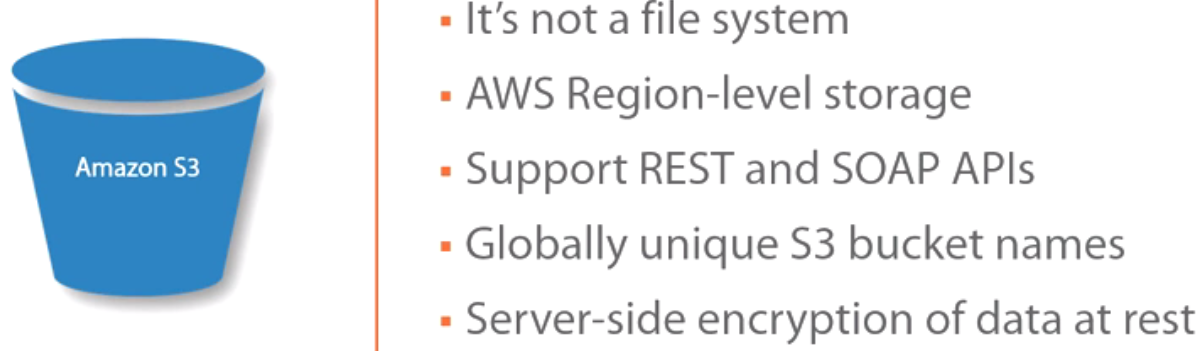
**Working with Amazon S3**

**Module Introduction**

If I were to quiz you right now and ask you, what was the oldest service or what was one of the first services that AWS launched?? S3, by the way, was one of the first services that AWS launched back in 2006.

**Amazon S3 Characteristics**

🡪There's a misconception out there that S3 is a file system. In reality, S3 is not a file system. And if we were to characterize it, it would be **more like a database than a file system**.

It is used to store objects, to store things, but it's not used to operate in a transactional format, to have open files, to worry about locks on files, and all of that.

It is meant to store a completed file and maybe use that file but not open it and use locking mechanisms. So it's not a file system, and it's important for you to recognize that.

🡪When you **create a folder** on S3, it's not really a folder in that it doesn't actually create a folder on a file system because, again, it's not a file system, whereas it creates a representation of a folder. It's an organizational mechanism.

So for you and from your perspective, it looks like a folder. But in actuality, it is not a folder

🡪AWS or **Amazon S3 stores data on an AWS region level,** which means all of your data is going to be within that region. Now they do this for a couple of different regulatory and data sovereignty reasons so that you know that, Okay, I'm within this region, my data isn't going anywhere.

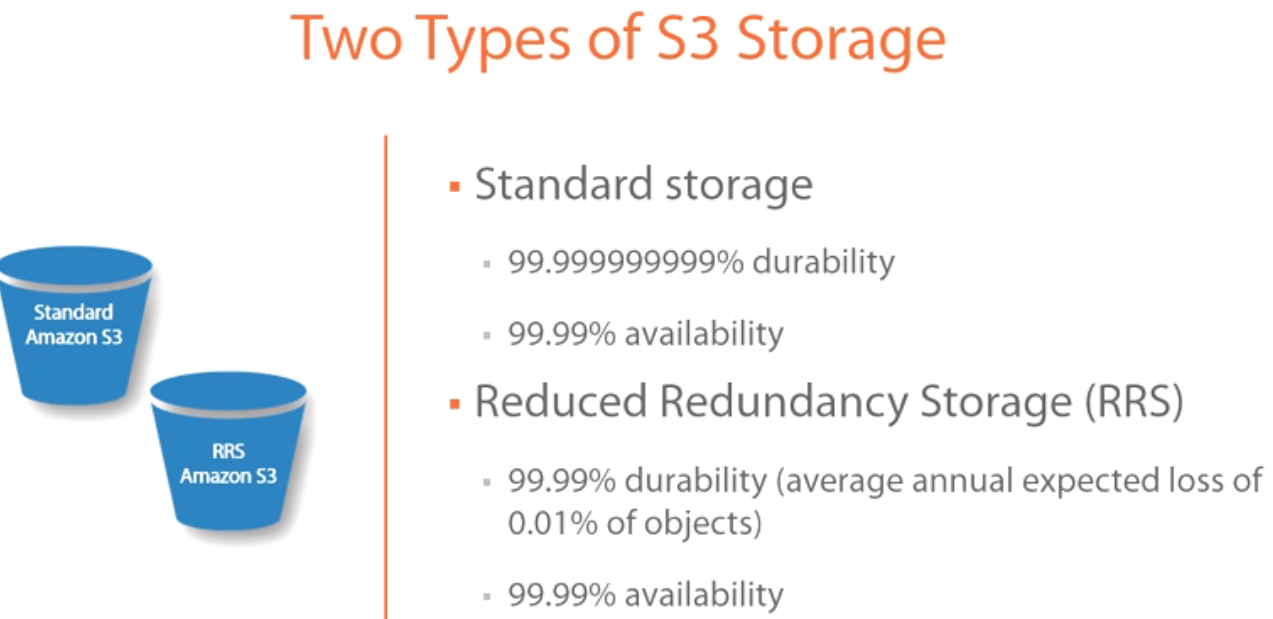
🡪 Replicate your data across a different region, again for protection purposes, maybe because you are required to protect the data and have it distanced from where it originally is by hundreds of kilometers or miles, we'll talk about that.

But when you store data on S3, it's region-level storage.

🡪It also supports REST and SOAP APIs. Now REST APIs are pretty much supported across the board with AWS. But it also supports SOAP APIs for those of you that are developers out there.

🡪You need to have a globally unique S3 bucket name. Now that means that when you create a bucket name, that bucket name has to be unique across all of AWS, not just your account, not just your environment, across all of AWS. And the reason for that is because your bucket is going to be internet accessible, so it has to be globally unique, so you'll have to tweak it a little bit to get to a name that nobody else is using.

🡪You also have server-side encryption of data at rest, which means if you choose, you can encrypt the data that's on your S3. Now you don't have to worry about key management and all of that stuff. AWS takes care of all of that on your behalf, but you also have the ability to encrypt the data that's sitting on your S3 bucket.



There are two types of S3 storage.

**Standard storage:** has the advantage of having an 11 9s durability. In theory, you will never lose data. It'll take millions of years for you to lose data based on this calculation. So it's an 11 9s calculation, ridiculous amount of durability because of the number of times that they replicate it.

From an **availability** perspective, they guarantee 99.99% of the time.

**Reduced redundancy storage (RRS).** Now that has a durability of 99.99%. What that means is it equals an average annual expected loss of 0.01% of objects. That's still very high if you ask me.

And then it has a 99.99% availability just like standard storage. Obviously, RRS comes at a reduced cost as well. 

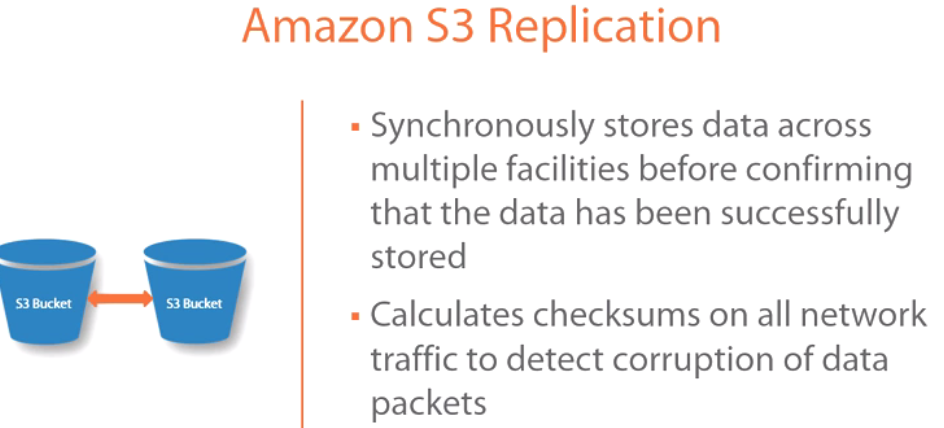
And it's intended to store maybe less important information than what you would store on your standard storage.

**We've been talking about replication, durability, availability. How does Amazon S3 do replication?**

🡪For starters, S3 will store data in multiple facilities and on multiple devices within each facility.

What that translates to is, basically, AWS is going to store, it's going to replicate data across multiple servers in the same availability zone and across availability zones, which is what they're saying here, multiple facilities.

They are not necessarily calling out an availability zone because an availability zone may be consisted of several facilities. **But what they're saying here is they're replicating across multiple physical facilities and on multiple devices within each facility.**

Now from an RRS perspective, because RRS is reduced redundancy storage, Amazon S3 will replicate these objects across fewer facilities. So instead of---let's say the standard is six, maybe this one will be across three facilities. It'll still be spread out across multiple devices within each facility, but it does it a fewer number of times, again, because it's that reduced service. 

-🡪Now the other interesting thing about how replication works, because it's replicating the data across multiple facilities, across multiple devices, Amazon S3 will not flag the data as having been successfully stored until every device in every facility responds back with an acknowledgement that it's successfully written to its device. So imagine if they were doing it across ten facilities and ten devices within each facility, they're able to do that synchronously because of the distance and how they have it configured, but all of these devices, all of these facilities have to respond with a successful store before they can acknowledge that a particular object has now been successfully stored. So a lot of redundancy, a lot of durability there.

🡪Also in order to detect corruption, AWS or Amazon S3 will calculate checksums on all network traffic and on all data packets. So that will help in detecting corruption so that you can be notified quickly.

Let's talk **about Amazon S3 features**. 

-->Amazon has the capability of doing **versioning**. If you enable versioning on your particular bucket, you are then able to preserve, retrieve, and restore every version of every object. So that's very powerful, again, from a durability, from a redundancy perspective on the object level. So versioning is very important, and I'll show you how to enable that in a second.

🡪You also have **cross-region replication (CRR).** We've talked about the fact that when you're storing data within an Amazon S3, it's region-level storage. And they do that, again, because of compliance reasons, data sovereignty, data residency, etc. However, with those same regulations, with those same regulatory compliance needs, sometimes it is required that you store backups of this data in a distant facility, somewhere far away from where the production one is at. For that matter, you can enable cross-region replication or CRR, and I'll show you how to enable that in the GUI as well.

🡪You also have something that I think is really cool, which is **multifactor authentication delete.** Now this is available to you via API. You would have to configure it via API. What that allows you to do is if you wanted to have higher levels of security against your S3 bucket to protect against deleting information, what you can do is require that whoever's going to be deleting objects from S3 needs to have a multifactor authentication device of some sort. It could be a hardware device or a software device, but you need that added level of security. Very cool!

🡪With S3, you can also **give time-limited access to objects**. So you create a URL, for example, and you're sharing this URL. You can basically have a time bomb on this URL where it won't work after three days. So the user will have access to, or whoever you give it to will have access to this particular resource for a set amount of time, after which it expires.

🡪**Audit logs**--you want to be able to tell who's logging in, who's logging out, who's deleting, who's putting information. You can enable audit logs, and I'll show you that in a second as well.

🡪 **Event notifications**--in the event that something happens, you can take action against it within an S3 bucket. So, for example, if there is a PUT or a GET command that gets initiated, so if somebody uploads something to an S3 bucket, then you have the option of triggering an event. So you can control some of these as well.

🡪**Data lifecycle management's** also a very cool thing I think, which allows you to control the amount of data that's on your S3 bucket, and I'll show you where you can do that, and we'll get a little granular as far as how to control it.

But what that does is it essentially is once you upload your data to S3, maybe after a certain period of time, some of this data starts to age, and you want to move it onto cheaper storage. So you can automate the data lifecycle management maybe to offload that to AWS Glacier, which is cheaper storage. Its long-term retention cheaper storage, so you can do that as well, and we'll talk about automating that particular thing as well.

🡪Finally, **permissions**. Permission is a very important topic because it controls who gets access to your objects and to your buckets. Let's bring up another slide that focuses a little more on permissions. So with permissions, you have four levels of permissions and security that you can apply to S3.

1. **IAM policies** ( identity and access management). you have the ability of configuring IAM policies to control access to your S3 buckets. It's user-level security. It's very granular from a security configuration perspective.
2. **Bucket policies**. Bucket policies allow you to, again, get very granular. It's bucket-level security.

You're able to get very granular in terms of who gets access to what. Now you're probably looking at this and you're like, **well, there're two. Which one should I use when? And that's a fantastic question.**

That's going to depend on how you are building the security and configuration, how you want to manage security within your AWS environment.

**Bucket policies are very useful if you wanted to manage S3 security outside of the AWS environment for whatever reason. If you want to manage everything from a centralized location, then I would say always go with IAM policies. It's the biggest umbrella,** it gives you the largest granularity, and it's a single pane of glass, single area where you can go to configure everything.

**But if you just wanted policies just specifically for S3, then bucket policies would be an ideal place.**

🡪You also have **access control lists, or ACLs**. Now, ACLs are **a legacy access control mechanism. It was there before IAM and before bucket policies. So it's still there.** And this is funny because when I think of ACLs and compare that to bucket policies or even IAM policies, for some reason I remember NTFS permissions and share permissions. Do you remember those?

Typically you would give the share everything, and then NTFS you would kind of control it. And the most restrictive permissions apply. Well, this is kind of similar, and when we get into the GUI, you'll see kind of what I mean here.

But with ACLs, you have limited options from a security perspective, and I'll talk about who you can apply it against. And, again, I'll show you that in the GUI.

But, again, the most restrictive permissions apply. So you can always combine ACLs, buckets, and IAM policies, and the most restrictive will always apply.

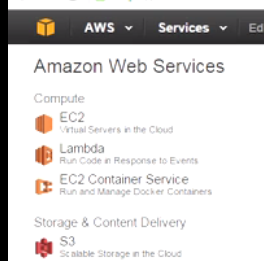
**The ACLs apply to objects and to buckets, and we'll take a look at that**. A

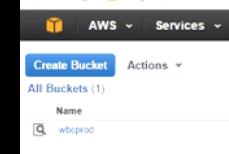
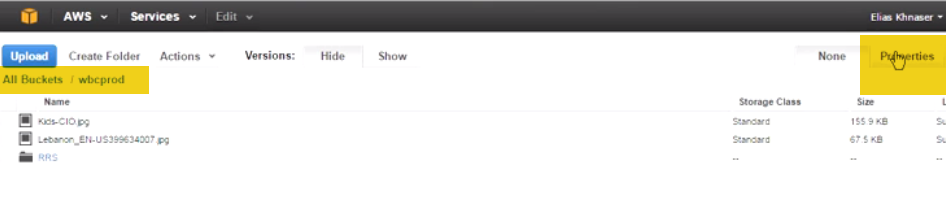
🡪 **query string authentication(**presigned URLs). Presigned URLs can grant temporary access to your Amazon S3 resources. Again, we talked about that a second ago, but this is where you can give a URL, for example, that expires after three days, and after that, your users cannot access this particular URL..

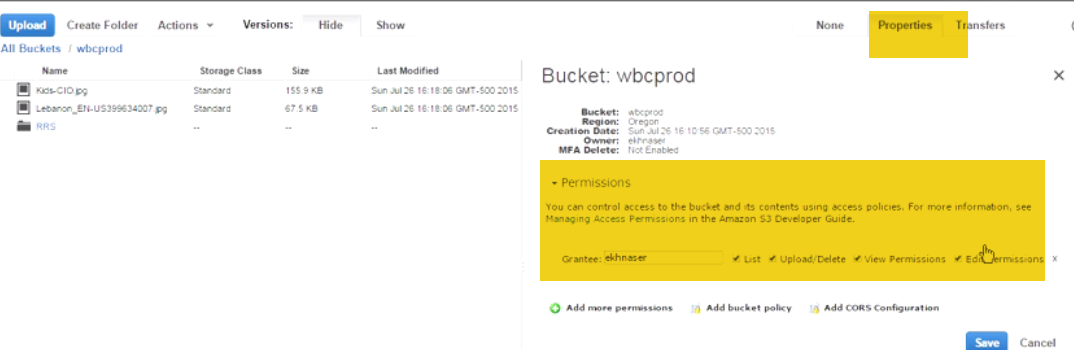
Again, this is access through scripting and APIs, not necessarily through the Management Console.

**Working with Amazon S3 via the AWS Management Console**

🡪From the AWS Management Console🡪under Storage and Content Delivery🡪 S3



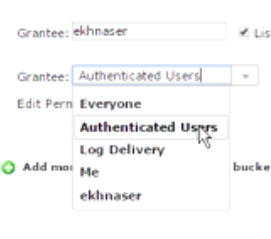
From the previous course, the Fundamentals course, you should be familiar with how to create a bucket. I have a bucket called wbcprod, Wired Brain Coffee prod. I'm going to go ahead and select it, and then we're going to go ahead and click on Properties.  

If we expand the Permissions tab here, the first thing that you're going to be greeted with is ACLs. This is where you would configure ACLs. 

And right now what I have is a particular user, which we're granting ekhnaser list, upload/delete, view permissions, edit permissions, etc.

If you wanted to add another user, what you can do is click on Add more permissions, and then from the drop-down, you have a list of things or a list of grantees that you can give access to.

**Everyone--self-explanatory. It's going to give access to everyone. You don't want to do that. It's the least secure, but you have the ability**.

And this is kind of what I was saying it reminds me of the share permissions and the NTFS permissions. This will show. And then with share permissions, you always remove everyone and put Authenticated Users at least because everyone means everyone in the world, and then you'd control how they access it.

**Authenticated Users** means everyone that has an AWS account or an Amazon account will have access to whatever permissions you give here. Then you

**Log Delivery** if you're configuring logging on the server. It's kind of like a service account. You will have access to it.

**The Me account** is more of a group. Now think of this more in terms of this is outside of IAM, so outside of identity and access management. So if you go and create a user account, under identity and access management, that doesn't make it part of the Me group. **The Me group is simply the user that is connecting or the group that is connecting into this particular user account. So it's very S3 specific.**

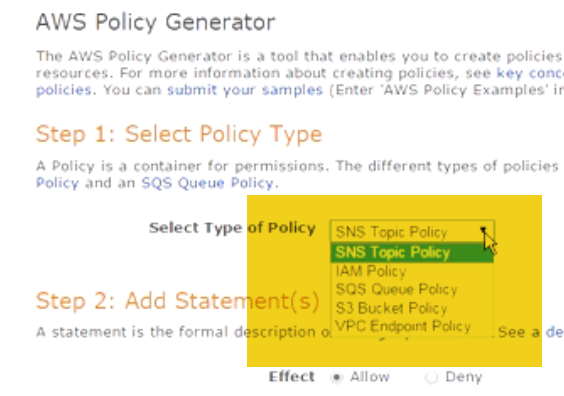
And then you have ekhnaser, which is the user that I am logged in as.

Once you've selected who you want to give access to, again you can control these permissions up here from a checkbox. Very simple. Very easy permissions.Again, in this case, I have access to everything.

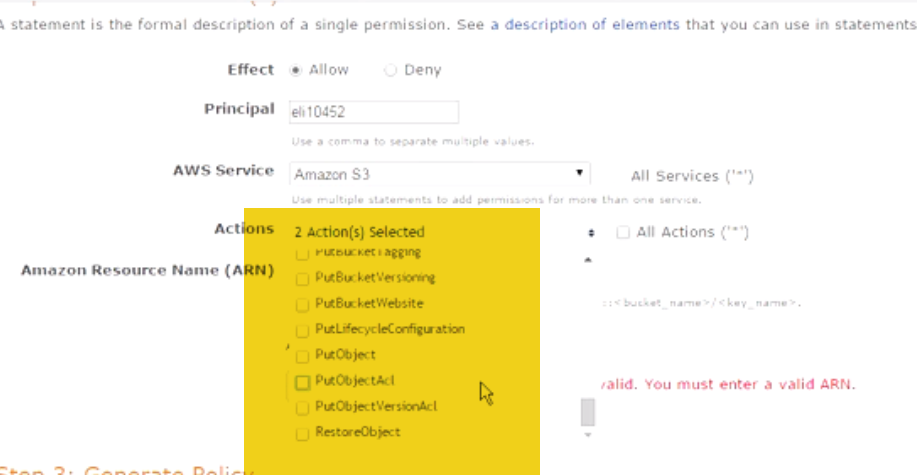
**If you wanted to get granular with the level of permissions** that you're doing by specifying, for example, a bucket policy, then what you want to do is **click on Add bucket policy**.

 If you are very skilled in scripting, then you can go through and write the whole script. If you're like me, you're going to need some help. So what they've provided you with is kind of a policy generator.



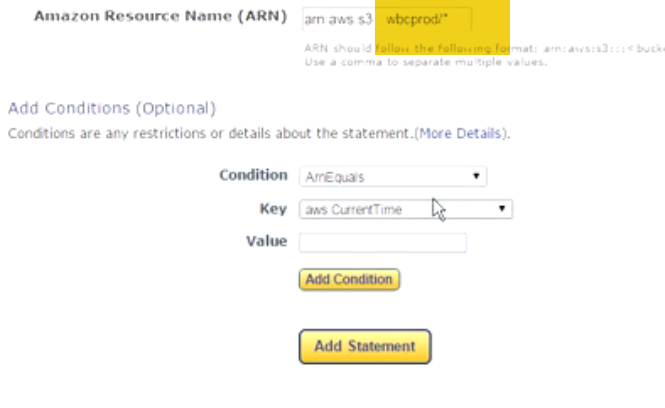
So if you click on it here, this policy generator, that was just a tiny little bit too zoomed out, so we fixed that. And once you get to the AWS policy generator, what you want to do first is select the policy type. Now the policy generator allows you to generate several different types of policies, not just bucket policies. So from the drop-down menu here, you'll find that you have simple notification service policies, IAM policy, which we talked about in the presentation. 

We're not going to cover it here, we'll cover that later when we get into security. It has a dedicated lesson. Simple queuing service policies or S3 policies. This is what we want. You also have VPC, but we're going to go with the S3 bucket policy.

So once I select that, if we come down here, the option or the effect is to either allow or deny. Who is the principal that you want to allow or deny? Who do you want to execute this against? So, for example, you can do this against---I have a user called eli10452, for instance. 

The AWS service that you're trying to apply this to--Amazon S3 in my case. So, once you select the bucket, you can't change this anymore. All services within Amazon S3, this is what you're selecting here. And then the actions. What exactly are you trying to do? So do you want to allow to create objects? Do you want to delete? So you go through this entire list that allows you to customize what they're getting or not getting access to. So it's very granular in those terms. Once you have that assigned, then you come down here to where it says Amazon Resource Name or the ARN.

**The ARN name**: what you have here is ARN, then you have a colon, partition, service, region, account name, etc. So to see an example of how this would work, so, for instance, look up here, you'll find ARN, then you'll find AWS, S3, etc., etc., and it starts to go.

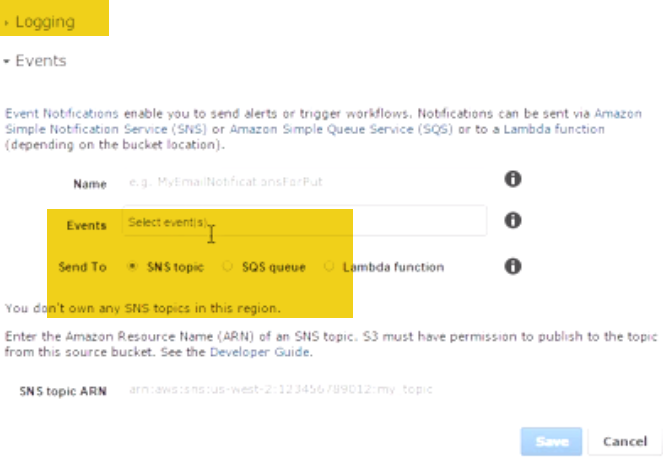
Now you have to put the bucket name, for instance, and so on and so forth. So you get an idea of how to configure this. Now, again, this is a little advanced configuration, advanced administration that you would have to read about and be familiar with scripting in order to generate it. 

But, for instance, if you were to copy, if we were to copy this particular line right here, and take it back into my policy generator, and go ahead and paste it. Now remember that I have to change the bucket name here because this bucket is not really mine. So let's go ahead and select this whole thing and put in wbcprod.

So this is essentially what it would look like. You can add additional conditions if you wanted to. So, for example, these are if statements--if something happens, then take specific actions. And you can specify value. So depending on what you're trying to do with this, it will be able to create this for you, so it automatically generates this for you, and when you're ready, you just basically click on Add and badabing, badaboom, right?

So you can add additional ones. Think of this as a single line if you wanted to. Once you're done, you're going to go ahead and click on Generate Policy. And it will give you this particular thing. So if you go ahead and Select All, Copy, we're going to go ahead and close out of here. Let's close out of here as well. Go back here, and then you can paste it in here and badabing, badaboom. This should work. It might not work against all of the---some of my parameters might be off, but the point is this is how you would go and generate a policy that controls from a granular perspective exactly what you're trying to grant or deny access to.

So in my case, the invalid principal here is eli10452. I just made up that username, which is why we're getting errors here. But this illustrates the example and what I'm trying to convey. So I'm going to go ahead and close out of here. And, yes, we are going to get out of here. We're not going to make any other permission changes on this particular bucket. But what I did want to show you, for example, is if you went into a particular object that's in the bucket, so, for instance, let's select this one, you can click on the Details. Now remember when we talked about the fact that you have standard and reduced redundancy. You can't do that at the bucket level, but you can do that at the object level within the bucket. Now the object level can be a folder, or it can be a particular object because a folder is also an object. But I just wanted to draw the distinction. So in this case, this is one file. So it's a JPEG. You can come in here and say, Well, this is standard or this is reduced redundancy. You also have the option of doing server-side encryption by either selecting None or specifying what you're trying to do against it. Now if you wanted to do this against an entire folder within a bucket, I have a folder in here, once you select that folder, you go to Properties, and, again, under Details, you have the option of either doing standard, reduced redundancy and doing encryption against it. So this gives you, again, an idea of how to navigate these things. On the object level, you also have permissions. So if you have a bucket-level policy that you want to override for one or multiple users maybe, or you just want to customize it a little bit, you can apply an ACL against it very quickly here and then select the permissions that you want to apply. So, again, it gives you some more granularity.



🡪Let's go back into my initial, my bucket, and start going through the list. Now, we talked about the fact that you could do logging. So under logging here, you can enable logging, so enable and configure it so that it starts to log on this particular bucket.

🡪If you wanted to do event notifications, again you would expand Events, and you would configure it.

Again, the event name, what the action is, what do you want it to do when an event is triggered?

Do you want it to send it to simple notification services? Do you want it to send it to the simple queuing services? Do you want to send it to Lambda?

Lambda is a platform compute engine, it's an event-driven compute engine. So instead of having a server that you're sending this to that will then take action against a particular event, what Lambda will do is, the actual platform will initiate that particular action that you want. So it's very powerful stuff.

Again, simple notification, simple queueing services against a particular event that happens is where you can configure this.

🡪We talked about versioning. So enabling versioning is really easy. Basically as long as you have the required permissions under versioning, you're going to come in here, you're going to click on Enable Versioning. 

you can now go into cross-region replication. We talked about the fact that sometimes you want to be able to replicate the data within your S3 to another region, whether to protect it, for regulatory reasons, etc. What you can do here is expand this, and you either do not want to do this or, Yes, I want to do this. If you select that you want to do this, are you doing it against this particular bucket? Or are you doing it against the prefix in this bucket? So you can get very granular again as far as how you want to control it. Once you select it from the drop-down, you then select the region that is the destination region. So AWS gives you that flexibility. This is important. Why? Can anyone tell me why this is important? Because in some countries, even in ours, in the United States or in Europe or wherever you're in, you want to make sure that the region that you're replicating to is within the data sovereignty rules and regulatory requirements. So, for example, if you're in Europe, you might not want to replicate to the United States. That might breach data sovereignty and residency requirements. So you want to select a region maybe in Europe. And if you're in the United States, you probably want to select a region within the United States. And here we start getting into states and all of that stuff. But you want to be able to control which regions it's going to from a regulatory perspective, so this is where you can do that type of stuff. You can also configure the destination bucket in that region where you're going to. For the purposes of our example, we are not going to do this. I just wanted to show you where it was. Let's go back up here to lifecycle. We talked about the fact that you can control lifecycle management. Lifecycle management allows you to manipulate the data once it hits S3. So, for instance, you can click on Add rule, and what you're doing here is you're applying this rule to the whole bucket, or you're applying the rule to a prefix. So if I'm going to apply it to the entire bucket, we're going to click on Next, now the action is, What would you like me to do? Action on Current Version--Do Nothing. From the drop-down menu, you have the option of archiving it only. If you select Archive only, what this will do is, When do you want to archive to Glacier? Remember, Glacier is the low-cost, long-term storage that AWS offers. So, for instance, you can say, I want to archive after ten days of upload. So as soon as I upload an object, ten days later I want you to move that particular object to Glacier. Action to Previous Versions, again, previous version of the document because we have versioning enabled. So once there are several versions of the same object, what do you want to do with previous versions? So maybe you don't want to do anything with current versions, but maybe previous versions you want to archive them, so it gives you that level of flexibility again so you don't have several different versions of this, and you have them for long-term retention if you choose to do it. Now from the drop-down, you also have the ability to also select Expire Only. Now what does Expire mean? This means that after x amount of days, the current version will expire, and a new version will be generated. Again, this is all a function of versioning to keep files updated and kind of a backup for this particular file. Think of this as a word document that you have opened. After several minutes, it will automatically save. This is kind of the same thing where after a day or two or three or a week, it expires the current version and versions a new one just so you always have updated ones. You can also couple the expire of the current version with maybe a delete of the previous version. So you have flexibility again there to control it. So maybe instead of archive here, you want to permanently delete. So you want to expire the current version, which forces a new current version, and the previous version you want to permanently delete. Or you want to go to archiving of that previous version. So you see, very, very flexible configurations that you can control here. You also have the option of doing Archive and Then Expire, which means if you select this, you are archiving the object after x amount of days and then you are expiring it if you want to in order to generate a new version. And then once you have a previous version, then you can take control against that as well, which means then maybe you can permanently delete or archive and then permanently delete, which is really cool because maybe you move it to archive first for 100 days. And then maybe after 100 days you permanently delete this. So, what you're seeing here with lifecycle---data lifecycle management is the amount of granularity, the amount of control that you have to lifecycle manage your data. If your data is going to live forever, then maybe just archive and keep everything. But if you're able to get granular about when---maybe from a regulatory requirement's perspective, you're not required to keep data after seven years. Well, in that case, you can set your policies here. You can configure your lifecycle management so that it starts to delete things that are over seven years old. So, again, you can become very granular here with the amount of data lifecycle management that you can control. I'm going to go ahead and click on Cancel, and that is lifecycle management. We talked about cross-region. Tags is, again, the ability to tag things, so I just wanted to make sure that you are aware of those. Requester pays is kind of cool because if you have a particular object, for example, an S3, and you're making that object available to the world, maybe it's an ISO image, maybe it's a software that you want to charge for, whatever the case might be, you are charged for data transfer out of AWS. So if you do requester pays, then all of the charges associated with requesting that file or requesting that data out of S3 is billed against the requester instead of the owner of the bucket. So it gives you a lot of flexibility there and, again, controlling costs and the level of things that you can do with S3. Very powerful stuff. I hope you're really enjoying this. S3 is a really cool environment that allows you to do a lot of things. With that said, let's go ahead and switch back to our presentation and recap what we've learned.

**Module Summary**

🡪Amazon S3 characteristics. We talked about the fact that it's not a file system. If you were to think of it, it's more of a database.

When you create a folder on S3, it's not really a folder. It doesn't actually create a folder on a file system. It is a representation of a folder for organizational purposes. However, from you as the user's perspective, you see this as a folder, but it's not really a folder.

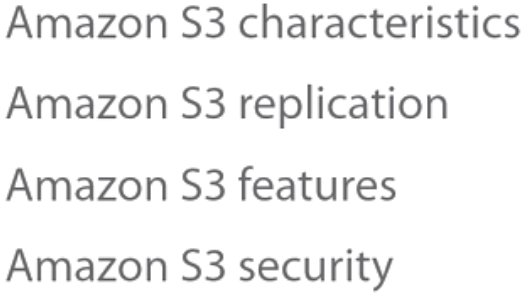
We talked about its durability, its availability, SLAs, the 11 9s. We talked about the fact that you've got two standard and reduced.

Standard will copy it across multiple facilities, multiple devices. Reduced will copy it across fewer facilities, fewer devices.

🡪We then moved on to talk about S3 replication, that, again, all of your data in S3 is stored within a region. However, you are able to do cross-region replication if you wanted to store this hundreds of kilometers way or hundreds of miles away as well.

🡪 We also talked about the fact that from a replication perspective, because it's being replicated across multiple facilities, multiple devices, it will not flag each object as having been successfully stored on S3 until all of the devices, all of the facilities respond with a successful write, which means it's going to synchronously replicate across all of these devices, all of these facilities.

🡪We talked about the S3 features. We talked about versioning, lifecycle management. We talked about event notifications, permissions. We then went into detail on security. We talked about the fact that you've got four levels of security--IAM, bucket policies, ACLs, query. We talked about all of those, and I went through and showed you where you can apply all of those within the Management Console.

****