mount this EFS file system I'll need to specify that it's an EFS file system using -t space efs and then I'll need to get the file system ID. That's no longer in my clipboard, so I'll need to get that again, copy it, move back to the terminal, I'll paste that in :/ once again, just like last time and then I'll need to specify the local mount point. So that's /mnt/efs. Again, It might take a couple of seconds, but now it's mounted into that folder and if I do a cd /mnt/efs and then do a listing this time, we can see we've got already that test.txt and if a cat it to show what's in it, you'll see the message that I used for the previous instance. **This is a multi user file system. I can have it mounted on both of these EC2 instances, and it could be mounted on hundreds of EC2 instances.**

Now for the exam, it's not really important to understand how to use EFS. That's something that's more applicable at a professional level or maybe for the SysOps Associate exam. For the Solutions Architect Associate, you need to understand why and where you'd use EFS. **So EFS is designed for large scale parallel access of data it's designed for potentially thousands of NFS clients to mount a single file system and access the data concurrently. So it makes it ideal for parallel or elastic workloads inside AWS. So essentially, if you've got any data where you want to access it from a collection of EC2 instances in a shared way then EFS represents an ideal file system to use. Examples of this might be shared media or themes or any other shared data for WordPress instances. So we can finally deploy a scaleable WordPress platform and have access to a shared file system. You could also use EFS to store shared bespoke logging information. Now, normally, you would use CloudWatch logs for this, but there do exist certain scenarios where you want to use a shared file system rather than use CloudWatch logs. Perhaps if you got particularly tight security requirements and you don't want to utilize the CloudWatch logs product. Well, if you do that, you can use EFS in order to store these on a file system that's accessible only from instances which you dictate.**

**Now, EFS does support backups, you can utilize the AWS backup service in order to get data backed up off these file systems into something more resilient. You've also got AWS Data Sync, which can act as a synchronization product to get data in EFS.** Now, usage of both of these in detail is probably beyond the scope of the Solutions Architect Associate exam but I will be covering AWS backup elsewhere in the course. So I'll make sure that I show you how you can use AWS backup to handle the back up of EFS file systems. **From a resiliency perspective, the file system is region resilient. So the data restored across multiple availability zones. So you don't need to worry about a single AZ failure impacting your ability to access data. Remember, though we are creating these mount targets in multiple availability zones, so you need to make sure that you do create a mount target in multiple availability zones so that you can tolerate the failure of one of these.**

**Now some really good use cases for EFS: You might be using it for big data and analytics, where you need to access a file system from multiple locations. It might be useful for certain media processing workflows like video editing, studio production, broadcast processing, anything where you'd need a strong amount of data consistency and to be able to access that data from multiple instances. I've mentioned WordPress, so content management and web serving using a shared set of data or if you are deploying any sort of Linux system where you've got multiple instances, you can use it for a shared home directory platform. So rather than having home directories on each individual EC2 instance you could use an NFS or EFS share to be able to access them from one central file system and have them presented to multiple EC2 instances. What you probably wouldn't use EFS for is single machine situations, so it's probably overkill to use EFS if you've only got a single EC2 instance, it is not object storage, so you can't access any data on EFS, for example, from Cloudfront. It's just not possible. That's something where S3 is used and you wouldn't use it for temporary storage is not efficient. Generally, you'd only use EFS for shared access to data**.

Now, **EFS does support two levels of storage so two storage classes. We've got standard and infrequent access, and you can use lifecycle management just like you can with S3 to move files on the file system between those two different storage classes and I mentioned this at the start of the lesson but EFS security is controlled using security groups. You do need to make sure that you have the required ports to allow EFS inside a security group.** Remember when I configured this I used a security group that had already got created. So if we look at this application, B instance, this is currently associated with this security group, and the security group is set up to allow any communications from anything that's associated with this security group. So because I use the same security group for the mount targets for EFS, it automatically comes under this rule, so that offers a really nice way of getting it working. But you do need to make sure that if you are using separate security groups, you allow the port and protocol used for EFS in an incoming way from your instances to the security group that's associated with these mount targets and just again to reiterate for the exam, **you're able to access EFS from inside the VPC across the VPC peers and the from any corporate data centers, as long as you have a private networking configured.**

