

Welcome back and in this lesson I want to cover something a little bit different. I want to talk about how S3 buckets can be configured to host websites. Now, **by default, S3 buckets are accessed using APIs and that's done using the console, the CLI, or directly using those API endpoints.** So to access a bucket in the U.S. East 1 region would be using the S3 public endpoint that's located in U.S. East or North Virginia. So that's one of these endpoint addresses but with S3 you can also enable an S3 bucket to be accessed using normal HTTP and a web browser. Now you might you is S3 using this architecture when you want to use it to host a normal HTML, style sheets, or Java script files or, alternatively, media files, such as audio, movies, or images. S3 can be used to host frontend code for serverless applications, so you might be running HTML with built in JavaScript and use that to host the entry point for a serverless application or you might use S3 as really economical object storage. If you've got a website, maybe it stores images, maybe it's a blog site rather than storing the media on that web server, you could store on S3 and direct the web server to point to S3 for those media files. So it makes a really effective location to offload static content to.   
**CloudFront can also be paired with S3. So you give your customers that improved speed and efficiency of content delivery, or make it easier for global users to access your content in a really efficient way.** So let's have a look at how this works in practice. So if I go to S3 and then I pick the ac-catpics1337 bucket that I've been using in this topic of the course. Well, I can go to properties. I could go to static website hosting and I can select to use this bucket to host a website. Now when I do that, I'll need to specify a number of things. I'll need to give it a document to use for the index of the bucket. So if I browse to a particular bucket and specify an object name, well, it'll use that object name but if I just browse to the bucket endpoint so the bucket URL itself by default, it'll load up this index document and so I need to specify that. Additionally, I'll need to specify an error document and that's the document that's used if I'm trying to browse to an unknown file or a file that someway got corrupted or if there's any other errors that occur accessing objects then it load up this error document. So before I do that, I need to upload the objects to this bucket. So first, I'm going to go ahead and delete all of these objects that were uploaded in previous lessons. I'm just going to select to delete them and then I follow the same process for this test folder. So I'll just remove that now inside the folder for this lesson in the course GitHub repository I put some sample files I'm going to use to upload to this S3 bucket. So I'm going to start by doing that, I'm going to upload some objects to this bucket so I'll click on upload and then add some files. So in the course GitHub repository it's in lesson files. It's in the 05 storage content delivery folder S3 architecture, static website hosting, and then I'm going to upload all of these four objects. So it's two image files and two HTML files. I'll accept all the rest of the defaults and click on upload. Okay, so once that's done I'll click on Properties. Go back to static website hosting, enable it for real this time, and in the index document box, I'll type index.html and then in the error document box I'll type error.html. Now that I've got both of these set, I'll just copy this endpoint URL into my clipboard. This is the URL that I'll use to access this S3 bucket. I'll just make sure I've got it in my clipboard and then I'll hit save. So that's the bucket enabled for static website hosting. So next I'm going to go ahead and open that URL in a new tab. When I do that, I get a 403 forbidden status code.

**Now, the reason for that is just because I've enabled this S3 bucket to be a static website. By default, all objects inside an S3 bucket are private only and because of that, anyone who's accessing this bucket as an anonymous user when you're accessing an S3 static website hosted bucket you're coming in as an unauthenticated or an anonymous user and so, by default, this anonymous or unauthenticated user has no access to any of the objects inside this S3 bucket.** Now, to fix this 403 error, I need to do two things, and both of those are inside this permissions tab. The first is I need to remove the block public access settings because in this case, I intentionally want to make this S3 bucket publicly accessible. So I'm going to click on edit and switch off all of these block all public access settings. So now that that's unchecked I'll go ahead and hit save. Now because this is a serious change that I'm making. I'll need to confirm that I'm okay doing this so I'll type confirm and then click Confirm to explicitly approved this change.   
**The second thing I'll need to do is adjust the permissions on the objects in this bucket so it'll allow them to be accessible or read by an unauthenticated or an anonymous user and I could do that in one of two ways.** I could go back to the overview and select each of these objects individually, click on actions, and then go to make public. Now, what that would do is set an ACL and those individual objects, remember, I covered ACLs in a previous lesson, and it would use that ACL to make them publicly accessible. What I want to do, though, is to click on permissions, select bucket policy and create a bucket policy for this bucket that applies those permissions to all objects in the bucket. Now, I've got one that I've prepared earlier, and I'll make sure this is in the lesson description. Essentially, this allows any principle so anyone accessing this S3 bucket to perform this particular action, which is s3:getobject and it allows any principle to perform this action on any object inside this ac-catpics bucket and I need to correct the name slightly. I need to change it to 1337. So ac-catpcics1337 and if you're doing this in your own environment, just make sure you use your own bucket name rather than the one that I've got set. Once after a little hit save and I'll be informed that this bucket now has public access. So AWS are really serious about warning you whenever you're making any potential mistakes but in this case, it's fine. We want the bucket to be publicly accessible. Now that this bucket policy is in place, it means that any identities accessed in this bucket will fall under this policy and that means that I should now be able to access objects that are in this bucket. So if I hit refresh it will load the index document, which is Larry the cat. So this is the index.html document, and it loads an image that's also stored as an object in this S3 bucket.

Now, this is the index document. If I try to load a nonexistent object, So if I put forward slash and type asdf.jpg this is an object which doesn't exist in this bucket. If I do that, it's going to load the error document so the error.html and this error.html file contains this text and it links to an object inside the bucket called oops.jpeg and if you recall, if I go back to the S3 bucket this is one of the Jpeg objects I uploaded along with these error and index HTML files. So this is a simple example of creating an S3 bucket and setting static web hosting. Now, there are other really cool features that you can do with S3 static web hosting.

**You could add a custom domain.** The way that you can do this is you can create a bucket and you can name the bucket the actual DNS name that you want to use. So I've got a personal blog and it's called cantrill.io So I could call my bucket cantrill.io and create that bucket. I've already got a bucket called cantrill.io so it won't allow me to create this, but if I did want to use this bucket and use a custom DNS name then I could create a bucket with the custom name that I wanted to use and then go into Route 53 create a record in my domain that points at that bucket and that's how I can provide a friendly name that my customers or clients can use to access my bucket.   
**You could also define redirects**. So if I go back to static web hosting I've got the option of redirecting requests through to another bucket and I can specify the options there, but also I can specify a full set of redirection rules. Now, this is well beyond the scope of what you'll need for the exam but I do want you to be aware of the functionality in case you're using S3 in production.

Now there is one last feature that I want you to be aware of and that's known as CORS or cross origin resource sharing. **Now CORS is a security measure that you can build into a web application to allow an application or script that's running inside one bucket to redirect to another bucket. Now you might use this if we've got an application running out of a bucket is called www.linuxacademy.com, and it's loading scripts or images from another bucket that's called secretcatimages.com. The place that you define this is inside the permissions tab, and then you go to the cause configuration.** Now it's not something that you need to be aware of for the Solutions Architect Associate exam. You need to be aware that CORS exists because you might face a question where you're attempting to use this architecture and you getting security errors and if that's the case, then you need to apply a CORS configuration. Other AWS exams, so other associate exams, the professional exams, and even the speciality exams do need you to be aware of how to use CORS end to end. So I will make sure that include a link in the lesson description, which will give you extra information if this is something that you're interested in but I want to keep this focused for what's important for the Solutions Architect Associate exam. So you need to know that CORS exists. You need to know when it's used so if you want to allow resources to be pulled from different buckets or different locations. They need specify a CORS configuration.

Now one final point before I finish up this lesson, **you are able to configure S3 server access logins. So if you do use an S3 bucket as a website you can configure a full set of Apache web server style logs** will be covering this in much more detail later in the course when I talk about monitoring logging, but it did just want to mention it. At this point, you can configure server access logging. You need to enable it on a bucket and then specify a target bucket and optionally a target prefix and when you do that, any logs that are generated by accesses to this bucket will be stored in log files in the target bucket using the target prefix and again.