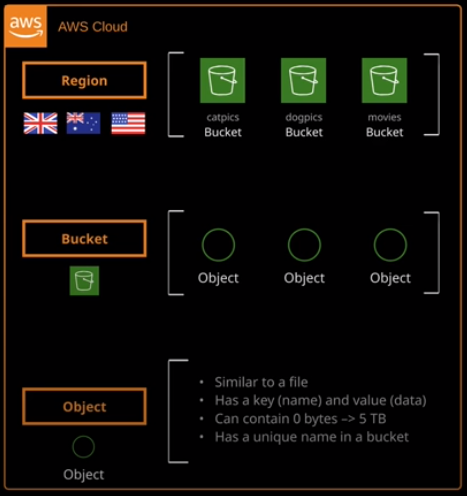
One of the oldest products available within AWS and also one of the most useful and common services to use and that is simple storage service, which is also known as **S3**.

Now S3 is an object storage system, and at a high level, you can think of this like a file server, which you might have used in a corporate environment only one which can deliver extreme levels of performance, do so globally, and cope with a near unlimited amount of data. S3 is great for storing text files, photos, audio files, movies, large binaries, logs, and even huge scale data sets covering mapping, weather analysis, and even large-scale genomics.

It's essentially perfect for anything where the primary requirement is to store and provide access to content using web style protocols, it isn't a file server, so you can't interact with it in the way you can with a file server but you can access objects using a web browser from the command line or via any applications specifically designed for S3.

S3 is a global service. You'll be able to select the region when you create an S3 bucket. The basic entity inside S3 is the bucket, and it's a good physical comparison to make, because a bucket in S3 is a container which can store things just like its physical counterpart. Now buckets have a name and that name needs to be unique.





I've got three example buckets cat pics, dog pics, and movies. The name needs to be unique not just in the region that you're in, not even in your account, but globally across all regions in all AWS accounts and when you create a bucket you'll pick a region for that bucket to be located in. The first step of creating bucket to pick a bucket name. So if I do pick something obvious like cat pics an attempt to create this bucket we'll see that the bucket name already exist just to reiterate, it needs to be globally unique but I do want cat pics to be in the name because that's what it's going to store. So I'm going to try cat pics-ac which are my initials. I'm going to select U.S. East, North Virginia because that's the region name that I want this bucket is going to be in and attempt to see if this name is taken. So click on next. In this case, the name is available.

S3 is a service where you can make these objects available to the public internet or indeed, any identity that you choose. By default, though as a protective mechanism the bucket is blocked from all public access. So this is a feature that was implemented relatively recently by AWS and if you do enable this setting then it is impossible to allow public access to this bucket and it is the default so it's something to keep in mind. We'll be asked to review the settings. They're all good and we'll click on Create Bucket. So at this stage we created the bucket and remember a bucket is simply a container in S3. It doesn't itself cost any money. It's just something that contains other things.

Now I could also create another bucket in this account. So let's do that. I'll go ahead and create bucket. This time I'm going to be inclusive. So I'll pick dog pics-ac and click on next. Assuming nobody's taking that name, which they haven't I'll skip through to the end of this and create bucket now when you store anything inside S3, it's replicated across availability zones in that region. In this particular example I've created the catpics and dogpics buckets in U.S. East, North Virginia. So any data that's stored inside these buckets will be replicated across the availability zones in that region.

So if you need a resilient way to store data inside AWS, then S3 should be a candidate to do so. Now buckets are designed to store things and they store objects. So a single bucket is capable of storing multiple objects. So let's go in and take a look. Now, I'm going to introduce a concept at this stage which might be slightly too technically deep for the associate certification but I think it's an advantage to know this straight away.

A bucket is actually a flat structure. It can't contain folders. Whenever you do see folders or folders within folders or even deeper levels of nested folders, they're not really folders. They're just files that S3 shows as a folder.

Buckets can store an unlimited number of objects and can also store an unlimited amount of data. The name of objects need to be unique in terms of where they're located, so I could have an object called cat.jpg directly inside the bucket and then also one inside the Winky folder, and that would be fine but I couldn't have two objects, both called cat.jpeg, at the top level of the bucket so it needs to be unique.

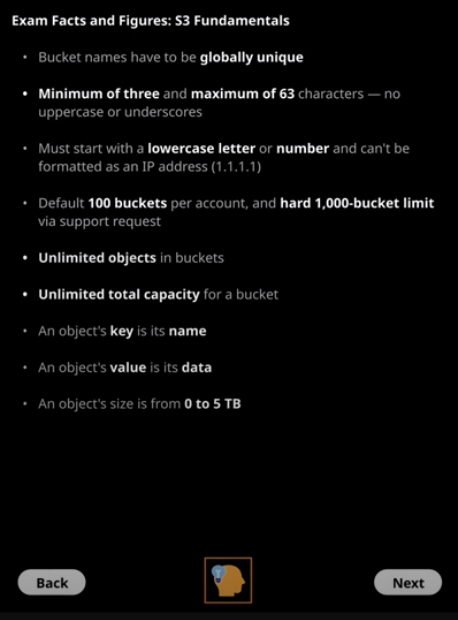
Now an object in many ways is just like a file that you would store on a file server. It could be empty. It could be a short text file, the whole business report, a large movie, or even a big data sized data set.

Objects are actually slightly different than a file in that they're a name value or a key value pair. The object key is its name. The value is the data of the object, so let's upload some objects and just see how they look inside the console. So let's click on upload and then add files and I'm going to select a number of objects to upload to this bucket. For this example, it's going to be a picture of each of my four cats. So these are the four objects I have uploaded the pictures of my cats they look like files and for the most part, they behave like files. If I click on one of these objects and then click on open, I'm able to see that is just an image of one of my cats, and I can open it in a web browser using HTTPS. Now, for the most part, objects do behave like files they do add additional features above and beyond what a file is able to do. One example of that is if I click on the properties tab for this particular object, I'm able to select metadata,

So, an object allows you to attach additional metadata. It might be S3 managed metadata or it could be custom metadata that you, as the user of S3 can add. So if you're using an S3 bucket to store data from a third party application, then you might want to use it to store a file ID, a customer ID, or some other customer related metadata that attaches itself to this object. So it's a way in, which S3 can integrate with any third party applications and don't worry. So this is how an object is actually a key and a value well, the value part which represents the data of that object. So, in this particular case, it's the binary data for the image Penny.jpeg. The data could be anywhere in size from zero bytes to five terabytes. So that's 5,000 gigabytes or five million megabytes. So pretty big. It's feature rich, and you're able to select from different types of storage. You're able to select encryption, various different types of replication, etc.

You've got buckets inside an account those buckets are containers. They contain objects, objects are keys, and values. The key is the name of the object, so penny.jpeg and then the value is the data that the object stores. In this case the image data, which is a picture of my cat Penny.

Now, when objects are in a bucket they could be accessed over HTTPS. They can even be made public or presented as a website. One of the most important elements to understand early on with S3 is when and where you would use it as a product. I mentioned at the start about S3 replicating data across availability zones in a region, and that's a huge part of the value that S3 adds. If you need data to be **safe** **and secure then S3 is perfect, it's not vulnerable to a single AZ failure impacting it. It can tolerate large amounts of failure within a region and still work correctly. It also scales. You don't need to be concerned about the performance because S3 largely handles any performance or scaling.** There are exceptions, but we'll cover those later in the course. **S3 is also great if you need to share data inside AWS. So if you have multiple servers which need to read and write, objects may be shared media for a web application or maybe patient's scans or other medical imaging for a medical application then again, S3 is perfect. If you need to offer this data to the public internet at scale then S3 is brilliant**. Now what it's not great for is that it can't be easily mounted into a server as a network drive letter or mount point. S3 buckets are also not something that you can attach to a server as a disk. That's something that's called block storage and **S3 is an object storage.** They're two very different things, and I'll be covering what block storage is in the EC2 topic of the course.



Now you've got a default of 100 buckets per account and a hard limit of 1,000 buckets. This is important because in the exam, if you get a question, for example, where you might be looking to design an application that stores data for, say, 10,000 medical patients, you might be faced with one of the answers to that question being that you would create a bucket for each one of those patients. Now, you know that there is a hard bucket limit per account. You know that you can't use S3 in that way. So these facts and figures are things that will influence which answer you pick in the exam so you can't create more than 1,000 buckets but there's nothing to stop you using a single bucket and then creating thousands, tens of thousands, or even millions of these folders and of course, now you know that they're not really folders, and that will really help you answer questions later on as we proceed throughout the course.

The bucket can have unlimited total capacity, so a single object can be anywhere from 0 to 5 terabytes but because you can store unlimited objects in a bucket you can have unlimited capacity inside a bucket. Now, you might hear me refer to an object name or an object key. They're basically the same thing. They both represent the name of the object. If I talk about an object value or an object data that represents everything else. So in the examples of these cat pictures, it represents the binary data as well as any associated metadata or other data around the object.