Background:

We will practice the concepts we learned in the AWS S3 (Object Storage) Lecture

\*Provide the required screenshots with your account ID in them to get full credit for this exercise.

\*Follow the steps and provide a screenshot with your account ID for each step.

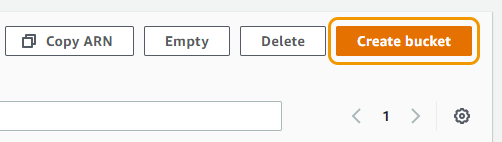
\*Please empty and delete your S3 bucket once you are done with the exercise

**Problem #1: Creating an S3 Bucket**

#### Every object in Amazon S3 is stored in a bucket. Before you can store data in Amazon S3 you must create a bucket.

You are not charged for creating a bucket; you are only charged for storing objects in the bucket and for transferring objects in and out of the bucket.

1. To find and open S3 you can use the search bar or click on the **Services** link in the upper left-hand corner of the screen to bring up the services menu. Under the "Storage" heading select **S3** or open the Amazon S3 console [here](https://console.aws.amazon.com/s3)
2. Click the **Create Bucket** button. You will be taken to the "Create bucket" page to begin setting up your bucket.



1. Enter a name in the "Bucket name" field. The bucket name you choose must be unique across all existing bucket names in Amazon S3. One way can make your bucket name unique is by prefixing your bucket name with your initials and your organization's name. e.g. [your initials]-[your org]-s3-lab

Bucket names must comply with the following requirements:

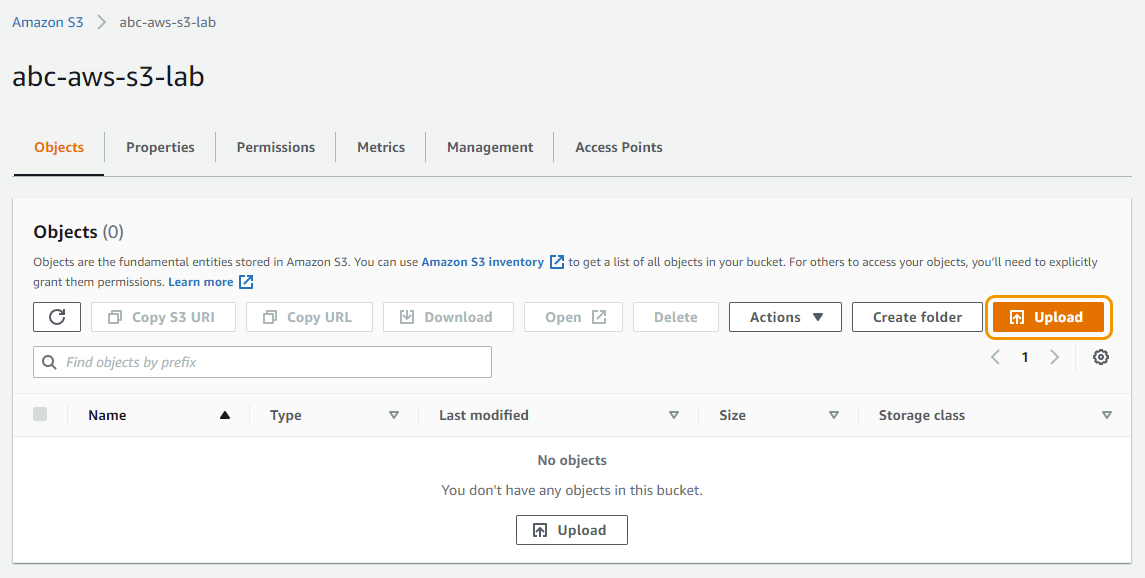
* Can contain lowercase letters, numbers, periods (.) and dashes (-) only (No UPPERCASE letters!)
* Must start with a number or letter
* Must be between 3 and 255 characters long
* Must not be formatted as an IP address (e.g., 265.255.5.4)

1. In the **Region** drop-down list select the same region where you setup your web host with CloudFormation.
2. The next section is "Block Public Access settings for this bucket". We will be working with a private bucket so leave **Block all public access** checked. You could set your bucket up for public access giving users and applications the ability to access the objects within the bucket via a unique DNS address. We will not be doing that for this lab. In one later section, we will go through the process of setting up access to your objects without making the bucket public.
3. You can leave the rest of the settings as default for now, we will be enabling versioning on our bucket later in the lab. Now click on the **Create bucket** button.

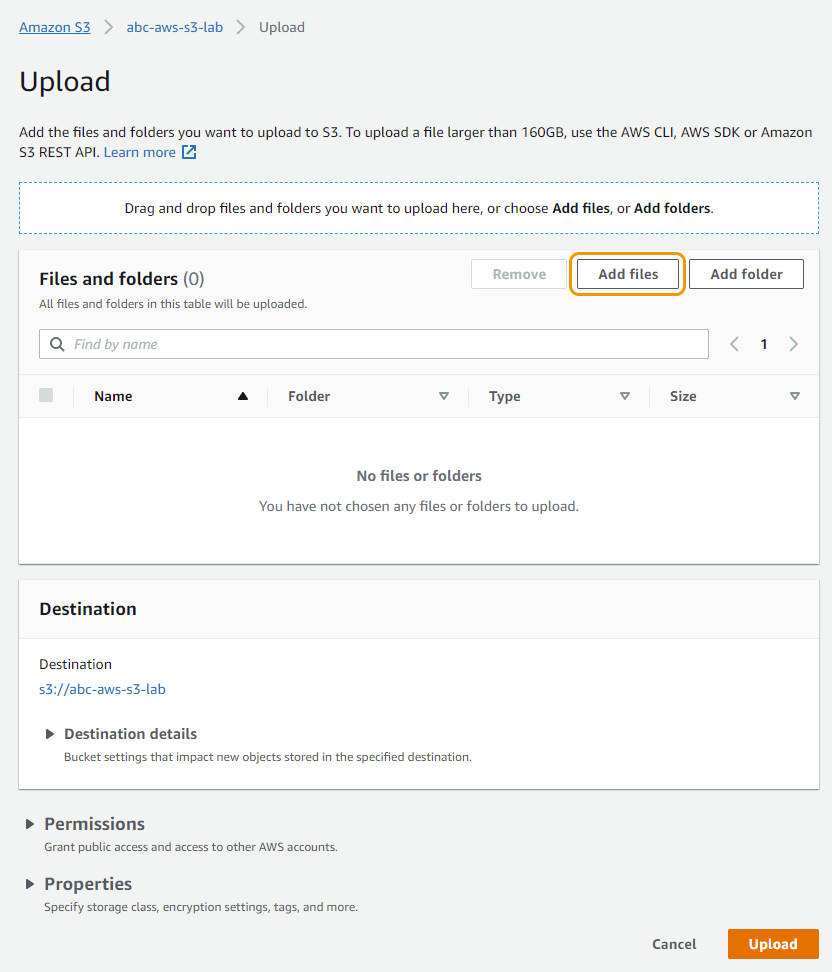
**Problem #2: Adding Objects to your S3 Bucket**

An object can be any kind of file: a text file, a photo, a video, etc. When you add a file to Amazon S3, you have the option of including metadata with the file and setting permissions to control access to the file.

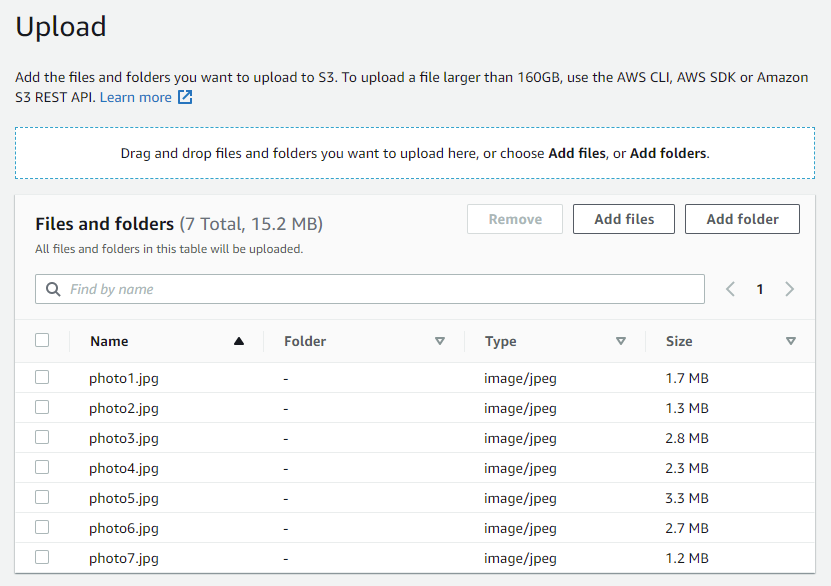
1. You are going to upload 7 photos into your bucket. Download this zip file and extract the photos onto your local hard drive: [photos.zip](https://static.us-east-1.prod.workshops.aws/public/1afe8683-b03d-41b8-b285-726518136bab/static/common/s3_general_lab/photos.zip)
2. In your zip file confirm you have seven files named "photo1.jpg" through "photo7.jpg" and a directory named "V2" containing a file named "photo1.jpg".
3. In your new bucket's overview page click **Upload** under the "Objects" tab.



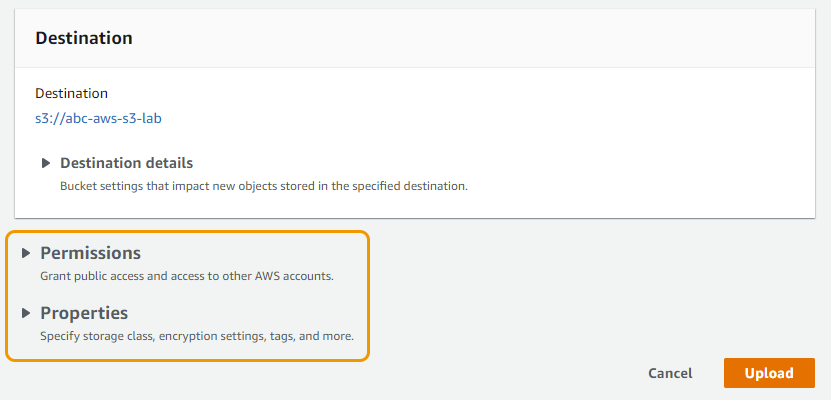
1. Then click on the **Add Files** button to select your files for upload. Upload the files "photo1.jpg" through "photo7.jpg" from the root of the "photos" folder, ignoring "photo1.jpg" in the "V2" folder as we will use it in a later section of this lab.



1. After you have selected the image files, the "Upload" dialogue then should show the files you've selected to upload.

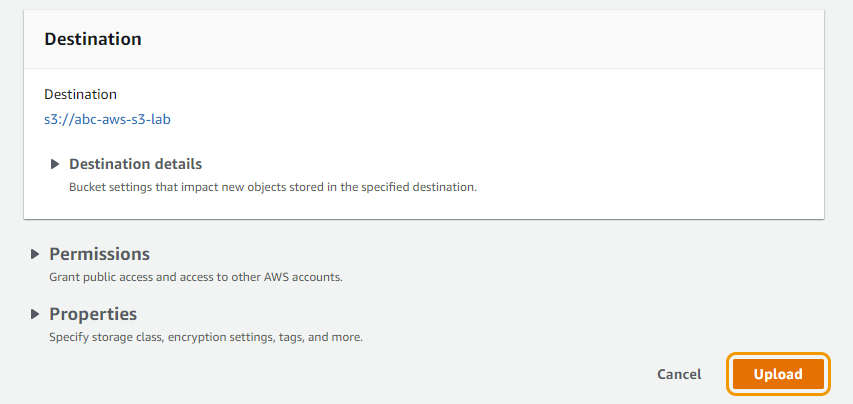


At the bottom of the page you can click on the arrows next to "Permissions & Properties" to explore the additional options for uploading these files.



1. With these upload options you can set your class of storage, encryption, tags, meta-data, etc. Depending on your use case you may want to choose other storage classes, but for this lab we will be using the Standard storage Class. All the other settings can be left on their defaults.

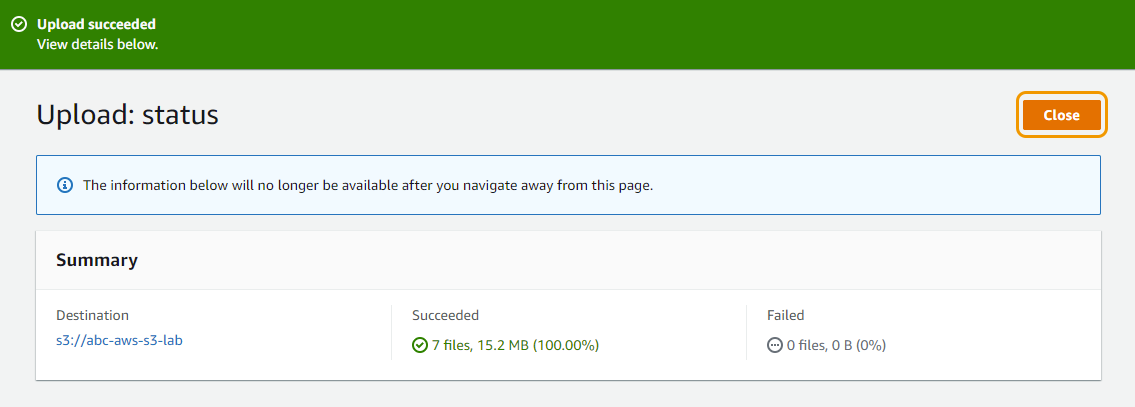
Click on the **Upload** button at the bottom of the page.



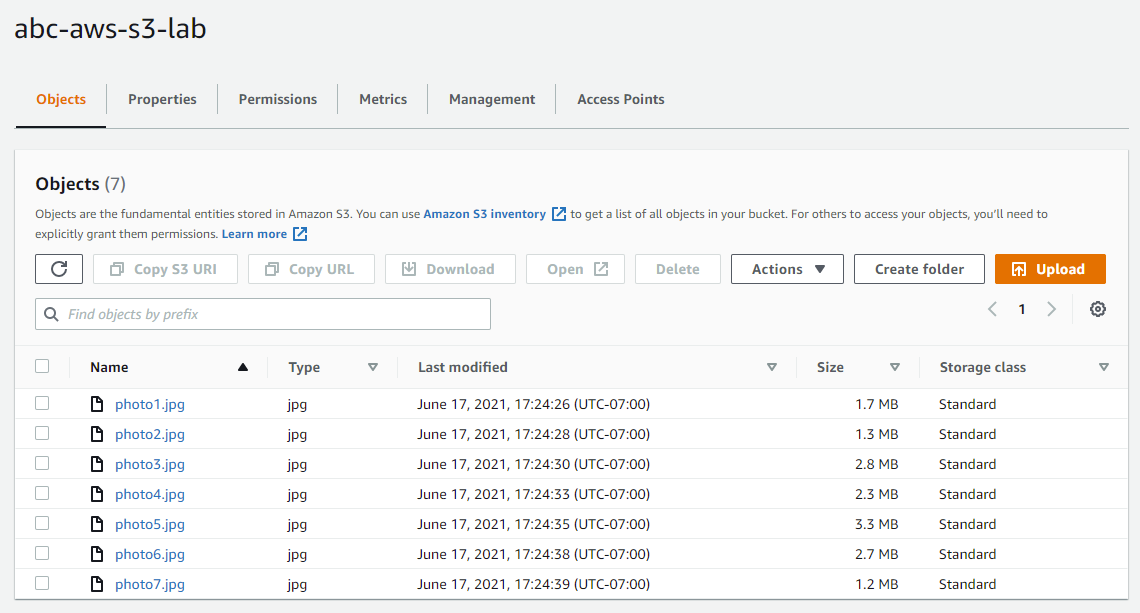
You should see your images uploading at the top of the page:



When the upload is complete you should see "Upload succeeded":



1. Now click on **Close** to return to the bucket overview page. You should see all seven photos in your bucket that are now S3 objects.

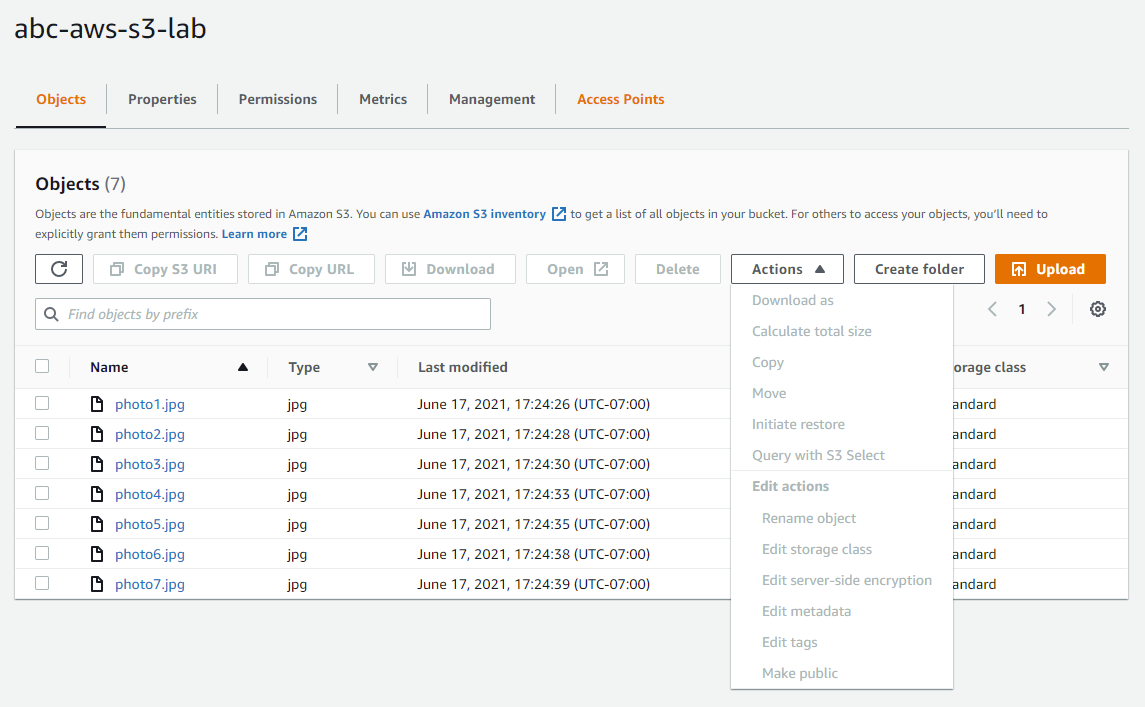


#### You are now ready to move onto the next step

**Problem #3: Working with Objects in the S3 Console**

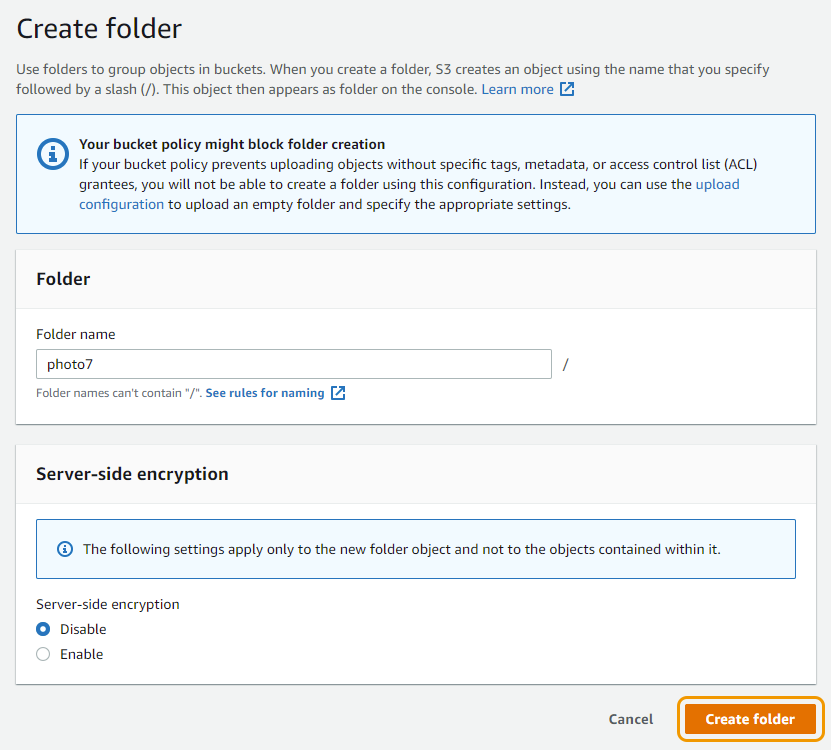
The S3 console gives you many familiar file system commands. You can rename, move, copy, delete, and view; to name a few.

You also the have the ability to create a folder (also known as a "prefix"), add or remove metadata, edit the storage class and copy the S3 URI or URL of the object. We will run through a few of these actions in this section of the lab.

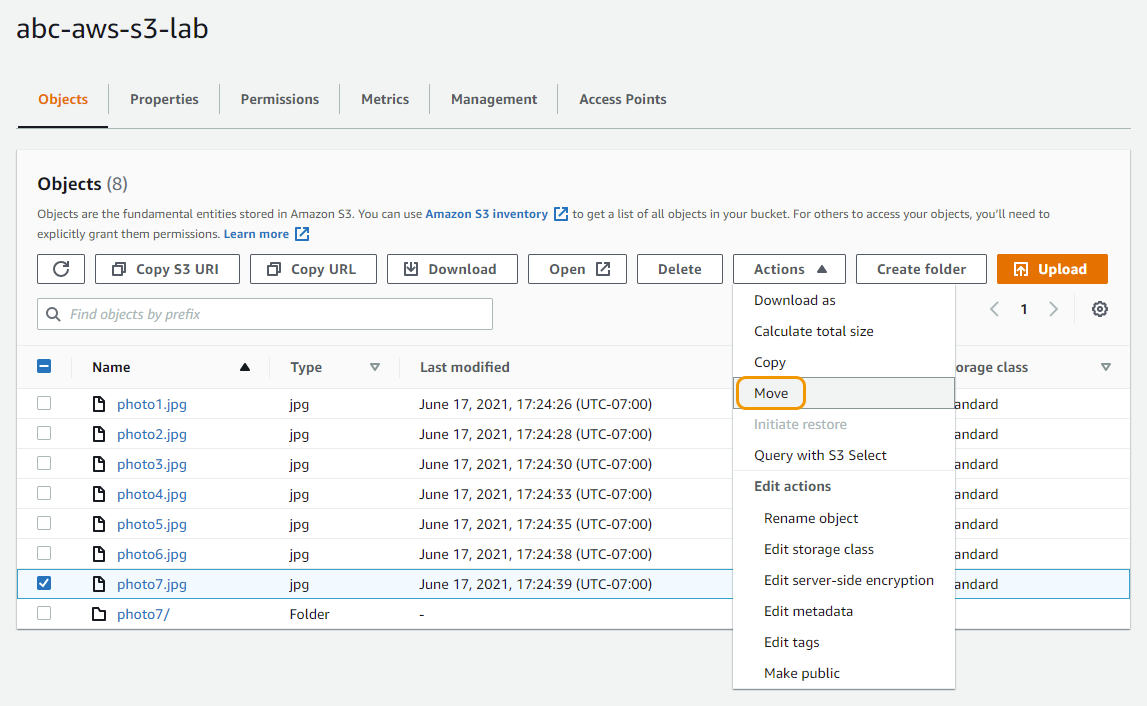


#### Moving an object in S3 With the move action in the S3 console, you can move an object to a folder (prefix) in the same bucket, to another bucket/prefix, or to an access point. For this example, we will create a new folder (prefix) and move the photo7.jpg object.

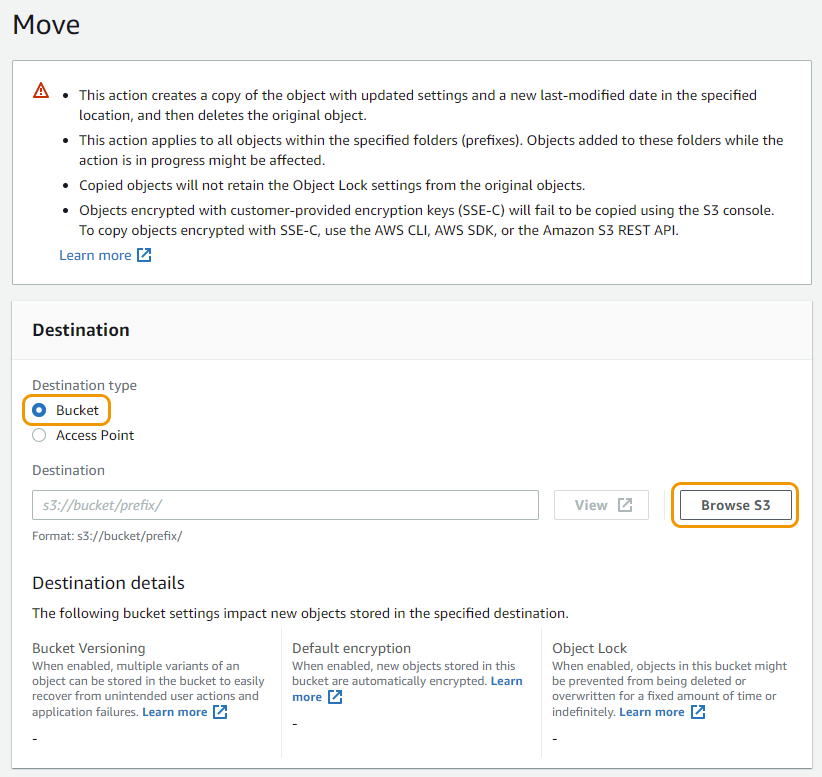
1. In your bucket's overview page select the **Create folder** button.
2. In the "Folder name" field put in the name "photo7" and then click on the **Create folder** button.



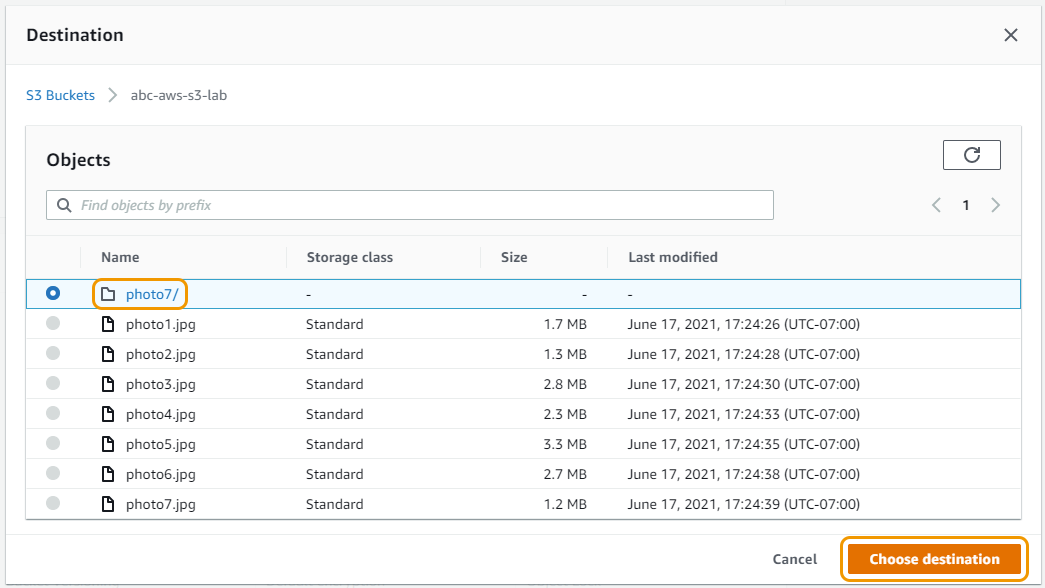
1. Now select the "photo7.jpg" object, click on the **Actions** dropdown, and select **Move**.



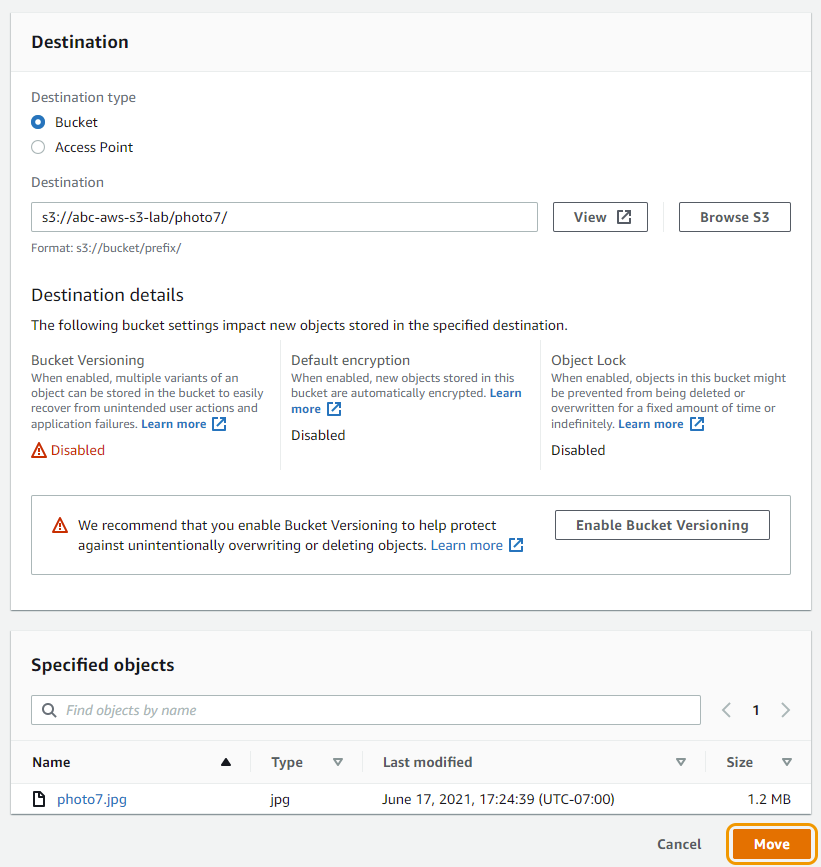
1. Under "Destination type" select **Bucket** and then click on **Browse S3**.



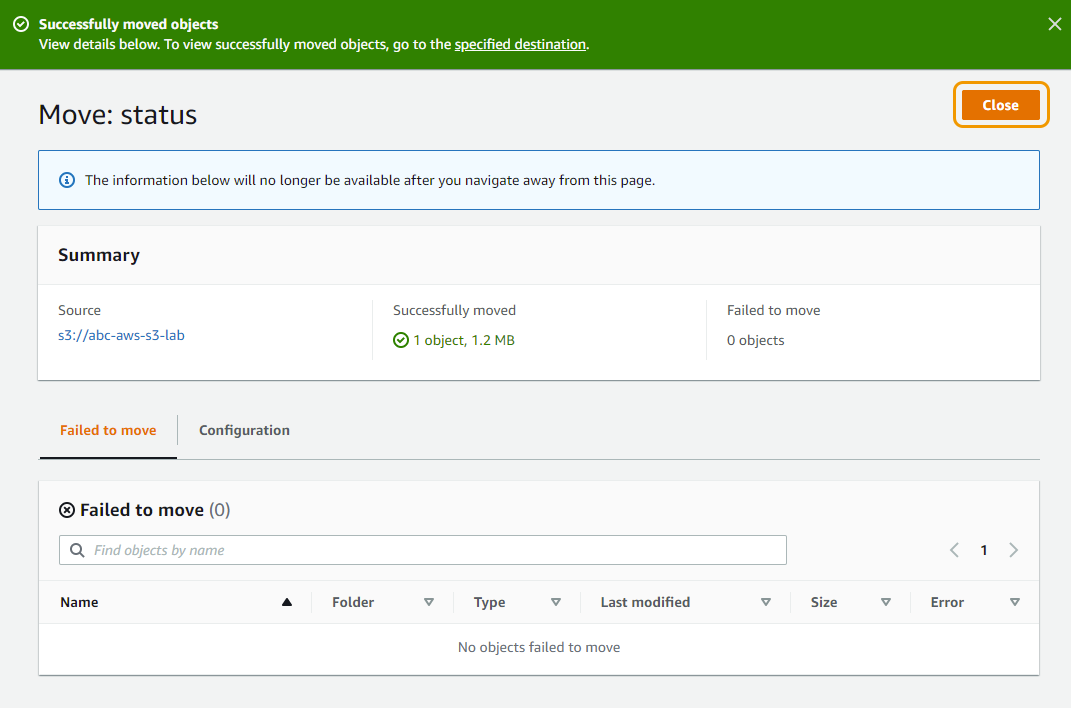
1. In the "Destination" popup select the **photo7/** folder (prefix) and then click on **Choose destination**.



1. Now that the destination has been set you can click on the **Move** button to move the object.



1. You should then be taken to the "Move: status" page showing that you "Successfully moved objects". Click on the **Close** button to leave the page.



1. You can now click on the "photo7/" folder and see that "photo7.jpg" has been moved.

Moving an object creates a copy of the object with updated settings and a new last-modified date in the specified location and then adds a delete marker to the original object.

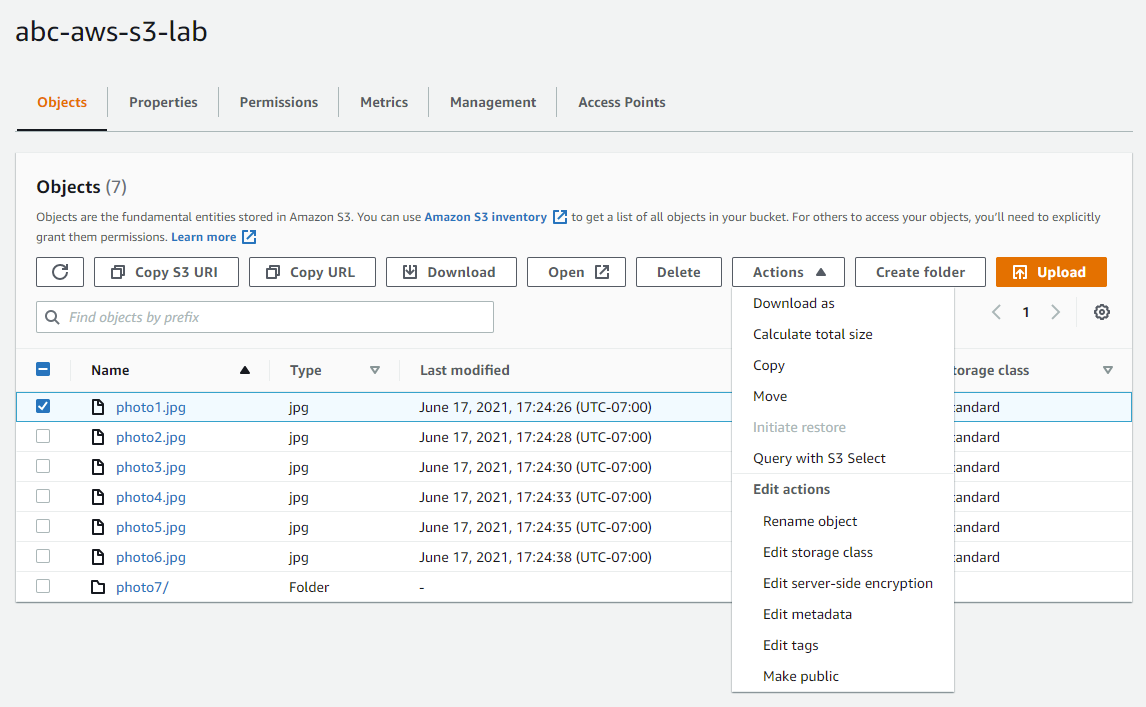
1. The steps are similar to copy an object, start by selecting the object or objects and then select **Actions** and **Copy**.

The "Move" command is one of many familiar file system commands. You can also "Download" files directly to your computer, "Rename objects", and "Open" many different types of objects directly in your browser.

#### S3 specific object actions

You will find several actions that you can take on an object specific to S3:

* **Copy S3 URI:** The S3 URI acts as an internal address for access to buckets and objects by some AWS services.
* **Copy URL:** Even though they are private unless you make them public, buckets and objects all have a URL. e.g. "photo1.jpg" in bucket "abc-aws-s3-lab" located in region "us-west-2" would have the URL <https://abc-aws-s3-lab.s3.us-west-2.amazonaws.com/photo1.jpg>.
* **Edit storage class:** This changes the [class of storage](https://docs.aws.amazon.com/AmazonS3/latest/userguide/storage-class-intro.html) for the object, this is best done by a lifecycle policy that fits the use case.
* **Edit metadata:** Object metadata is a set of name-value pairs associated with the object. This action creates a new version of the object with updated settings and a new last-modified date.
* **Edit tags:** You can tag buckets or objects to track storage costs or other criteria.



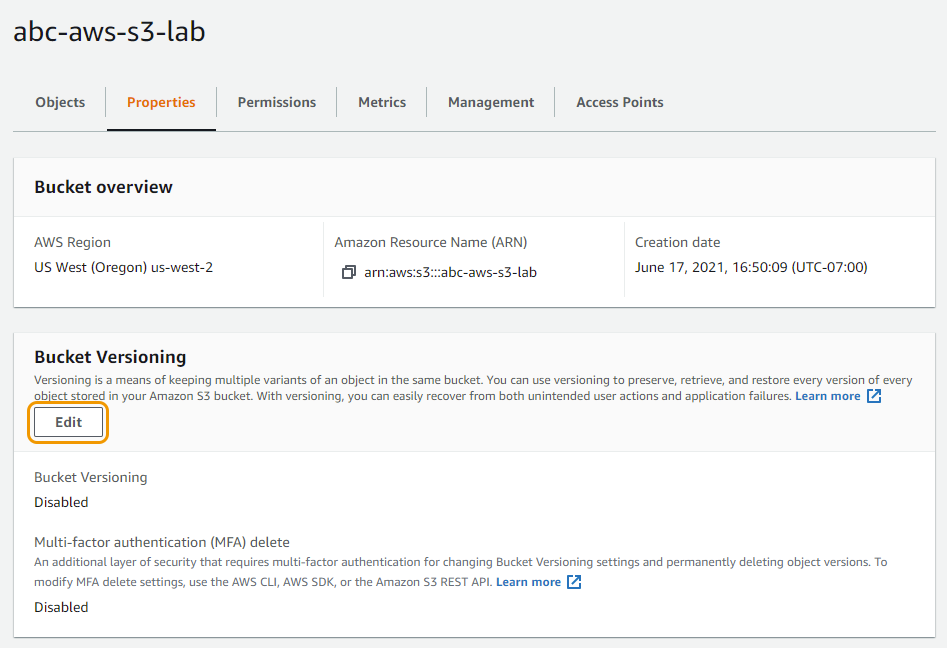
#### You are now ready to move onto the next step

Problem #4: Enabling Bucket Versioning

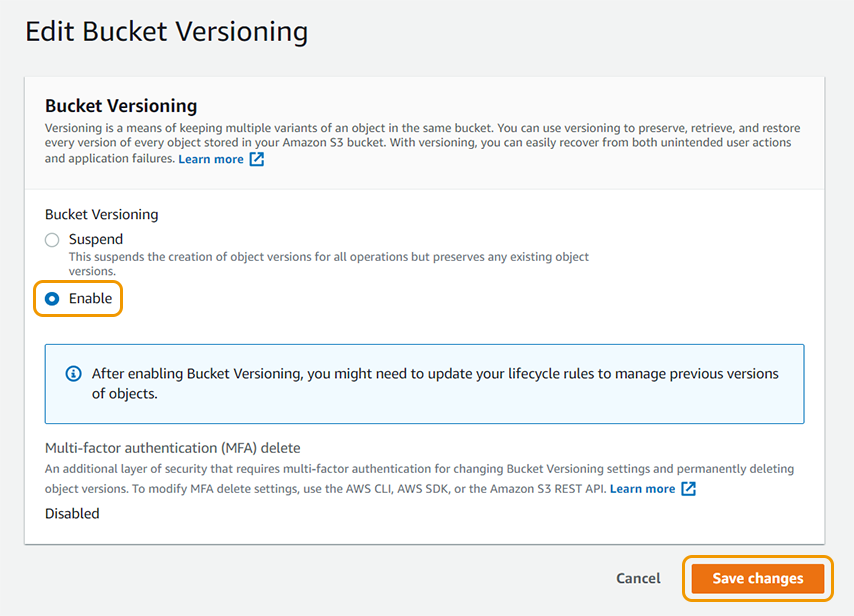
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

You enable and suspend versioning at the bucket level. After you version-enable a bucket, it can never return to an un-versioned state. But you can only suspend versioning on that bucket.

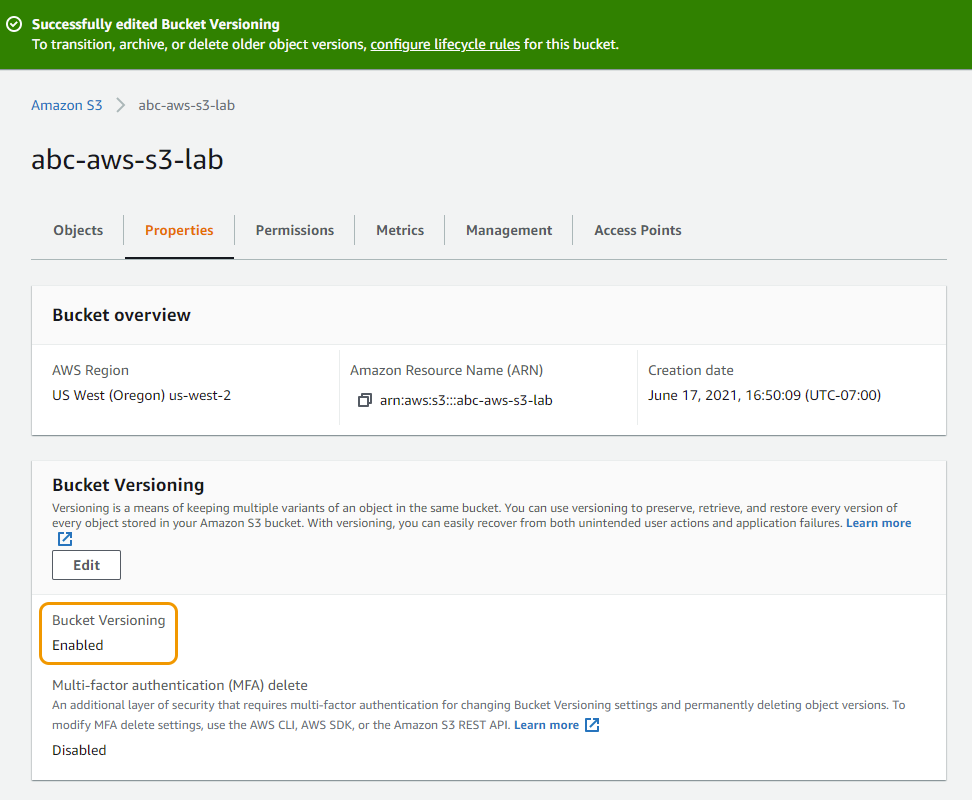
1. In the [S3 Console,](https://s3.console.aws.amazon.com/s3/) click on the **Buckets** link in the left-hand menu. Click on the name of the bucket you created earlier in the lab and then select the **Properties** tab. Under the "Bucket Versioning" heading select the **Edit** button.



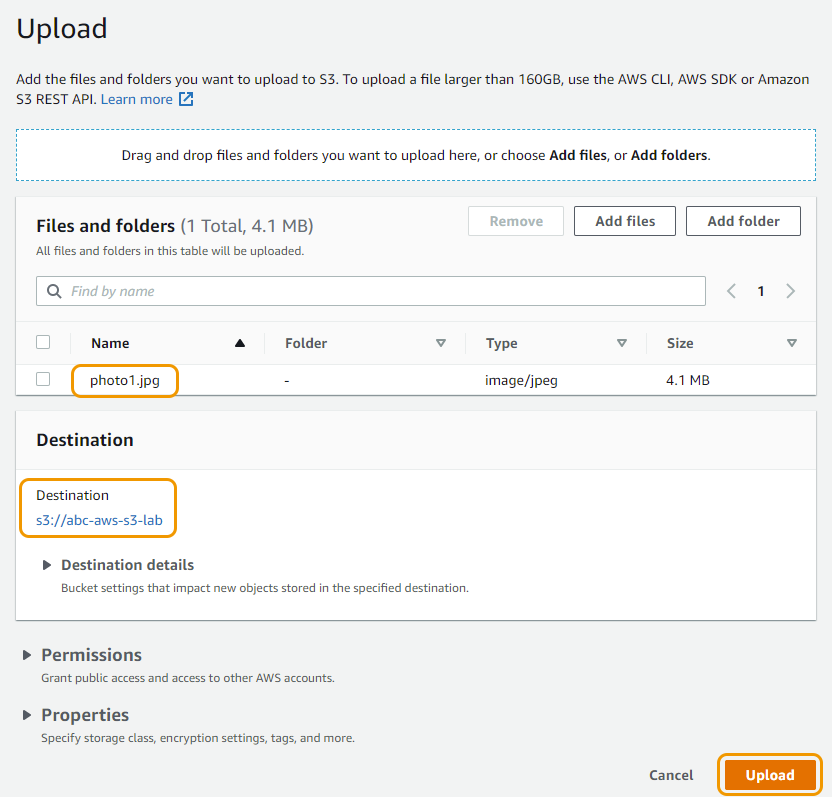
1. Select the **Enable** radio button and then click **Save Changes**.



Bucket Versioning should now show as "Enabled"

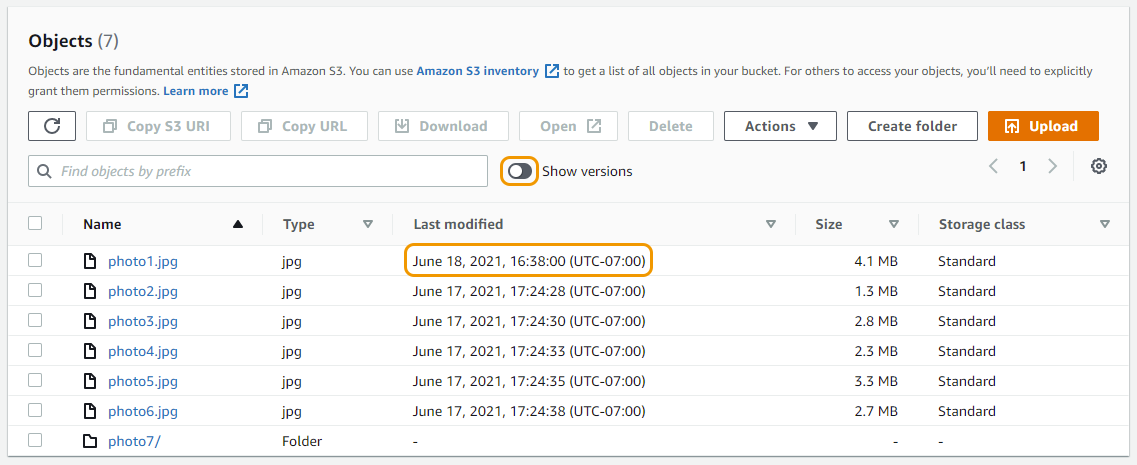


1. Now return to the "Objects" tab and select **Upload**.
2. Go to the folder with the image files you downloaded earlier in the lab and upload the "photo1.jpg" file from the "V2" folder into your S3 bucket like you did in the previous steps. Make sure you upload the photo1.jpg file into the same folder as the original photo1.jpg file.

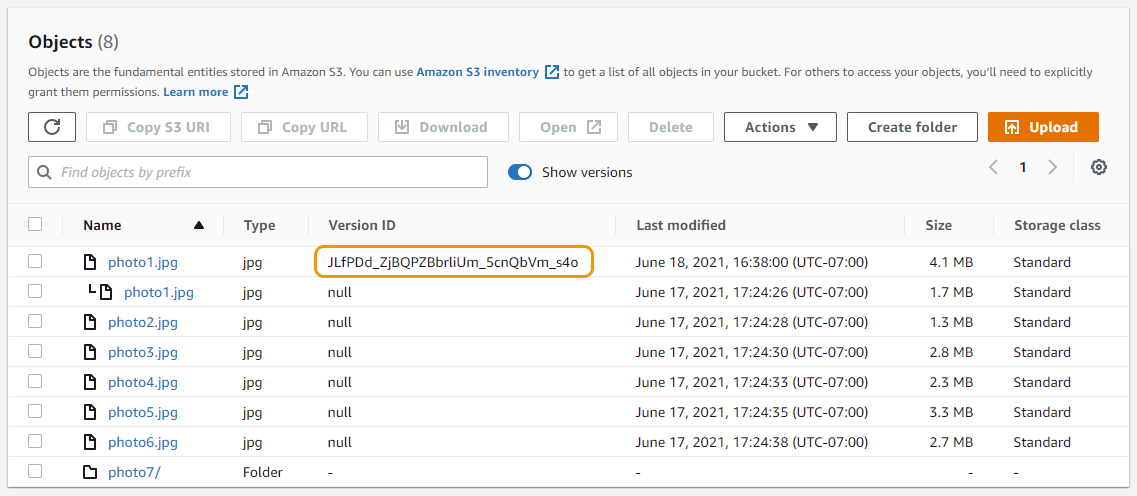


1. At first glance the contents of your bucket does not appear to have changed except for the "Last modified" date for photo1.jpg. (Which should show a more recent date/time than the other images)

Above the bucket contents you will see a toggle labeled "Show Versions", flip the toggle to display all the versions of the objects in your bucket.



1. You should now see a "Version ID" column and two different versions of "photo1.jpg". The newer version will have a unique "Version ID" and all original object Version IDs will be "null". The newer "photo1.jpg" will be the object used when the file is accessed.

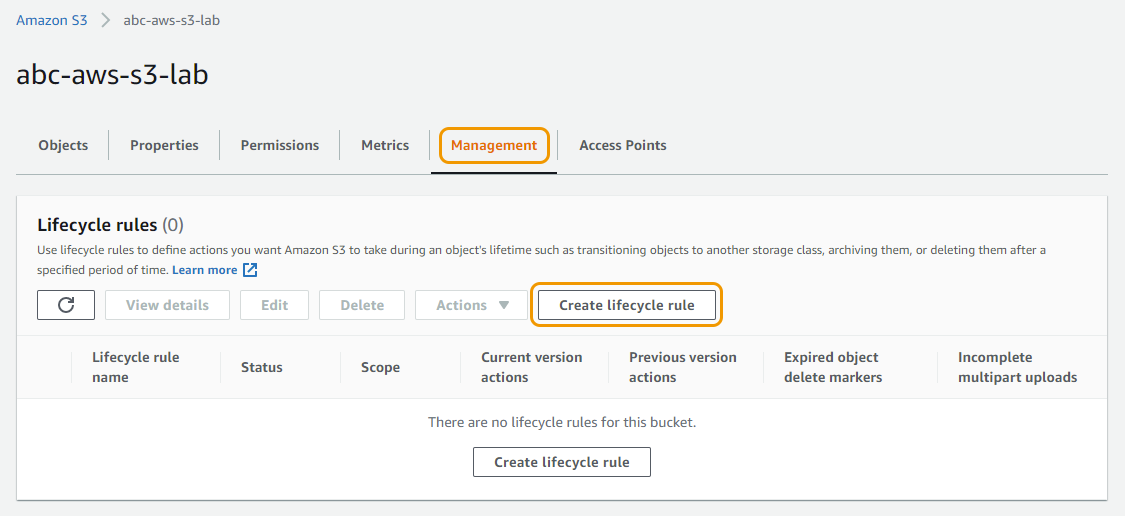


Problem #5: Setting Up a Life Cycle Policy  
You can use lifecycle policies to define actions you want Amazon S3 to take during an object's lifetime, e.g. transition objects to another storage class, archiving objects, or deleting objects after a specified period.

A versioning-enabled bucket can have many versions of the same object, one current version and zero or more noncurrent (previous) versions. Using a lifecycle policy, you can define actions specific to current and noncurrent object versions.

We are going to setup a lifecycle policy that will move noncurrent (previous) versions of your objects to the S3 Infrequent Access (IA) tier after 30 days and then delete them 30 days later.

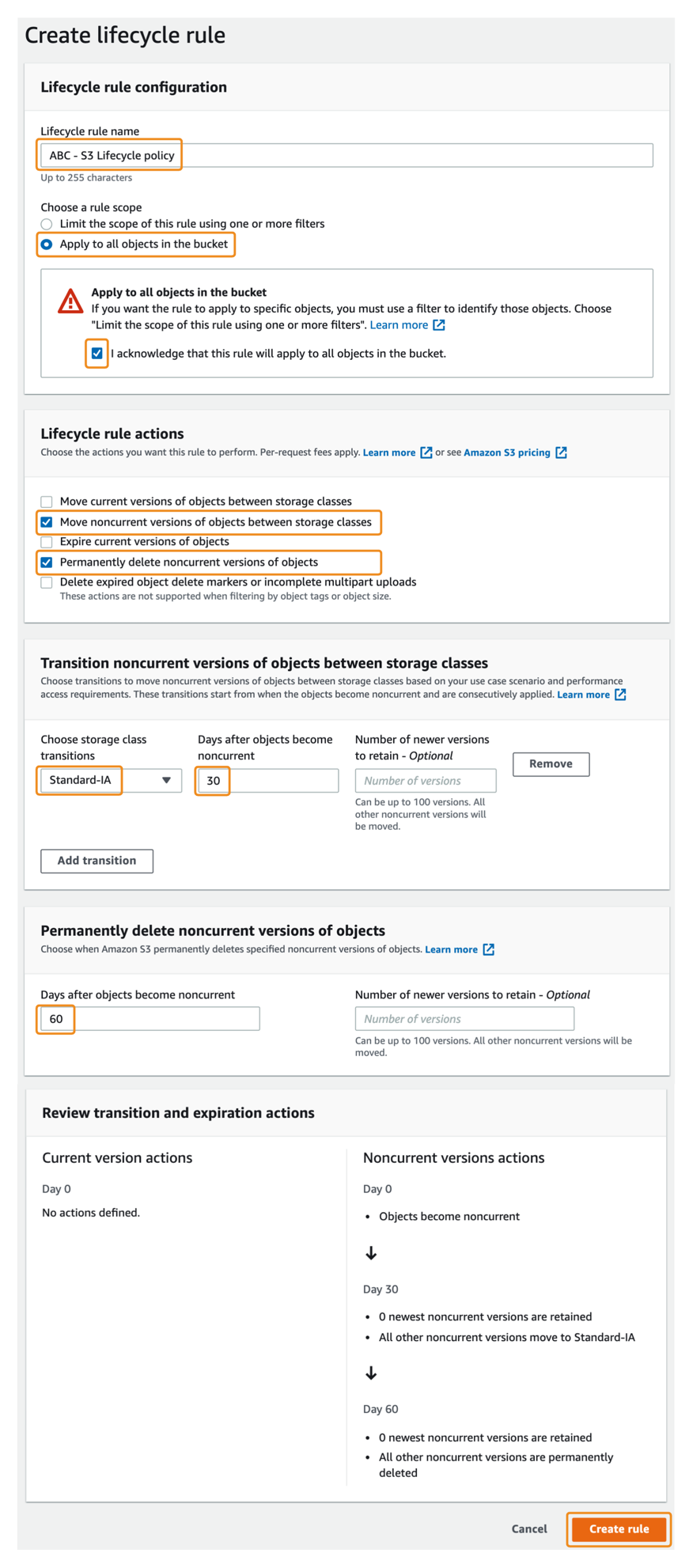
1. In your bucket's overview page, select the **Management** tab.
2. Under "Lifecycle rules" select the **Create lifecycle rule** button. This should then open the "Create lifecycle rule" page.



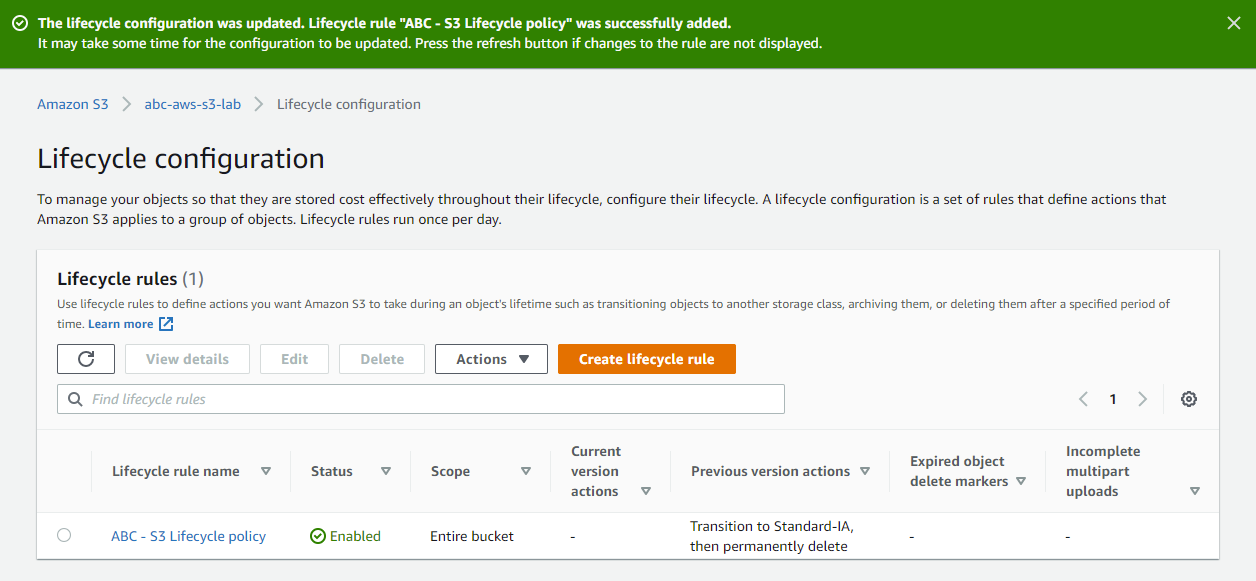
1. Give your rule the name [your initials] - S3 Lifecycle policy and select the scope as **This rule applies to all objects in the bucket** and put a **check** in the box acknowledging the warning. We could setup more fine-grained rules based on the prefix or on object tags, but for this lab we will apply it to the entire bucket.
2. Under "Lifecycle rule actions" put a check in the box next to **Move noncurrent versions of objects between storage classes** & **Permanently delete noncurrent versions of objects**. Selecting an action for a "noncurrent" version means these actions will take place on the older object version when it is replaced by a newer object version.
3. Under "Transition noncurrent versions of objects between storage classes" select **Standard-IA** for "Choose storage class transitions". Enter 30 for "Days after objects become noncurrent".

This part of the rule will move all objects from S3-Standard to S3-IA, 30 days after it becomes a previous version. This rule might be useful to save costs in S3 if the files being uploaded are frequently accessed within the first 30 days but only occasionally accessed after the first 30 days.

1. Under "Permanently delete noncurrent versions of objects" enter 60. This will delete an object 60 days after it becomes previous versions. (30 days after it is moved to S3-IA.)
2. At the bottom you will get a timeline summary of the rule you just setup. Select **Create rule** when you have finished reviewing the summary.



1. You now have a lifecycle policy that will move previous versions of your objects to S3-IA after 30 days and then delete them 30 days later.



#### You are now ready to move onto the final step

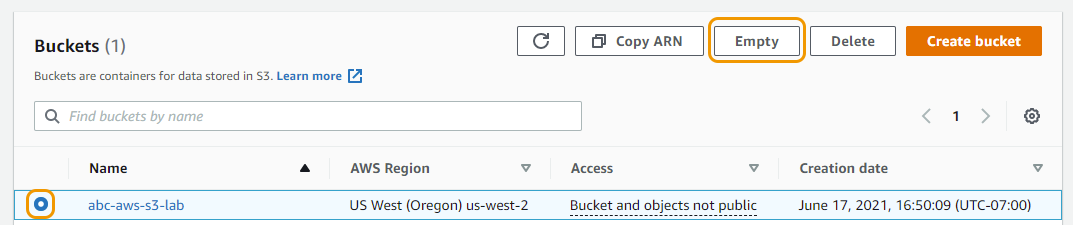
Problem #6: Deleting the Objects and the S3 Bucket  
The deletion of objects and buckets can be done programmatically through the API or via the console. If you no longer need the bucket and the objects you uploaded for this lab, you should delete them so you do not incur further charges on those objects.

You could go through the bucket and delete every file individually, but that is unnecessary when we can delete them all with one action.

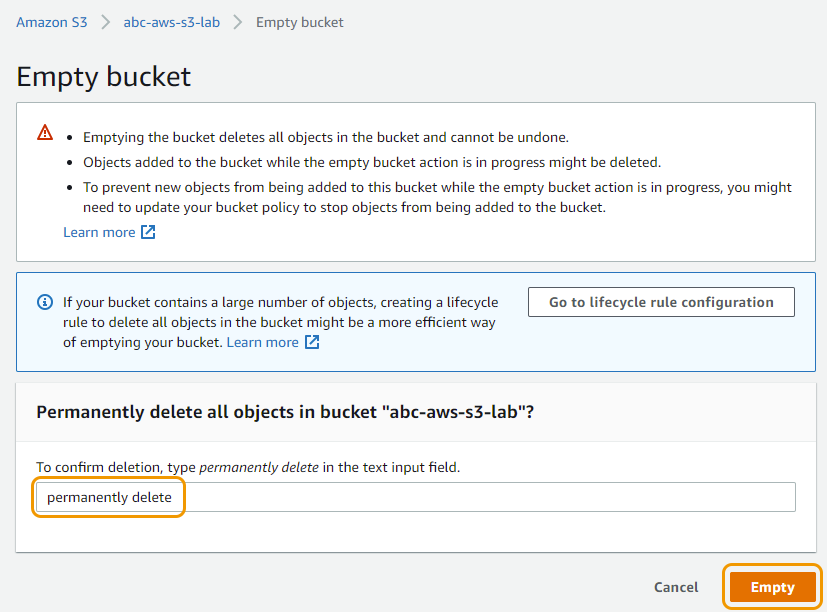
#### Delete all objects using the Empty bucket feature

If you want to delete all objects within a bucket at once, you can use the Empty option in the S3 console:

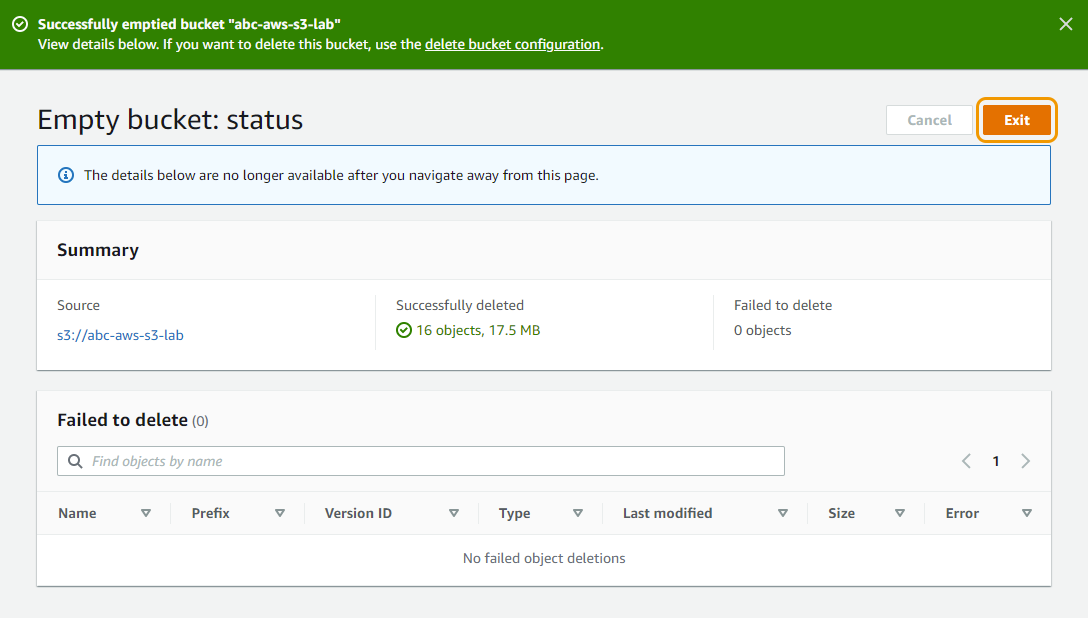
1. In the S3 console select the **radio button** to the left of your bucket and then select the **Empty** button.



1. You will be taken to the "Empty bucket" page where it gives you several warnings and requires you to type permanently delete in the confirmation field. Once entered you can then click on the **Empty** button to remove all the objects permanently.

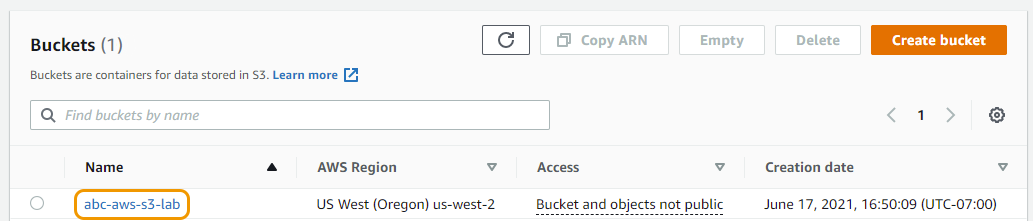


1. You will then be taken to the "Empty bucket: status" page with a message that you successfully emptied your bucket. Click on **Exit** to return to the S3 console.

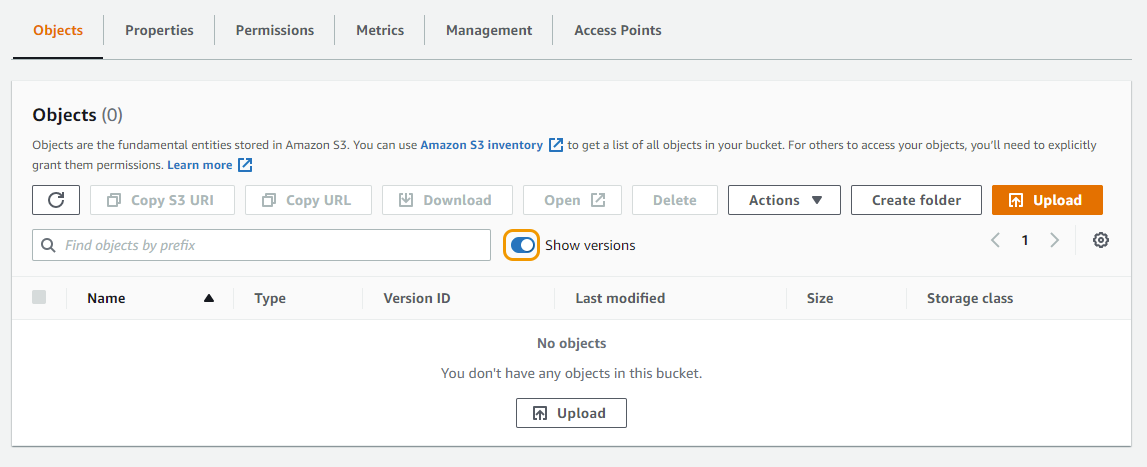


#### Delete your bucket

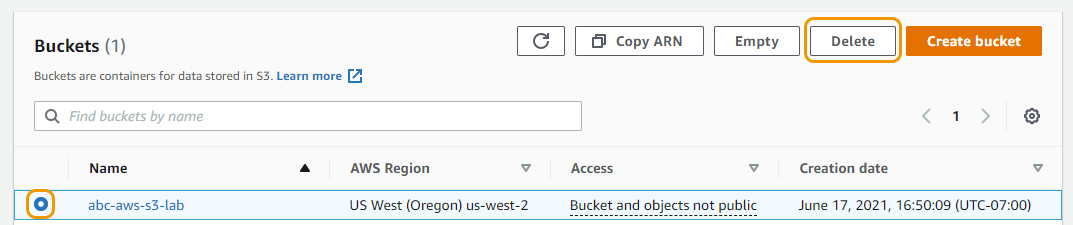
1. In the S3 console click on your bucket named **[your-bucket-name]** to open the overview page.



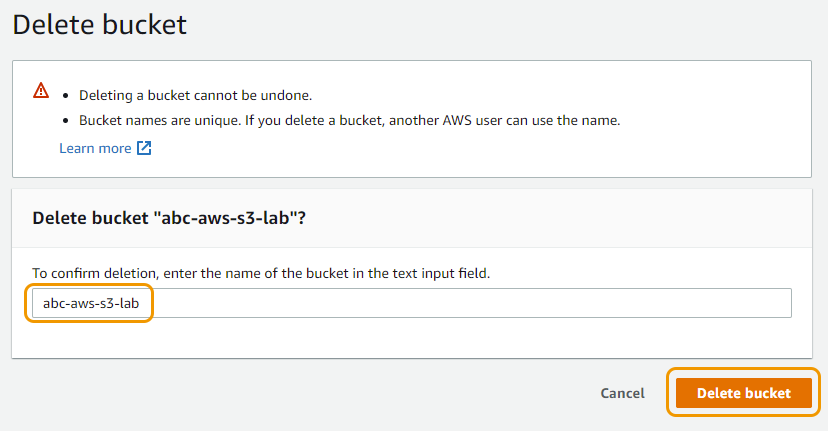
1. Your bucket should now be empty. Click on the toggle labeled "Show Versions" to confirm there aren't any previous versions of objects still in your bucket.



1. Return to the "Buckets" page in the S3 console, select the **radio button** to the left of your bucket, and then select the **Delete** button.



1. On the "Delete bucket" page you will need to type in [your-bucket-name] in the text input field and then click on **Delete bucket**.



1. Once deleted you will be returned to the S3 buckets page with the message "Successfully deleted bucket [your-bucket-name]".



Two things to keep in mind before you delete your bucket:

1. Deleting a bucket cannot be undone.
2. Bucket names are unique. If you delete a bucket, another AWS user can use the name.

Congratulations on completing the S3 lab!

Lab References:

**This lab is a slightly modified version of the S3 immersion day lab**  
https://catalog.workshops.aws/general-immersionday/en-US/basic-modules/60-s3/s3/1-s3