

Check incomplete multipart uploads if AWS S3 storage size is abnormally big huge compared to the origin storage

You should check incomplete multipart uploads if your Metrics in the S3 Management Console shows larger size than you actually use.

In my case, I found the total size of S3 bucket was 180TB, but actually size I uploaded based on the origin storage was 40TB. I tried to compare all the files one by one, but haven't found any difference.

I learned AWS S3 keeps incomplete multipart uploads constantly and that won't be deleted as long as user does not make any actions. This is something unusual architecture I haven't expected, so I paid for that incomplete multipart uploads, because people may expect the incomplete multi uploads would be deleted automatically as long as that parts are not integrated for very long time.

- I found an article that AWS will charge the cost for the incomplete multipart uploads as long as I do not call Abort for the on-going uploads. https://docs.amazonaws.cn/en_us/AmazonS3/latest/userguide/abort-mpu.html
- The automatic discovery will be essential, because you can not know the previous handle in case of abnormal termination or something similar to that.

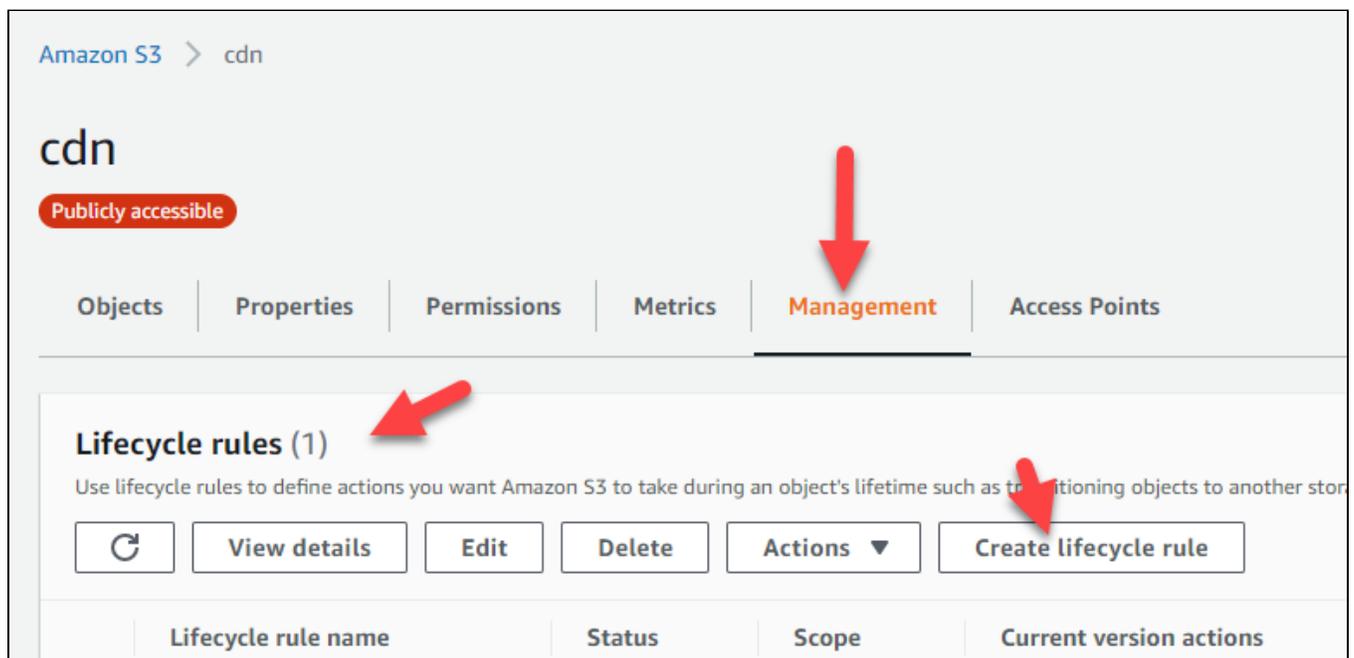
```
$s3 = new S3Client([\n    'version' => 'latest',\n    'region'  => 'us-east-1'\n]);\n.\n.\n.\nabnormal termination
```

- One of my complaint is that most of user does not know what happen until they pay more than they expected. That is what user should take care and such an information is not easy to find on AWS Web Site.

Here's the action you can do, so you do not need to waste your money for the incomplete multipart uploads.

The action user should do is "Creating lifecycle rule" in the S3 console.

What you should do is select your S3 Bucket and you will see the menu like below:



Once you "Create lifecycle rule", you will need to put/select options like below:

Create lifecycle rule

Lifecycle rule configuration

Lifecycle rule name

Up to 255 characters.

Choose a rule scope

- Limit the scope of this rule using one or more filters
- This rule applies to *all* objects in the bucket



This rule applies to *all* objects in the bucket

If you want the rule to apply to specific objects, you must use a filter to identify those objects. Choose "Limit the scope of this rule using one or more filters". [Learn more](#) 

I acknowledge that this rule will apply to all objects in the bucket.

Lifecycle rule actions

Choose the actions you want this rule to perform. Per-request fees apply. [Learn more](#) or see [Amazon S3 pricing](#)

- Transition *current* versions of objects between storage classes
- Transition *previous* versions of objects between storage classes
- Expire *current* versions of objects
- Permanently delete *previous* versions of objects
- Delete expired delete markers or incomplete multipart uploads
When a lifecycle rule is scoped with tags, these actions are unavailable.

Delete expired delete markers or incomplete multipart uploads

Expired object delete markers

This action will remove expired object delete markers and may improve performance. An expired object delete marker is retained if all previous versions of an object expire after deleting a versioned object. This action is not available when "Expire current versions of objects" is selected. [Learn more](#)

- Delete expired object delete markers

Incomplete multipart uploads

This action will stop all incomplete multipart uploads, and the parts associated with the multipart upload will be deleted. [Learn more](#)

- Delete incomplete multipart uploads

Number of days

Integer must be greater than 0.

If everything goes fine, you will see the summary like below:

Timeline summary

Current version actions

Day 0

Objects uploaded

Previous version actions

Day 0

Objects become noncurrent

The final action you should do is click "Create rule"

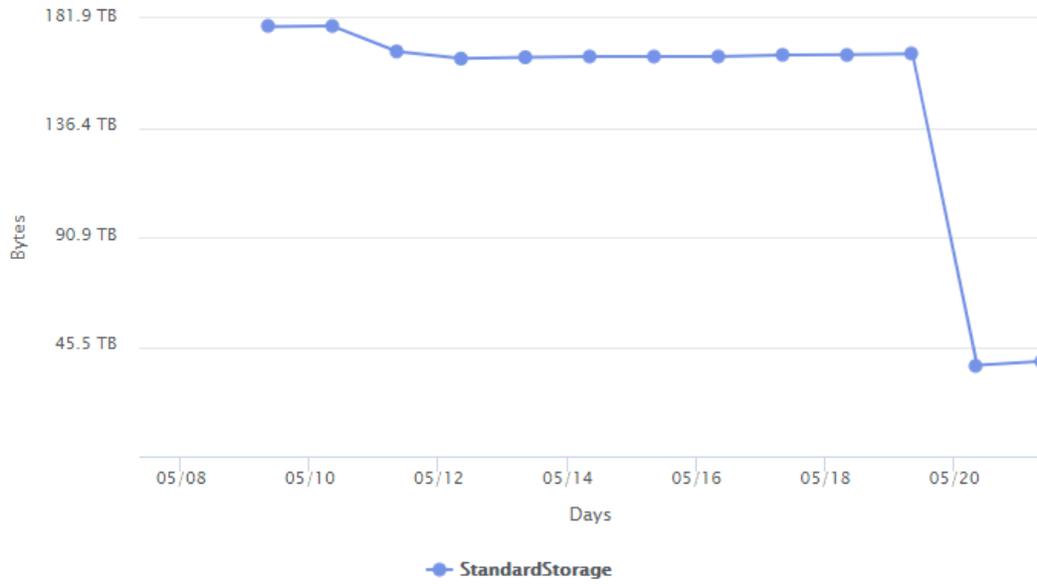
[Create rule](#)

Once you setup the above rule, your rule will be executed in 48 hours and its result will be like below:

Total bucket size

Amount of data in bytes stored in this bucket.

[View in CloudWatch](#)



Total number of objects

Total number of objects stored in this bucket for all storage classes.

[View in CloudWatch](#)

