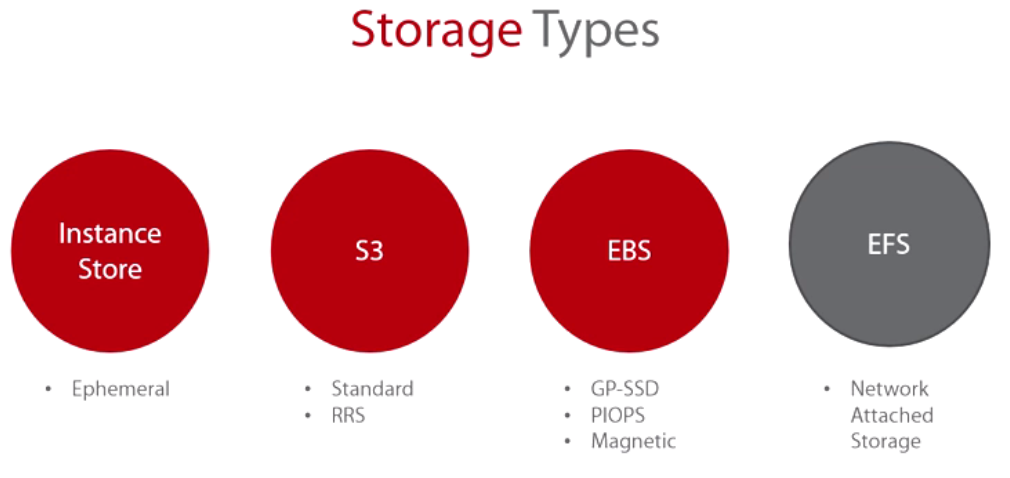
**Working with Elastic Block Storage**

**Module Introduction**

If you thought compute was the only layer of the stack that was important for performance, think again. Whenever you're dealing with performance specifically with applications, you have to make sure you're understanding the characteristics of the storage that it's being run on. Elastic block storage presents storage to your EC2 instances.

**Understanding AWS Storage Types**

AWS as a platform is ever-changing. It's very dynamic. It's always adding new features and capabilities. And so is the case for storage types.

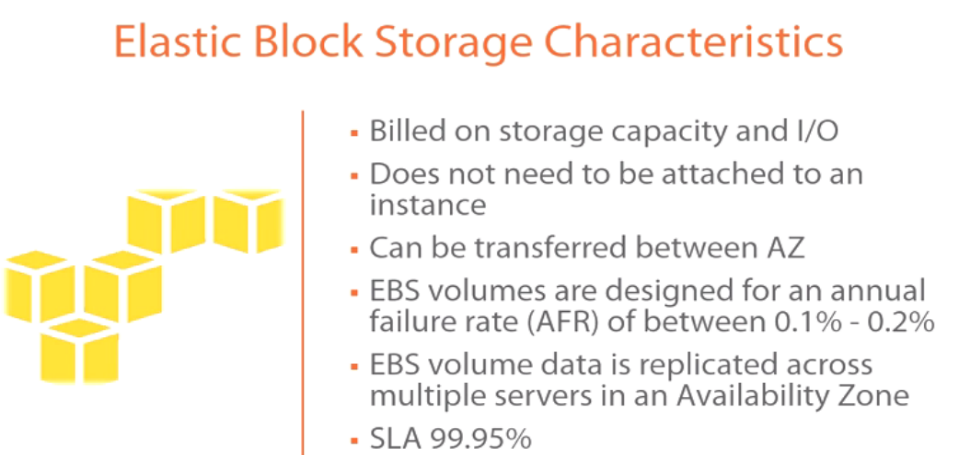


Instance store, S3, EBS or elastic block storage, and we also have EFS or elastic file system.

* Now instance store is ephemeral storage. This is temporary storage.
* S3 is the object-based storage. It comes into two flavors. You have the standard and then you have the reduced redundancy storage.
* EBS you have general purpose SSD, provision IOPS, magnetic or spinning disk, and the
* EFS is network-attached storage. EFS is a recent addition to the storage-type family. It wasn't available back then. It's outside of the scope of this particular course, but what it is essentially is a managed file system. It's based on NFS4, and today it only supports Linux instances. But essentially you can attach it to a Linux instance, and the file system itself will grow and shrink as you add and remove data. It's very cool. It's under preview.

**Elastic block storage characteristics.**

* Let's start with the fact that it's billed on storage capacity in I/O. So while you're getting billed for the instance by the hour, you're also being billed for the storage, how much you're using from that storage, how much I/O performance is associated with that instance, that's also---you're also paying for that.



* It does not need to be attached to an instance, which means that you can detach a volume from the instance, delete the instance if you want to, create a new instance, and reattach it. So it has its independence. It can be attached, but it also can live without having an instance associated with it.
* It can be transferred between availability zones, so you can move it around between availability zones if you need to attach it to a different instance that's in a different availability zone.
* EBS volumes are designed for an annual fault or an annual failure rate (AFR) between 0.1% and 0.2%. So you can see here they're going to be extremely highly available. The likelihood of this failing is very, very slim.

When you compare this with the average failure rate for a hard drive, it's like 5% or 10% I believe, so you see that the percentage is huge here.

* EBS volume data is replicated across multiple servers in an availability zone. It's actually replicated three times to three different servers. So, again, every time you write anything, that particular write is written to three different servers in that availability zone, again, to make sure that you're getting the highest level of availability possible.
* The SLA is 99.95% as well for those of you that are interested in knowing these SLAs and just in case it comes up on the exam as well.

Let's take a look at the **EBS volume types.**

There are three EBS volume types-- general purpose SSD, the provisioned IOPS, and the magnetic.

**What to use these different volume types for, like what types of use cases??.**

For example, the **general purpose SSD---**by the way, that is the default volume type. So previously the default volume type was magnetic. Today the default volume type is GP2, which is a general purpose SSD. The use cases are system boot volumes. Virtual desktops are great use cases for it. Small to medium-sized databases are also a good idea. **Development and test environments.**

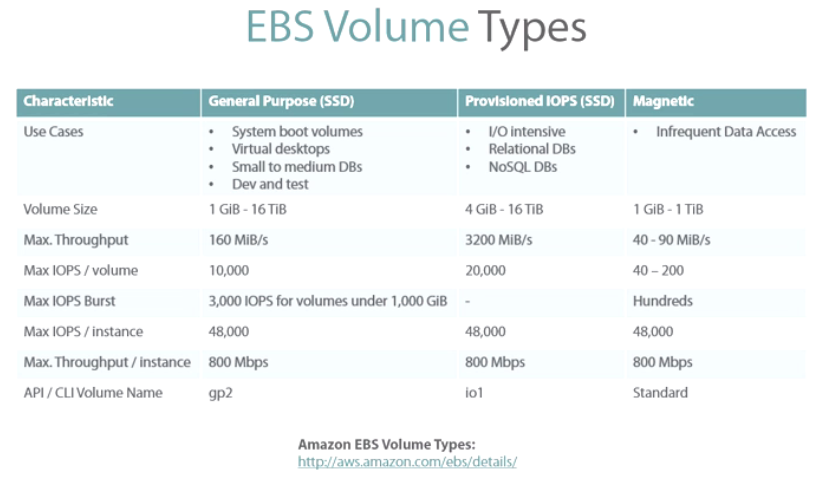
Conversely when you look at **provisioned IOPS,** provisioned IOPS are guaranteed IOPS. They are predictable IOPS. So those have another use case.

For example, anything that's I/O intensive, relational databases, anything that's a NoSQL database--all of these higher-end databases that you might want to run on AWS, provisioned IOPS might be your best bet.

**Magnetic.** To be honest with you, it's infrequent data access. It's almost archiving. I know it's funny, but I still think there's a lot of use cases that you can probably get away with magnetic. But the fact that SSD has come down in price and it's the default on AWS, you'll probably want to use that. However, magnetic is still available.

And we'll talk about some different ways of improving IOPS using the general purpose SSD and using also magnetic if you wanted to avoid provisioned IOPS because provisioned IOPS come at a very costly, more expensive price of course.

**Now volume sizes,** you can---for the GP SSD, the smallest is a GB, the largest is 16 TB. You can stripe with those so, again, you can take several of these volumes, put them together to get this large volume. The same thing for provisioned IOPS--4 GB to 16 TB.

And then magnetic is 1 GB to 16 TB as well. 

These volumes also have the **ability to do bursting**. So, for example, general purpose SSD and the magnetic have bursting capabilities. Provisioned IOPS does not.

What bursting basically means is every now and then, you're able to burst. So if you have---so 3000 IOPS per volume that's under 1 GB, so typically when the machine powers on, it will burst so that it powers on very quickly, and then it will settle down at lower IOPS levels.

And the same thing is for the magnetic as well. Now the magnetics will go into the hundreds of the IOPS where when you’re bursting with GP SSD, you're in the thousands.

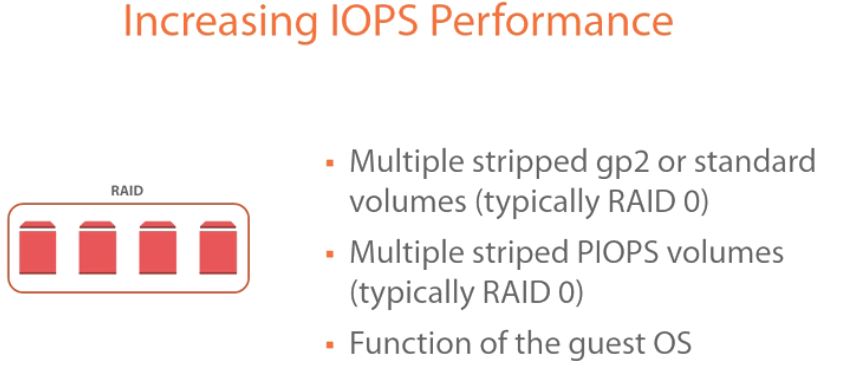
Now the one thing that I wanted to draw your attention to here because it may be confusing, may not depending on how you read it, but the max IOPS per instance can go up to 48,000 across the board. Now, I previously said that the max IOPS per volume was 10,000, but keep in mind that's one volume.

I can take several volumes and stripe them together. However, the maximum IOPS that a single EC2 instance can sustain is 48,000. So---and that will also vary because you'll have to maybe add CPU and memory to sustain that much I/O. But, again, that's the maximum that you can go to. Maximum throughput is 800 almost across the board. Now with the magnetic, you just won't be able to stripe that many. You'll never be able to get the IOPS that high, but theoretically it's there as well.

**Increasing IOPS Performance**

So I mentioned earlier that there is a method by which you can use to increase the IOPS performance. And what I mean by that is striping.

So you can create one volume, two, three, four, multiple volumes. And what you can do then is put them in a RAID group, RAID 0 for example. And as a result, you are combining the I/O performance of all of these drives and arriving at an IOPS level that is acceptable to you. This might be a cheaper alternative to provisioned IOPS.

Remember provisioned IOPS are predictable, but they also come at a cost, so they can be very expensive. **So instead of doing that against a provisioned IOPS, maybe you take several general purpose SSD volumes, you put them in a RAID group, and then you're getting the same level of performance.** Now keep in mind, when you're doing RAID 0, you always risk losing one hard drive. And as a result, you're going to lose the entire RAID. Nothing is there to stop you from doing RAID 5 or RAID 6, etc. Again, RAID is going to be a function of the operating system. v

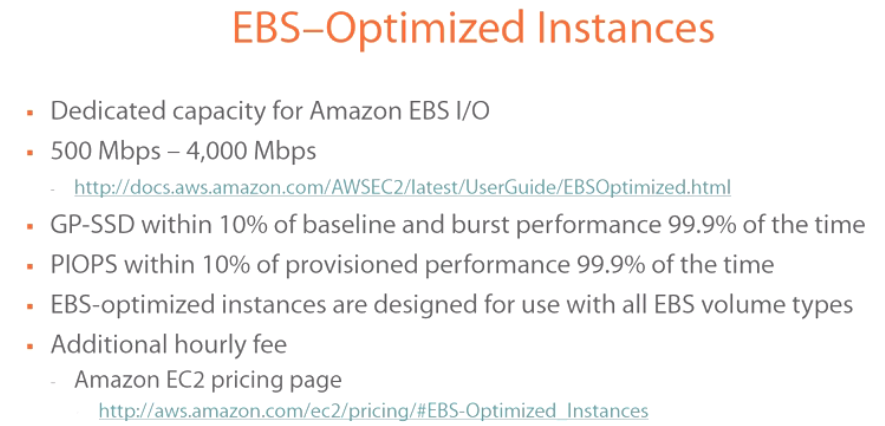
* you can take multiple stripped GP2 or standard volumes, typically RAID 0 but nothing will stop you from doing RAID 5 or 6, etc.
* For multiple striped provisioned IOPS volumes as well. So, if the maximum provisioned IOPS per volume isn't enough, you can also take multiple provisioned IOPS volumes and stripe them together. Again, it's typically RAID 0, but you can always do different RAID levels.
* This is a function of the guest operating system. So this isn't something that you're going to configure within the AWS platform. Once you are attaching these volumes to the instance, if it's a Windows machine, for example, you're going to disk manager, and you're creating the RAID at that level. So keep that in mind. Now, again, with RAID 0, you lose---you risk losing a drive. And as a result, you lose the entire RAID, which is why you might want to consider RAID 5 or others.

But keep in mind what we talked about earlier--the average yearly loss rate is 0.1 to 0.2%, so it is very, very low. RAID 0 is going to give you the highest level of performance compared to 5 or 6 or others. So keep that in mind as you're building these. This might take some time for you just to build a comfort level with AWS, just to get that feeling that this is going to work, this isn't going to break, etc.

Now also keep in mind, EC2 instances from a storage perspective aren't designed for you to have long-term storage. They are designed for you to have those instances as disposable. If you wanted to have long-term storage, you would use S3. That has the highest durability level, etc. So keep that in mind also. You have to have a different mindset when you begin to deploy on AWS than you had when you deployed on-premises. On-premises we built everything to never fail. In the cloud, you have to build everything with the expectation that it is going to fail.

If you can accept this, then RAID 0 and this type of an approach becomes very, very appealing to you.

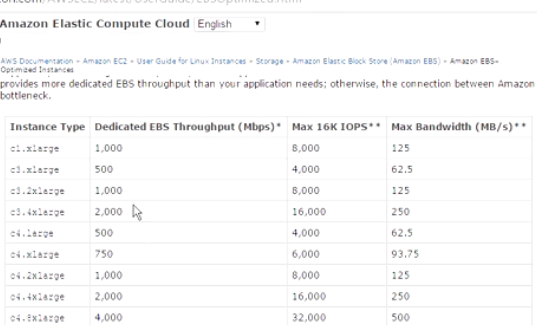
Again, you have the option of doing provisioned IOPS and saying, Hey, I need 200 or 300, 400, whatever, 1000, 2000 IOPS. You can specify that, and that will be provisioned. That's a predictable IOPS level that you can use for your applications. Or you can stripe multiple general purpose SSD volumes and get a higher level of IOPS as well. In the earlier slides I showed you kind of the levels that you can go up to from a general purpose, and you saw you can get pretty high up from an IOPS perspective.

**EBS-optimized instances** 

* Now while increasing IOPS performance is important, you also have to keep in mind that **optimizing your instances to leverage better IOPS performance is also important,** which brings us to EBS-optimized instances.

And what it's doing here is we essentially dedicate---you're dedicating capacity for Amazon EBS I/O. So it's not enough to just improve the IOPS performance, you have to also improve the I/O that's going to enable you to drive that level of IOPS to your instance. So that's important.

* Now EBS-optimized instances are not available for all EC2 instance families.

**So, for example, you couldn't enable optimized instances on a micro family type.** So if you're provisioning a micro T2, for instance, you wouldn't be able to enable this. Now when you do enable this, you're going to get a performance boost anywhere from 500 mbps to 4000 mbps.

Let me show you the instance families that are supported and kind of what you should expect from a burstability perspective or an optimization perspective. So if you quickly Amazon EBS-optimized instances, you'll get to this page.

This page essentially tells you the instance type, the family that is supported. So, for example, you won't find any Ts here. So you can scroll all the way down and take a look at which of these family types you can enable EBS-optimized instances for, and it also gives you the optimization level that it can accomplish on this particular instance. So keep this in mind as you are configuring this.

**But, again, I just want to make sure that you are aware this isn't available for all instance families.**

🡪The GP SSD--now what it's doing here is you're able to optimize within 10% of the baseline and burst performance. And this is available 99.9% of the time. So you're able to optimize the instance from an I/O perspective 10% of the baseline and burst performance. Whatever the baseline performance is, 10% of that is the expected optimization that you can expect. And you're able to do this 99.9% of the time, and this is on a yearly basis. So almost all the time you can do this.

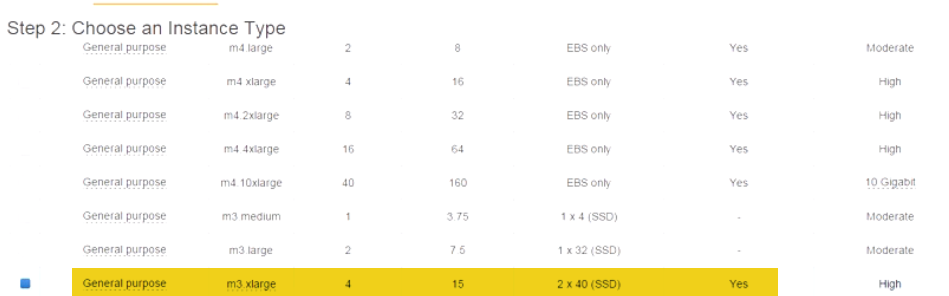
🡪From a provisioned IOPS perspective, it's also 10% of the provisioned performance. So whatever you've provisioned from an IOPS perspective, 10% of that is what you should expect. And you should also expect to be able to do this 99.9% of the time every year.

🡪Now EBS-optimized instances are designed for use with all EBS volumes, which means you can do this with GP SSD provisioned IOPS, and you can also do this with magnetic. Now, it does come at an additional hourly fee, so I would encourage you to check the Amazon EC2 pricing page to see what in addition to the regular hourly fee that you would have to incur to get this optimized performance. But in some cases, this might be worth it because of the workload that you're trying to drive from this EC2 instance.

**Configuring Optimized Instances**

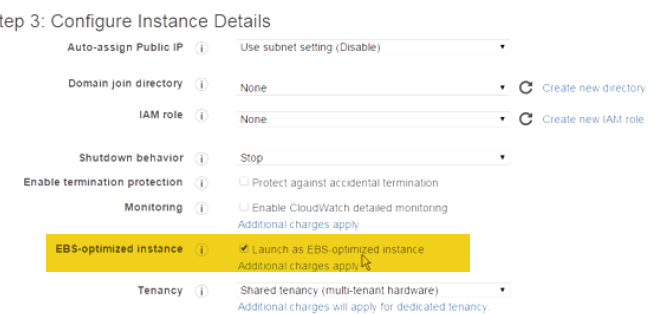
From the Management Console, the way to enable EBS-optimized instances is 🡪 go to EC2🡪launching an instance. We're going to pick on the same AMI that we've been using, which is the Microsoft Windows Server 2012.

**And because EBS-optimized is a function of the EC2 instance family, we have to choose a family that will support it.**

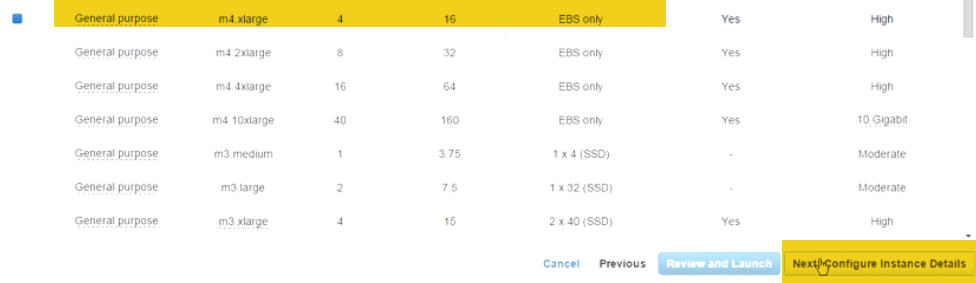
So, for example, let's go with an m3 extra large,  and let's click on Next. If I scroll a little bit, you'll find that you have the option here of doing EBS-optimized instances.

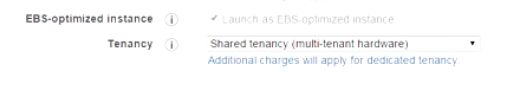
If you select that, badabing, badaboom! Now you can go back to this previous page that I showed you and scroll down to the m3 extra large, and you'll find exactly how much it's going to boost the performance for you.



**Now keep in mind there are some instances that are going to have the EBS-optimized instance checkbox enabled by default.** It's just a function of how this particular instance has been created. So you don't have the option of disabling it. 

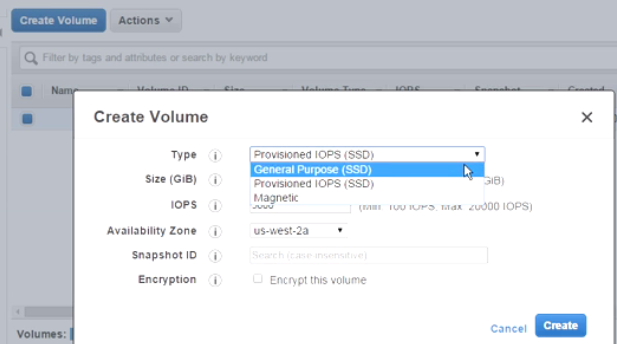
They are basically presenting you a virtual machine or an instance that is optimized to deliver a certain level of I/O and a certain level of IOPS. As a result, you cannot enable or disable it.

🡪But, for example, the m3 extra large here we have the option of enabling that and disabling. Let me show you what I mean by it'll have it enabled by default. 

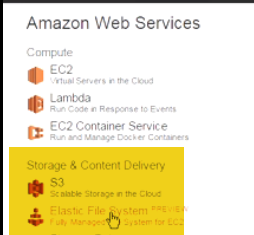
🡪 for example, let's go with the m4 extra large. I believe this one has that enabled by default. See? Under EBS-optimized instance, this one is already enabled. You don't have the option of disabling it at all. 

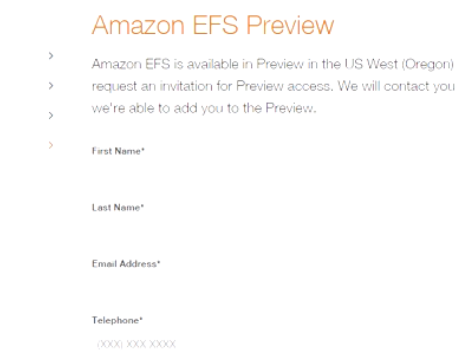
🡪Because general purpose SSD is now the default, let me show you how you would create a volume and how it's defaulting to it.

So, right now I have one particular volume here. And this volume is created as a result of the AMI that we deployed. It's 32 GB in size.

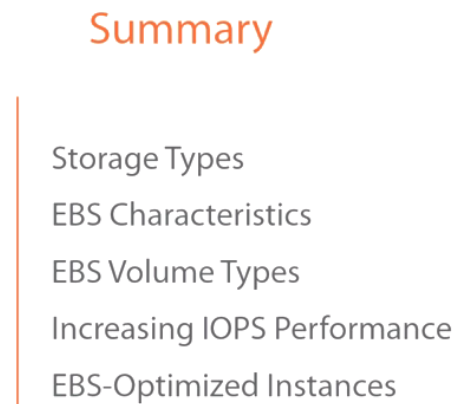
If I go ahead and click on Create Volume, it gives you the option---first of all, it **defaults right away to general purpose SSD**. And even before that, just so you know, gp2, so by default when we created this AMI, it automatically defaulted to general purpose SSD, and this is the terminology that they use for it. 

You’re finding it here from the drop-down menu, you can go with provisioned IOPS, and you can go with magnetic if you wanted to.

**Now the other thing that I said I would show you is how to enable elastic file system**?? EFS is a new service. It's a new storage service that AWS is making available. It's a network attached storage. Now currently it's in preview. However, the way to access it is if you go back out to the main Management Console, what you're looking for now is storage. And under Storage what we're looking for is the elastic file system.  If you click on the elastic file system, if you have not filled out the form yet, you have to request to be invited to use EFS because it's still in preview. Once it's out of preview, you won't need to do this step anymore. So you fill out all the information. You click on Submit. And once AWS approves it, then what you can do is you can go out there and start creating a share.

Now, remember, EFS is a fully managed file system. It grows and shrinks as you add and remove data from it. It's based on NFS4. Today it's only supported on Linux instances, but that might change. There's no snapshot and copy as of this recording. But, again, a lot of this could change in the future. But EFS, again, it's great for not having to manage the file system anymore.

**Module Summary**

* So what did we cover in this lesson? Well we started out by talking about the different storage types. We talked about Ephemeral, S3, EBS, EFS, and the different components of each.
* EBS in particular, we talked about the fact that you have general purpose SSD, provisioned IOPS, magnetic.
* With S3 we had the standard and the RRS so reduced redundancy storage. Ephemeral is temporary storage. EFS is network attached. 
* We then talked about the different EBS characteristics, what you can do with it, what you can't do with it, some of the limits, the SLAs, what you can expect from a failure rate per year, etc.
* We talked about the different EBS volume types, and we compared them side by side in a table from a performance perspective, maximum IOPS, maximum throughput, the size, the volume size, and so on and so forth.
* And then we talked about methods by which you can increase the IOPS performance. Specifically what we talked about is how you can take several volumes, multiple volumes, stripe them together, and this is a function of the guest operating system, and as a result combine their I/O capabilities to get a larger IOPS performance level.
* We finally talked about EBS-optimized instances. And what EBS-optimized instances allow you to do is increase the I/O, the network I/O, not just the IOPS level but also the IO that's driving that IOPS. And we went through the demonstration, and I showed you where you can enable that, and
* we also showed you how you can enable or how you can start using EFS although it's in preview and a little bit outside of the scope of what we're covering.

**EBS Snapshots and Replication**

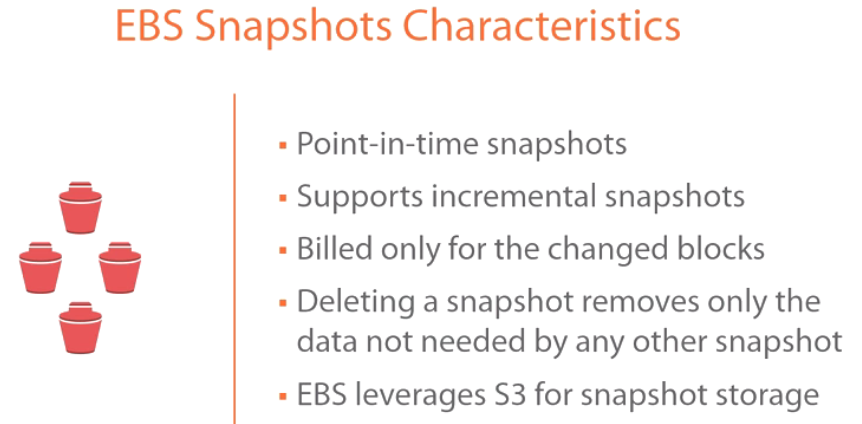
**Module Introduction**

Having the ability to backup your volumes is very important not just from a recoverability perspective but sometimes because :

you want to be able to back them up and move them to another region.

Maybe you want to share them.

You just want to be able to make them mobile.

**EBS Snapshots Characteristics**

🡪A snapshot is a point-in-time snapshot, which means as soon as you click on the snapshot icon, it's going to freeze that instance, and it's going to take a copy of it or it's going to freeze it and make a copy of it immediately. That's a point-in-time snapshot.

That is not going to take into regard anything that's going on on that server. So if you have a database and something's being written to that database, you are probably not going to have a clean cut. You might have to recover some of these tables or do some management to that database once you snap back or use a revert to snapshot. So **it's not a clean cutover, which is why in a lot of cases, you either want to power down that instance and then take a snapshot because you're ensuring that nothing is running on that instance when you take the snapshot.** So that's kind of a clean way of taking the snapshot. So just keep that in mind.

🡪Taking a snapshot of a running instance will take the snapshot as soon as you click on that button. Now recall I said there was some good news and there was some not so great news. Now, again, for exam purposes, it's all great news but from a realistic perspective.

EBS snapshots support incremental backups, which is really cool because then you don't have to keep taking a full snapshot every time. It's only taking a snapshot of the blocks that have changed, which is great. But if you've ever worked with backups, you'll know that incrementals aren't always the only solution to use, and they're not going to protect you against everything else. So the only thing that you can do from a snapshot perspective is an incremental.

If you wanted a more robust backup solution, you would have to go to third party and maybe even revert back to an agent-based within the actual instance to do a full VM backup. Or if you're trying to restore certain files within the VM, again the snapshot won't allow you to do that. So that's where this particular method won't be very useful if you're trying to restore a file within the instance or within the snapshot as opposed to just reverting everything back.

So keep that in mind. It does provide a level of protection. It does provide some backup. But if you want more granular, a lot more control, then you probably want third party, and you might have to revert back to an agent.

🡪Now it's built only for the changed blocks. This is really cool because what you might end up doing is taking several snapshots of a particular instance. Now if a snapshot is, let's say, 10 GB in size and the changed blocks are 1 GB in size, then it's only going to bill you the second time for that 1 GB. It's not going to bill you every time for 10 GB and then 11 GB, which is really cool because if the initial snapshot is 10 GB, the second time you take a snapshot it's only going to bill you for those changed blocks.

🡪Now deleting a snapshot removes only the data not needed by any other snapshot. This is really important. If you have five snapshots that you've taken, traditionally what we're used to is make sure you don't delete the initial, the first snapshot because that contains the information needed to be able to boot back the operating system for that machine to function properly.

What AWS does is it ensures when you're deleting any snapshot that you---that it has enough information to revert back and make sure that snapshot is functional. So when you're looking at your snapshot change, we don't worry about, Well, should I delete this one first or not delete this one? Because it relies on it, it doesn't matter.

AWS knows, and it retains enough data to make sure that any snapshot that you revert has enough data to function properly.

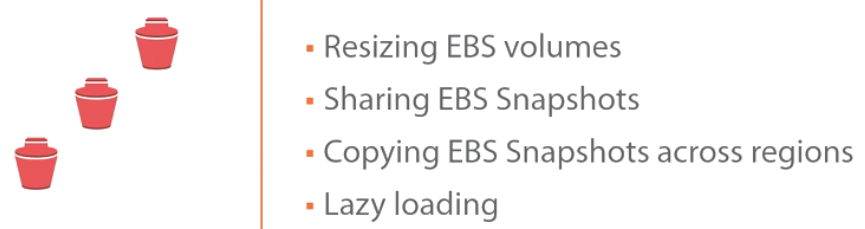
🡪EBS leverages S3 for snapshot storage. Now what does that mean?

Well when you take a snapshot**, EBS has to store it somewhere. S3 becomes the storage area for EBS.** Now that's funny but think about it. **EBS is only storage intended for what?** For EC2 instances. That's the only purpose it has there. Anytime you want to use any kind of long-term storage or anytime you want to store anything that is not intended for instances, well then you would store it on S3.

In this case, EBS needs to store a file. What is it going to use? It's going to use AWS's storage service, which is S3. As a result, keep that also in the back of your mind that **EBS leverages S3 for snapshot storage. Every time you take a snapshot, it's going to place it on S3.**

If we goto S3 bucket, after Taken an EBS snapshot, you will not find the SD EBS snapshot there. You won't be able to see it. When we go into the Management Console, I'll show you where in the Management Console you can go to see your snapshots. It's essentially under the storage of an EC2 instance. There's a snapshot section that you click on.

So while it stores it in S3, it doesn't necessarily store it in your bucket, so you won't be able to move it or apply any kind of actions against it from your bucket. So keep that in the back of your mind as well.

**EBS snapshots features**. ****

🡪Now it's good for resizing EBS volumes, and I'm going to demonstrate all of these when we get into the Management Console. But let's say you have an 8 GB boot drive, and you wanted to resize that. Well, it would be really difficult for you to resize it. You would have to delete it, create a new one with the size that you want and maybe copy all the contents, etc.

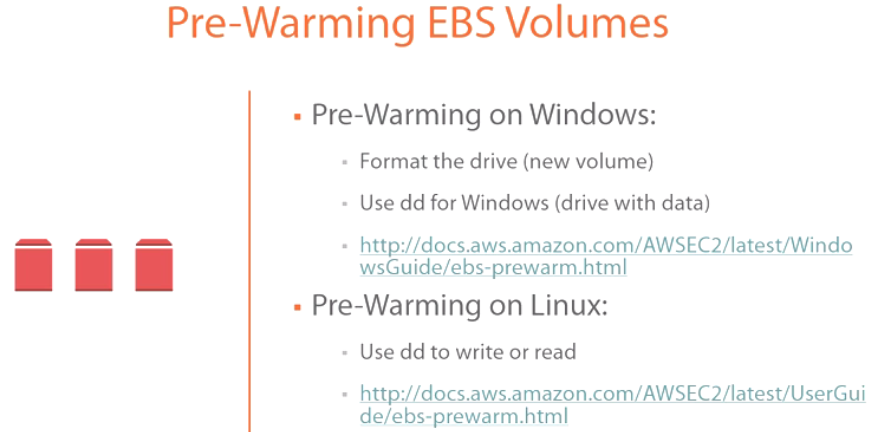
Well you can take a snapshot of that volume, and while you're taking the snapshot you can assign it a new size. So let's say you're going from 8 GB to 32 GB.

And then you can detach the old volume, attach the new volume which, remember, is an exact snapshot of the 8 GB one, and all of a sudden you now have a 32 GB volume. So that's kind of an easy and quick way of resizing EBS volumes.

🡪You also can share your EBS snapshots. So if you have colleagues, if you have maybe another organization, if you just wanted to put it out there in the community and you wanted to share a particular snapshot maybe because you've customized it, maybe it's a locked-down hardened version, you can make it available so others can use your snapshot to provision an AMI and to provision instances from it as well.

🡪You can also use this snapshot to copy it across regions. So, remember, snapshots are available within the same region. Availability zones will have access to them. You can create instances in several availability zones and use snapshots to kind of connect to them. If you wanted to make these snapshots available in other regions, maybe you are a multinational organization that's using AWS in several different regions, then you can use snapshots that copy, and I'll show you how to do that as well.

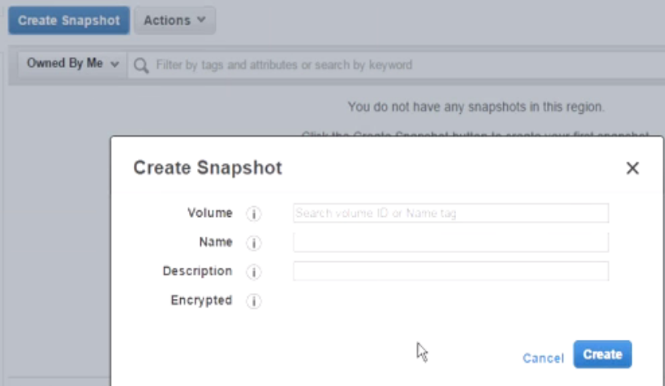
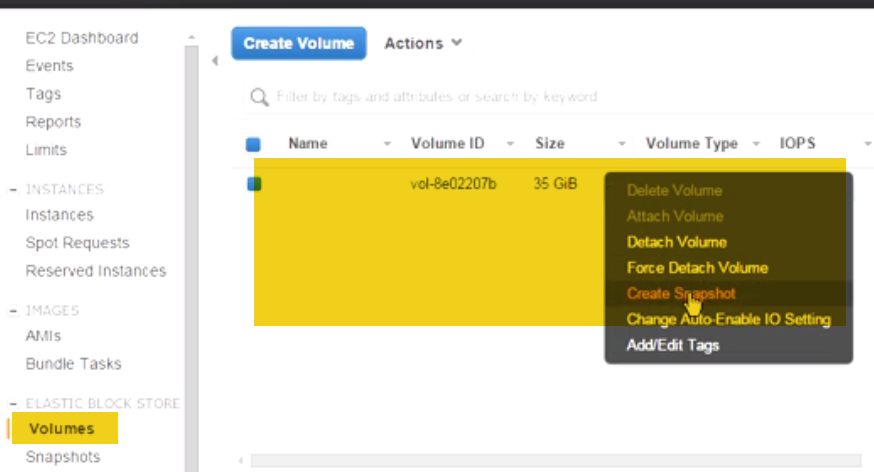
🡪 Now one thing that's interesting about snapshots is that it has something called lazy loading. Now **what is lazy loading?** Now remember when we talked about the fact that when you take an EBS snapshot, that snapshot is stored on S3. When you go back to recall that snapshot because you want to attach it to an EBS volume, the recall process is going to be slow, so you're going to get the initial bits, enough bits to power on that instance, but it's going to be very slow because, again, it's getting information out of S3. So it's going to be very, very slow.

****

There are ways of getting around lazy loading in order to ensure that you're boosting the performance of your virtual machine or your instance, and I'll get to that in a second. However, in most cases as you are recalling this data, most of us--- it's not going to matter. We're not going to know the difference. It's going to be fine. However, in the event that we do not want performance to degrade at all and we want to avoid this lazy loading process, we can go to something called a pre-warming of EBS volumes. Now there's a process for doing this on Windows, and there's also a process for doing this on Linux. So if the snapshot that you are recalling happens to be empty volume, formatting the volume will accelerate your access, your IOPS level access to that volume because you're writing across all of the blocks in that volume. And as a result, you're going to be accessing all of those blocks so S3 is going to be sending all of those blocks much, much quicker. If there's data on the volume that you want to preserve, then you want to use something like dd for Windows, and you can download that. It's a free download. And what that will do is it will read all of the blocks at the same time. What that forces S3 to do at that point is to send those blocks a lot faster. The same thing applies for Linux. If you didn't care about the data that's on there, then you can just wipe it, so you're writing across all of the blocks. If you did care, then you can use also dd to read across all of the blocks, which will accelerate a recall process from S3. Now, again, this is because that snapshot resides on S3.

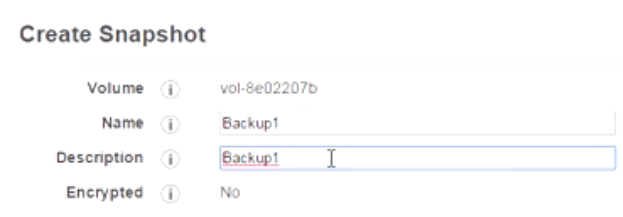
**Working with Snapshots in the AWS Management Console**

So to access snapshots: get into EC2. 🡪Under storage that you have access to snapshots. So if you click on it, right now it's empty. We have not taken a snapshot of anything yet.

1. You can create a snapshot right away doing it this way, or 
2. You can also go to the volume that you want to shapshot, that you want to protect.. 

you can right-click on it, and you can go to Create Snapshot.

, let's call it Backup1, for example. You can also give it a description..

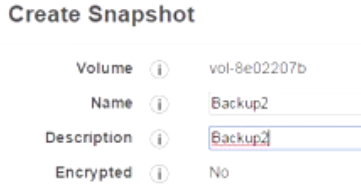


Now the first time that you take a snapshot, you'll find that it might take a little bit of time for this snapshot to create. If it does take a little bit of time, just wait for it depending on the amount of data that it's creating.

The second time you take a snapshot, and again I don't have anything here that I've modified, but the second time you take a snapshot, it's going to be instantaneous.

So if you click on Create Snapshot here again.

Volumes, right-click on it again, and do Create Snapshot, and this time let's do Backup2.

Whoops, I've got to be able to be in the right spot. Backup2, and let's give it the same description. 

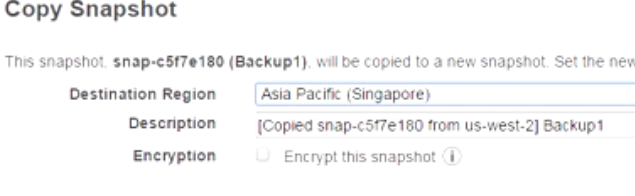
If it took time the first time you did this, the second time you do this, it will be very, very quick.

So you see it's almost instantaneous.

🡪 Now recall when we were talking about this during the presentation, I said that **you can pretty much delete any snapshot in the chain, and you don't have to worry about being able to revert back and that functioning properly.**

So right now this was the initial snapshot (backup1). I can delete this very easily and revert using this. AWS is going to maintain all the information, all the data necessary in order to make sure that Backup2 is going to power on properly.

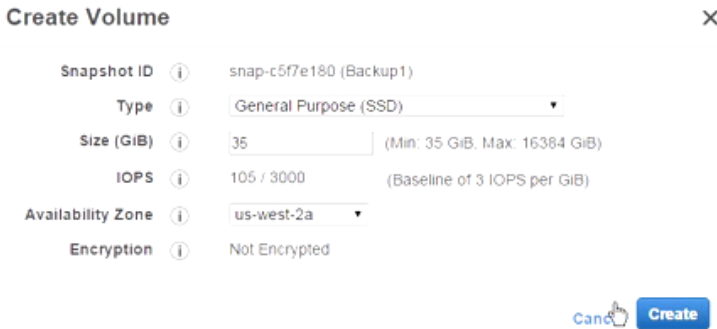
🡪Now, if you wanted to copy a snapshot across regions, if you wanted to be able to move it, to make it portable, it's very easy to do also. You essentially right-click on it, and Copy, you have the  Destination Region. Depending on the region that you're trying to target--- I mean, look at this, it's going to give you all the regions that AWS has access to, so if you wanted to copy this to Asia Pacific, you can do that. You can give it a description if you wanted to. Are you going to encrypt this or not? And then you click on Copy and badabing, badaboom,

 **once you switch back to that region, you will now have access to that snapshot there. And then you can associate it with instances, and you can create images out of it as well.** So it's going to take time.

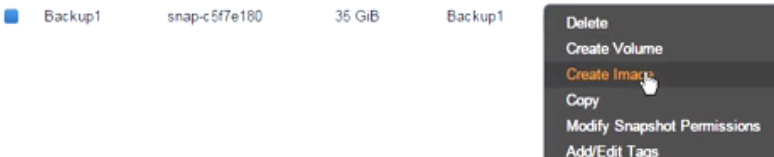
🡪However, what I want to do is **assuming this was moved onto a different region, and you're in a different region now,** what you can do is essentially right-click on it, and you can either 

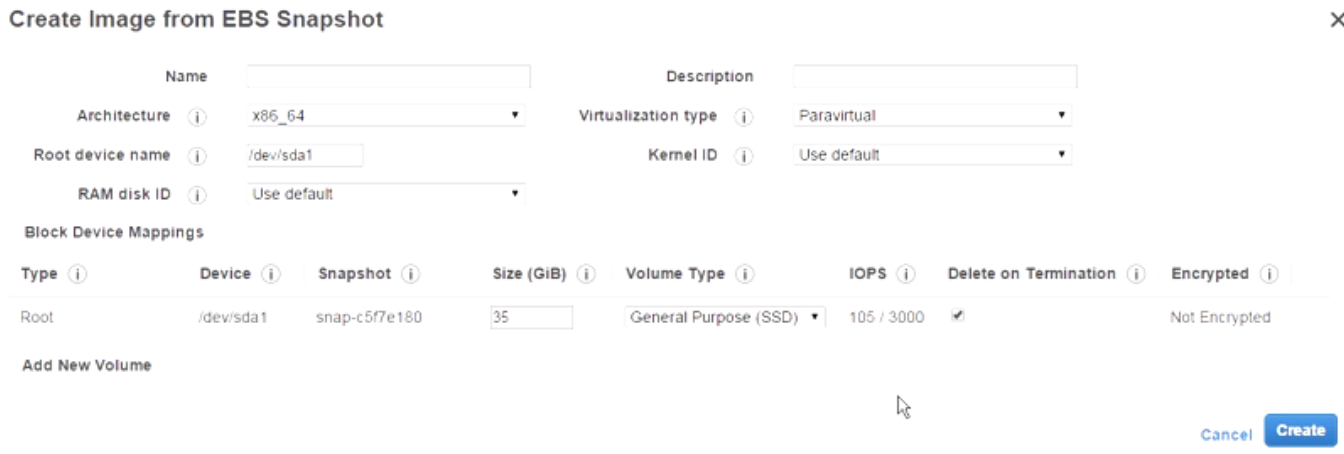
Create a volume from this, and at this point, it gives you the option of,

Okay, well what type of volume do you want to create? And, again, you're familiar with this. Either the general purpose SSD, provisioned IOPS, magnetic.

You can specify the size, etc., etc. And you can just create it.

🡪What you can also do is if you right-click on it, you can create an image out of this thing. So now you're customizing an AMI image for this thing.

So you can give it a name. You can go through and customize everything, and then you are creating it.

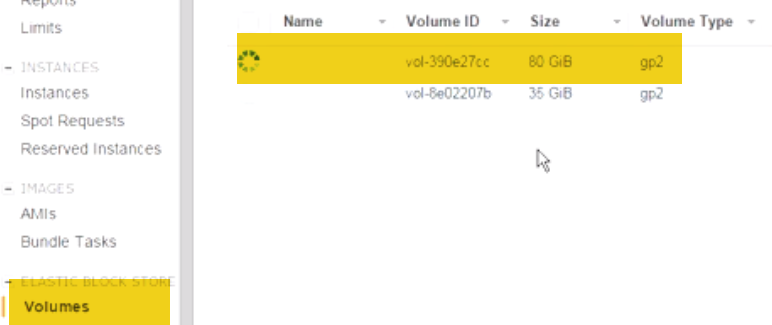
**So that gives you the ability to then deploy this image very easily. Essentially it would become a catalog option for you to deploy for when you go to launch EC2 instances.**

You can also make this available either through the community or you can make it available to your colleagues to deploy instances from it.

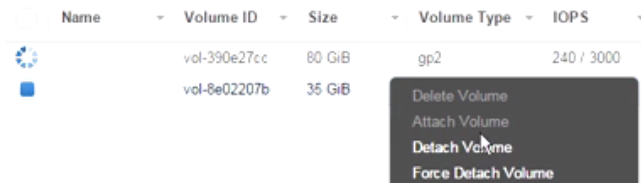
🡪Now recall I also told you that **you can use snapshots to kind of modify the size of the actual volume.** So right now we have---this snapshot is 35 GB in size. Let's say we wanted to make this 70 GB in size or 80 GB in size for whatever reason. 

What you do is you right-click on it, and you click on Create Volume. Now as you're going through the process of creating this volume, you can give it a different size. So, for example, you can say, I want this to be 80 GB.

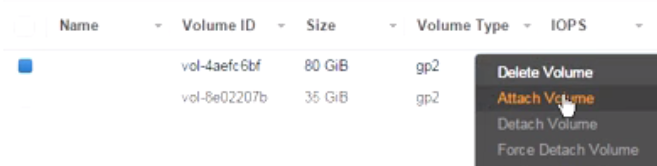
And you can click on it, and **within a matter of seconds, minutes,** whatever the case is, it's going to be working its magic here, you now have an 80 GB volume.



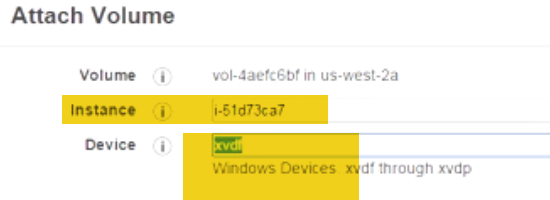
🡪 Now how do you attach this 80 GB volume to the instance?

1.) First of all, you want to detach the **instance or the volume that's currently attached to it,** so you come down here and say Detach Volume, Yes, Detach. (30G is already attached)

2.) Now what you want to do is select the first volume, which is the one that we just created that we want to attach. We're going to right-click on it, and we're going to click on Attach Volume.



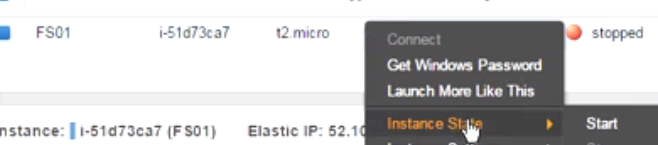
Now in the Attach Volume under the Instance, we're going to select the only instance that we have.

Now pay attention to Device. The device has to be mounted as the first device. Otherwise, your instance will not power on. 

So what **you want to do is remove this value that they add by default** and, instead, put the following value(/dev/sda1) in in order for your instance to boot up, in order for the device to be recognized as kind of the boot device. 

**So just keep that in mind.**

we've clicked on Attach, let's go ahead and see if our instance is actually going to power on.

We're going to right-click on it. We're going to come down here to State, and we're going to click on Start, Yes. And would you take a look at that? **Now it's actually booting up, and we've resized the volume very, very easily.**

**Module Summary**

**🡪EBS** snapshots characteristics: it's a point-in-time snapshot.

If you click on the snapshot button, it's going to freeze the VM, the instance, and take a snapshot immediately regardless of what's going on within the instance, which is why you probably want to take care when you're taking a snapshot. Maybe power it off to make sure that there's nothing open, no open files, no tasks that might be---that might interfere, etc. **We talked about the fact that snapshots are also intervals. So while it can be a backup strategy, it's not necessarily a rock-solid backup strategy. You cannot restore a file from a snapshot.** It's an all or nothing type of proposition. So also keep that in mind.

🡪We talked about the fact that snapshots are stored in S3, but you can't go to your S3 bucket and take a look. You have to go through the Management Console to take a look at those.

🡪You can also use snapshots to copy across to different regions. So we talked about the different characteristics of a snapshot.

🡪We then moved into EBS snapshot features.

🡪We talked about the fact that you can use a snapshot to increase the size of a volume, create a new volume with a larger size.

🡪We talked about how you can move the snapshot to a different region of the world if you want to create it, and how you can share it with the rest of the community and with your colleagues if you wanted to.

🡪We talked about how you can get around some of the lazy loading that happens because you're storing your snapshot on S3, recalling that snapshot is going to be slow. So there's a pre-warmup for Windows, a pre-warmup for Linux that you can do where you're either forcing it to read all of the blocks on the volume and, as a result, S3 is going to send them faster, or if it's an empty volume, you can just write across all the blocks and, as a result, you're going to claim it faster. And the same process exists for Linux as well.