**Lab Steps**

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console**button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

Task 2: Create an S3 Bucket

In this task, we are going to create an S3 bucket by providing the required configurations such as name, region, and ACLs.

1. Navigate to the **Services** menu at the top. Click on **S3** in the **Storage** section.
2. On the S3 dashboard, click on **Create bucket**button.

* Bucket name: Enter ***abcxyz***
* Note: S3 bucket names are globally unique, choose a name that is available.
* Region: Select **US East (N. Virginia) us-east-1**
* Object ownership: Select **ACLs disabled (recommended)** option
* Leave other settings as default.

1. Click on the **Create bucket** button.
2. Ignore any type of warnings.

## Task 3: Upload an Object

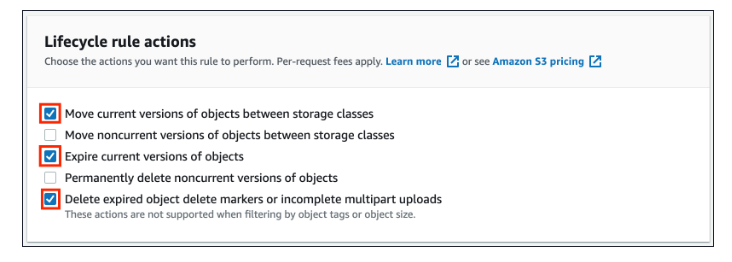
1. Click on the bucket name and click on **Upload**button.
2. Download our sample text file from [**here**](https://labresources.whizlabs.com/b94a82e4329f428b42f5de2489daf896/sample.txt) or upload any file from your machine.
3. Click on the **Add files** button.
4. **Browse** for our **sample.txt** file or the file you've chosen instead.
   * Click on the **Upload** button.
   * You can watch the progress of the upload from within the transfer panel at the bottom of the screen. Since this is a very small file, you might not see the transfer. Once your file has been uploaded, it will be displayed in the bucket.
5. Click on the **Close**button on the Top Right corner of the page.

## Task 4: Creating a Lifecycle Rule

In this task, you will learn how to set up a Lifecycle Rule for an S3 Bucket object. The rule includes storage class transitions and automatic deletion of expired objects and incomplete multipart uploads. By following the steps, you can create and enable a rule that will manage your objects according to your requirements.

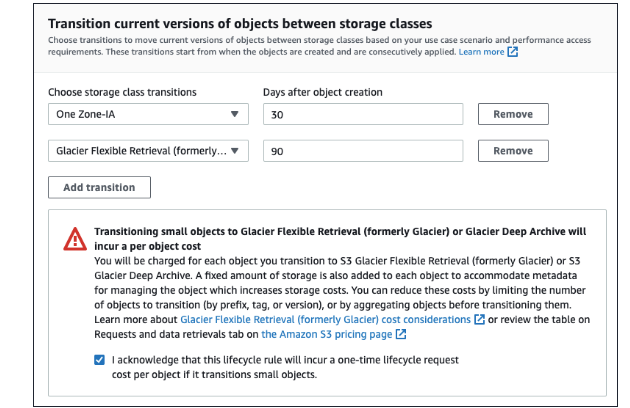
1. Click on the **Management**tab.
2. Under the **Lifecycle rules**, click on **Create lifecycle rule** to create a Lifecycle rule for the uploaded object.
3. Lifecycle rule name: Enter **whiztest**
4. Choose a rule scope: Select **Apply to all objects in the bucket**and **Acknowledge** the checkbox.

     5. Under **Lifecycle rule action**: Select the following checkboxes.



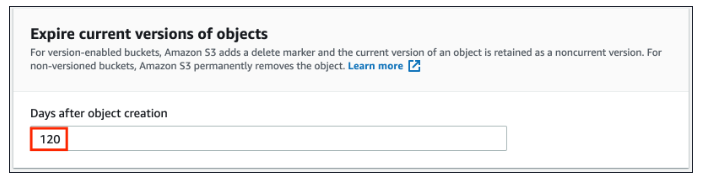
    6. For **Transition current versions of objects between storage classes** section,

* Choose storage class transitions: Select **One zone- IA**
* Days after object creation: Enter ***30***
* Click on **Add transition** button
* Choose storage class transitions: Select **Glacier Flexible Retrieval (formerly Glacier)**
* Days after object creation: Enter ***90***
* **Acknowledge** the checkbox.



   7. For **Expire current versions of objects** section,

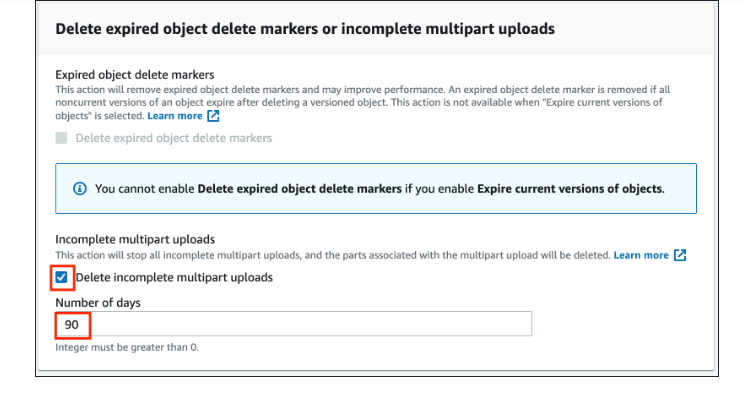
* Days after object creation: Enter **120**



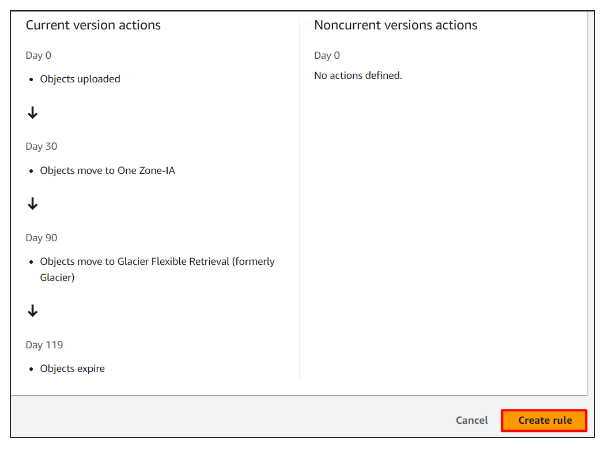
* Note: In this option, you are specifying that after **120** days, the objects will be expired (deleted automatically).

    8. For **Delete expired object delete markers or incomplete multipart uploads** Section

* For **Incomplete multipart uploads**: **Check the option** of Delete incomplete multipart uploads
* Number of days: Enter **90**



    9. Finally, review the transitions and click on **Create rule**button.



   10. Now a Lifecycle Rule has been created and is in the enabled state.

### ****Do you know ?****

Intelligent-Tiering storage class uses machine learning algorithms to automatically move objects between two access tiers: frequent access and infrequent access. This means that objects that are frequently accessed will be stored in the frequent access tier, while objects that are infrequently accessed will be moved to the infrequent access tier, helping to reduce storage costs without sacrificing performance.