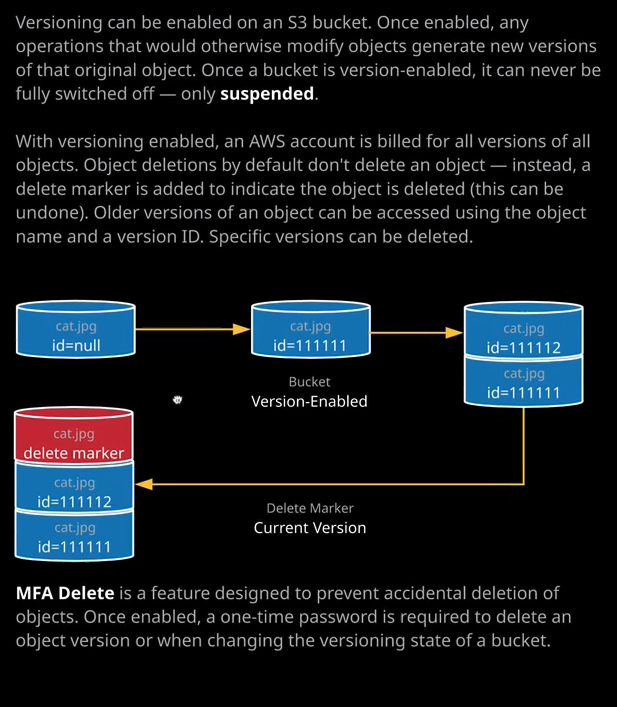
Welcome back and in this lesson I want to cover a really important feature of S3 especially for the exam and that feature is called **versioning**.



**Now, by default, versioning is turned off on any new buckets that you create in S3.** What this means is that inside an S3 bucket by default, you only ever have a single version of a specific object. So in this bucket I've got larry.jpg and oops.jpeg and if I were to upload any additional objects with the same name into this bucket it would overwrite these existing objects. If I were to delete these objects, it would delete these objects permanently. **So without versioning enabled, you only have a single object version and those objects can be permanently deleted. Now, versioning is enabled on a per bucket basis.** To enable it, to go to properties I'll click on versioning and I'll enable versioning and click Save. The important thing to understand about versioning at this stage is that you can never disable versioning. Once it's enabled on a bucket, you can only ever suspend versioning you can't fully disable it. When I enable versioning the bucket every object in that bucket is given a version ID. So in this example, cat.jpeg has version ID 111111 with versioning enabled, I can then upload a new version at the object. So I could upload a new version of cat.jpeg. What would happen in that case would be this new version of the object will be given a new ID in this case 111112. **Now with versioning, individual versions of objects are all accessible,** so there's nothing to stop me accessing cat.jpg the original version or cat.jpg the new version. From the console, you'd always see the latest version, but you can see individual versions. So let me demonstrate this. **So by default, versioning is hidden in the console**. So I've got larry.jpeg and oops.jpeg. Watch what happens if I go ahead and upload and new version of one of these files. So let's say I upload larry.jpg again. So I go through all of the defaults and click on upload without showing this by default, you see no change in this bucket but if I click on show versions, you'll see that we've got two different versions of larry.jpg. Now know that the version ID in reality is a lot more complicated than the numeric one that I've got in the diagram. This is just for simple illustration, but essentially every unique version of every object has its own unique version ID. Now, once I show this in the console if I'm doing this manually from the command line, I am able to see individual versions. I can reference individual versions of these objects and I'm actually charged from an S3 perspective for every version of every object that's in the bucket. So there are consequences if you do enable versioning you can't disable that you can only suspend it and when you enable it, every time you upload a new version of an object, you are still billed for those old versions. What's perhaps more critical is if now versioning is enabled and I decide to delete this larry.jpg object so I go to delete, delete that object. You may be excused for thinking that the object is permanently deleted, but that's not the case. If I go to show objects now instead of a deleted larry.jpg object, I actually have three versions. **When you delete an object in a versioned bucket what you're actually doing is adding this delete marker. It's essentially a way of indicating to S3 that you've deleted this object, and so essentially it hides it but you are still being billed for these previous versions. What you can do is remove this delete marker so I could select it and select delete. So I'll go ahead and delete this delete marker and then, if I hide versions, I've got my object back. I've essentially undeleted my object by deleting the delete marker.** So these are important things that you need to understand for the exam. **So when versioning is enabled, every object gets a unique version ID. If you do just delete an object, all you're doing is adding a delete marker. They are deletable, as is any individual version of an object. Now via the console or the command line you are able to select all versions of a particular object and delete those and what that will do is permanently delete that object so now we have actually deleted the larry.jpg object.** We're not able to get that back because we've deleted every single version of the object from the bucket**. So the default behavior of the console is to hide versioning so you can upload objects, upload new versions of objects, delete objects. Remember that doesn't delete it. It just adds a delete marker but you can now make the console behave in the same way as the command line by showing versions, and that gives you access to be able to permanently delete objects.**

The last thing I want to talk about before I finish up this lesson is a feature called **MFA Delete.** Now you don't need to know about this in detail for the associate level exam. **Essentially, MFA Delete allows you to enforce that you need to have an MFA token associated with your IAM user, and you need to supply a one time generated code from that device in order to interact so to delete objects inside an S3 bucket.** You don't need to know any more information than that apart from that the feature exists. So by enabling MFA Delete, you ensure that the only way that you can really object from an S3 bucket is by supplying this one time generated code and I'll make sure that includes some additional detail of this in the lesson description in case you want to do some background reading but you don't need to know anything more about it than just the feature existing for the Solutions Architect Associate exam. I just wanted to mention it because it's an important feature to know how it works. With that being said, though, that is everything I wanted to cover in this lesson. I just wanted to talk about versioning how you can enable it. **You can't disable it. You can only suspend it. How every object gets a version ID. How deleting an object just creates a delete marker and you can access and delete individual versions of an object.**