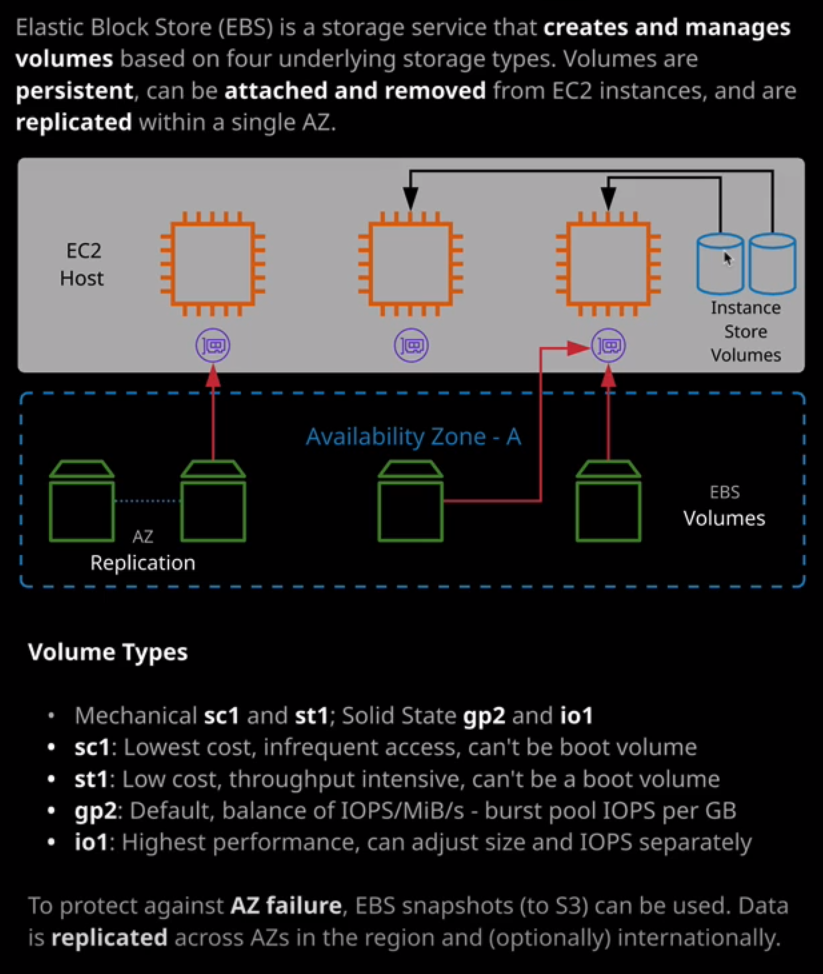
Welcome back and in this lesson, I'm going to be spending some more time going into the detail of storage architecture on EC2. I'm going to be talking about two different forms of storage: **instance store volumes which are the physical storage devices are attached to an EC2 host and available for the instances and EBS or elastic block store, which provides block storage to EC2.**



Now, **block storage is a type of storage which virtual machines can use as attached drives. So boot volumes and data volumes. If you've used a SAN or NAS before, that's block storage. EBS is a network storage product, so the volumes that you create in EBS are attached to EC2 over a storage network.** Now, before we talk about EBS, I want to talk about instance store volumes because they're a really important concept to grasp for the exam. Now instance store volumes are attached to an EC2 host. So if we look at the AWS documentation, we might have two EC2 hosts. Each of those EC2 hosts has one or more instances, and additionally, it has a number of physically attached storage devices known as **ephemeral devices, and these could be allocated to instances that are running on that host**.

Now **instance store volumes offer some of the best storage performance available to EC2 because they're directly connected to the EC2 host rather than being accessed over the network but that speed comes with trade offs because it's attached to the host, it means that if the host fails or changes, the storage is lost. Instance store volumes should be regarded as temporary. They're not resilient, and they're not persistent. They're included with the price of the instance, though, but only some of them.** Only **certain instance types come with included instance store volumes, a common example being the storage optimized instance types** that I mentioned in the previous lesson. So let's take a quick look at how this works, so if go to the EC2 console and I start provisioning an EC2 instance so I'll select launch instance. I'll pick the default Amazon Linux two AMI using the X86 architecture, and then I'll scroll down to an instance type that are no includes of instance store volumes, and that's the M5D instance type. I'll just scroll down a locate that one. The first one is the M5D.large, which includes a single instance store volume. It's a 75 GB SSD drive and I'll contrast that with the M5D24XL that includes four 900 GB SSDs. I'm not going to provision this, but just let's step through and see how it looks. So it's like that instance type. I'll go to next configure details, and then I'll skip past this because I won't be actually provisioning it and I'll go to add storage. **Now because it's an instance type that supports instance store volumes, you'll note that when I'm provisioning it, I get these four instance store volumes that are already prepopulated and these come included with the price of the instance. The volume type is listed as ephemeral, and that's how you know that it's an instance store volume**, **and then each of the volumes is mapped inside the operating system, using a different device ID. This is known as the block device mapping. So for each attached volume, whether it's instance store or EBS, it's going to be mapped to a device ID inside the guest operating system.** So I'm just going to cancel that. I'm going to go back following same process exactly as I did before, and I'm going to launch an M5D.large instance is slightly smaller, but it's still illustrates the point. So I'll select the instance, I'll go to next configure instance details. I'll be accepting all the defaults. I'll make sure it's using a public IP address. The storage I'll leave as default. Note that it's got this ephemeral volume listed in addition to the root volume. I go to next add tags. I'll configure the security group. I'm going to select an existing one that I've got configured already, which is set to only allow SSH from my IP address. So I'll review and launch click on launch. I've got a key pair already prepared so I'll acknowledge that I've still got that key pair and I'll launch the instance.

Now in the next couple of minutes I just want to use this time to show you exactly how these instance store volumes behave during certain activities on the EC2 instance. So what I'm going to do is I'm going to reboot the EC2 instance, I'm going to stop and start the EC2 instance and I want to show you exactly how this instance store volume behaves. So now that it's in a running state, I'll right click. I'll hit Connect. Then I'll fix up the permissions quickly. Then I'll copy the command to connect to the instance, move to my terminal and connect in. Now, don't worry at this stage, if you don't know Linux, I'm going to run a df -h command to list the file systems that the instance is aware of. Now the only thing we can say at this stage is this single root file system. So this is the EBS volume that's mounted to the instance by default. I can also run the command lsblk, which lists all of the known block devices. We've got nvme0n1 and we can see this is mounted to the root file system by this forward slash. We've also got this additional nvme1n1 device and that's the 69.9 GB SSD volume that's mounted. This is the instance store volume that I was telling you about moments ago. Now, to demonstrate this what I'll need to do is to mount it in the Linux file system. To do that, I'm going to run a sudo mkdir and I'm going to create a directory called /ephemeral. This is just a directory that I'll use to mount this volume to. So I'll make that directory, then I'm going to create a file system on this instance store volume. To do that, I'm going to run mkfs for make file system. I'm going to specify a file system type and use ext4 and then I'll specify the device ID and that's this nvme1n1. Now, don't worry if you don't understand exactly what these commands do. The main point of this lesson is to understand instance store volumes and we'll get to that momentarily. So I'll create a file system on this instance store volume that's mapped as this device. That'll take a couple of seconds and once I'm returned to the prompt it's completed. Then I'm going to mount this file system into this ephemeral directory to do this I'll use sudo mount and then /dev/nvme1n1. So that's the device that I just created the file system on and I'm going to mount that to the ephemeral folder and then we got this additional mount point /ephemeral, which is the instance store volume that's attached to this instance. I'm going to CD into that folder and I'm going to create a file. To do that, I'm going to run sudo and then nano, which is a text editor, and I'm going to call the file important.txt. I'll just put a message in here. Oh, my God, this is so important. I'll save the file, do a listing. We'll see that we've got that important.txt file that's on this ephemeral mount point. So this is now stored on the instance store volume. Now the first thing I want to do is I want to reboot this EC2 instance so I'm going to run sudo reboot. Now that's rebooting the instance from an operating system perspective. Essentially, it's just restarting the operating system, and this is going to be fairly quick to do and I'll be able to reconnect almost immediately. I'll just press up to retrieve the previous command and reconnect again. Now I have reconnected. If I run a df -k by default, it won't mount that volume because we've not configured it to but I can press the up arrow to get a command history. Find the command to mount that mount point, which I'll do now, so sudo mount /dev/nvme1n1 to ephemeral. If I go back to the ephemeral mount point to a file listing, we've still got our important.txt I can cat that just to demonstrate, and it's still got my message inside. **So by doing an OS level reboot, the instance store volume is maintained.**

This time I'm going to go back to the console, right click instance state reboot. Now this will force a reboot from an instance perspective. If I go back to my command line. Again, I'm disconnected. This time it will take slightly longer, but I should be able to reconnect after a few moments. So I've connected again, it's not mounted by default so I'll need to go through my history and just mount and volume move into the volume do a listing and again, the important.txt file is still there. Now the reason it's still there is because the EC2 instance through both of these reboots has stayed on the same EC2 host. I can force it to move to a new EC2 host. If I right click instance state and stop, **This will stop the instance. So, in effect, shutting down the guest operating system and the instance so the virtual machine hardware but what it also does is deallocate it from this EC2 host. When I started up again, it's going to be running on a separate EC2 host and that means new ephemeral or instance store volumes. This is one of the downsides of using instance store volumes because they're associated with an individual EC2 host if for whatever reason that instance moves between host or if these volumes fail, then you lose access to those original volumes and you get new ones allocated on a new host.** So I'm going to refresh this right click again instance state and then start. **When it starts again, it's going to have a new public IP address and a new DNS name** so I'll need to wait for it to complete in order to get the connection command. I'll right click, hit Connect. I'll copy the new connection command, move to my terminal, paste this in, and press Enter. I'll need to accept the authenticity again. I'm connected. This time I can scroll up in the history and attempt to mount this ephemeral volume again but when I do, I get this error. It says run file type. Bad option. Bad superblock on this device. Now the reason for that, as I mentioned a second ago, is what I'm attempting to mount here is a brand new instance store volume. It's not the same one that I had before, because it's on a brand new EC2 host. **So this is one of the downsides of instance store volumes. They're super fast, but the data is not persistent, so you can't use them for anything where you can't afford to lose that data.** For the exam, keep that in mind for **certain performance levels you need instance store volumes, but you need to be aware that for any data that you can't reproduce so any permanent data, anything beyond temporary storage or cached volumes or temporary databases, it's not really suited to instance store volumes**.