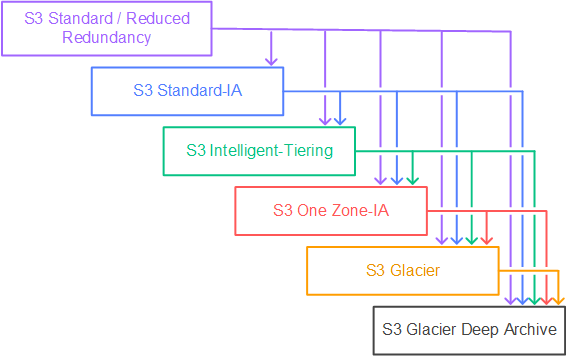
AWS S3 Object Lifecycle Management

# S3 Object Lifecycle Management

* S3 Object lifecycle can be managed by using a lifecycle configuration, which defines how S3 manages objects during their lifetime.
* Lifecycle configuration enables simplification of object lifecycle management, for e.g. moving of less frequently access objects, backup or archival of data for several years or permanent deletion of objects,
* S3 controls all transitions automatically
* Lifecycle Management rules applied to an bucket are applicable to all the existing objects in the bucket as well as the ones that will be added anew
* S3 Object lifecycle management allows 2 types of behavior
  + **Transition** in which the storage class for the objects change
  + **Expiration** where the objects expire and are permanently deleted
* Lifecycle Management can be configured with [Versioning](https://jayendrapatil.com/aws-s3-object-versioning/), which allows storage of one current object version and zero or more non current object versions
* Object’s lifecycle management applies to both Non Versioning and Versioning enabled buckets
* For Non Versioned buckets
  + Transitioning period is considered from the **object’s creation date**
* For Versioned buckets,
  + Transitioning period for current object is calculated for the object creation date
  + Transitioning period for non current object is calculated for the date **when the object became a noncurrent versioned object**
  + S3 uses the number of days since its successor was created as the number of days an object is noncurrent.
* S3 calculates the time by adding the number of days specified in the rule to the object creation time and rounding the resulting time to the next day midnight UTC for e.g., if an object was created at 15/1/2016 10:30 AM UTC and you specify 3 days in a transition rule, which results in 18/1/2016 10:30 AM UTC and rounded of to next day midnight time 19/1/2016 00:00 UTC.
* **Lifecycle configuration on MFA-enabled buckets is not supported.**
* 1000 lifecycle rules can be configured per bucket

## S3 Object Lifecycle Management Rules



### Supported lifecycle transitions

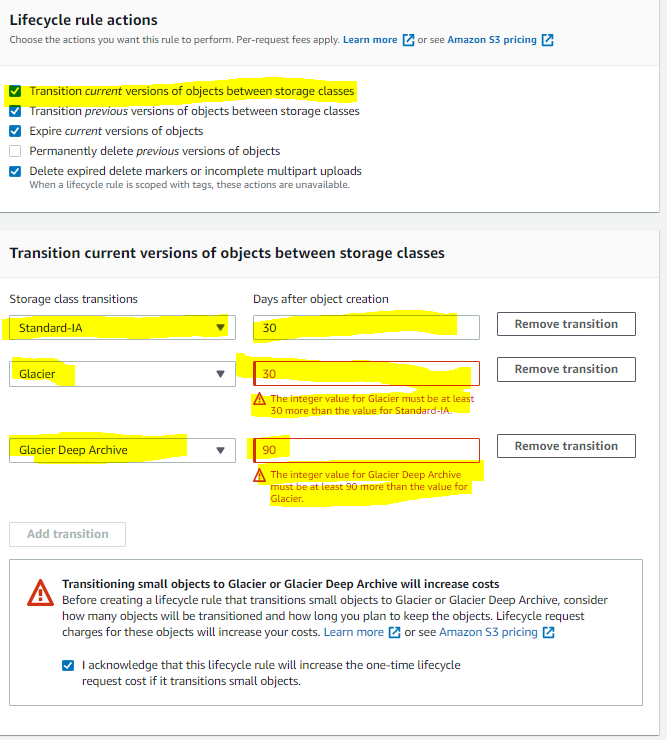
1. S3 Standard storage class –> other storage class.
2. Any storage class –> S3 Glacier or S3 Glacier Deep Archive storage classes.
3. S3 Standard-IA –> S3 Intelligent-Tiering or S3 One Zone-IA
4. S3 Intelligent-Tiering storage class –> S3 One Zone-IA storage class.
5. S3 Glacier storage class –> S3 Glacier Deep Archive storage class.

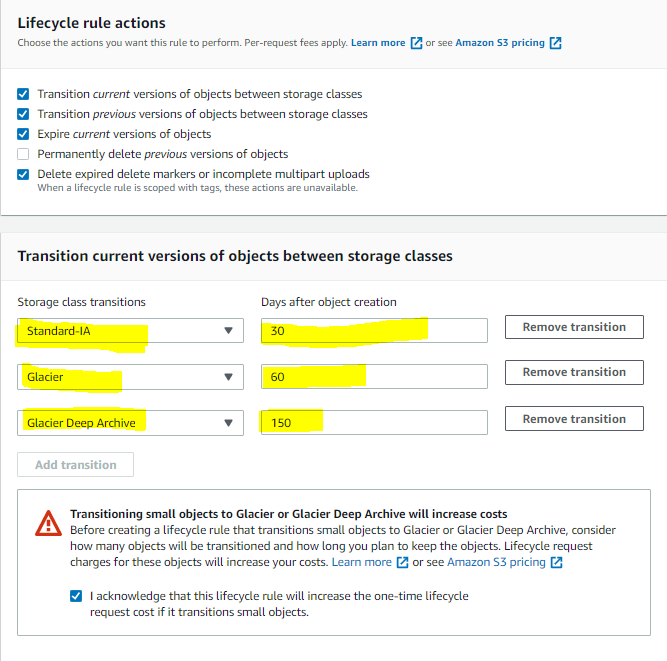
### Unsupported lifecycle transitions

1. Any storage class X S3 Standard
2. Any storage class X Reduced Redundancy
3. S3 Intelligent-Tiering X S3 Standard-IA
4. S3 One Zone-IA  X S3 Standard-IA or S3 Intelligent-Tiering

### Lifecycle transitions Constraints

1. STANDARD -> (128 KB & 30 days) -> STANDARD-IA or One Zone-IA or S3 Intelligent-Tiering
   * **Larger Objects** – Only objects with size more than 128 KB can be transitioned, as cost benefits for transitioning to STANDARD-IA or One Zone-IA can be realized only for larger objects
   * **Minimum 30 days** – Objects must be stored for at least 30 days in the current storage class before being transitioned to the STANDARD-IA or One Zone-IA, as younger objects are accessed more frequently or deleted sooner than is suitable for STANDARD-IA or One Zone-IA
2. GLACIER -> (90 days) -> Permanent Deletion OR GLACIER Deep Archive -> (180 days) -> Permanent Deletion
   * Deleting data that is archived to Glacier is free, if the objects deleted are archived for three months or longer.
   * S3 charges a prorated early deletion fee, if the object is deleted or overwritten within three months of archiving it.
3. Archival of objects to Glacier by using object lifecycle management is performed asynchronously and there may be a delay between the transition date in the lifecycle configuration rule and the date of the physical transition. However, AWS charges Glacier prices based on the transition date specified in the rule
4. For a versioning-enabled bucket
   * Transition and Expiration actions apply to current versions.
   * NoncurrentVersionTransition and NoncurrentVersionExpiration actions apply to noncurrent versions and works similar to the non versioned objects except the time period is from the time the objects became noncurrent
5. Expiration Rules
   * For Non Versioned bucket
     + Object is permanently deleted
   * For Versioned bucket
     + Expiration is applicable to the Current object only and does not impact any of the non current objects
     + S3 will insert a Delete Marker object with unique id and the previous current object becomes a non current version
     + S3 will not take any action if the Current object is a Delete Marker
     + If the bucket has a single object which is the Delete Marker (referred to as expired object delete marker), S3 removes the Delete Marker
   * For Versioned Suspended bucket
     + S3 will insert a Delete Marker object with version ID null and overwrite the any object with version ID null
6. When an object reaches the end of its lifetime, S3 queues it for removal and removes it asynchronously. There may be a delay between the expiration date and the date at which S3 removes an object. Charged for storage time associated with an object that has expired are not incurred.
7. Cost is incurred if objects are expired in STANDARD-IA before 30 days, GLACIER before 90 days and GLACIER\_DEEP\_ARCHIVE before 180 days.





## AWS Certification Exam Practice Questions

1. If an object is stored in the Standard S3 storage class and you want to move it to Glacier, what must you do in order to properly migrate it?
   1. Change the storage class directly on the object.
   2. Delete the object and re-upload it, selecting Glacier as the storage class.
   3. None of the above.
   4. **Create a lifecycle policy that will migrate it after a minimum of 30 days.** (Any object uploaded to S3 must first be placed into either the Standard, Reduced Redundancy, or Infrequent Access storage class. Once in S3 the only way to move the object to glacier is through a lifecycle policy)
2. A company wants to store their documents in AWS. Initially, these documents will be used frequently, and after a duration of 6 months, they would not be needed anymore. How would you architect this requirement?
   1. Store the files in Amazon EBS and create a Lifecycle Policy to remove the files after 6 months.
   2. **Store the files in Amazon S3 and create a Lifecycle Policy to remove the files after 6 months.**
   3. Store the files in Amazon Glacier and create a Lifecycle Policy to remove the files after 6 months.
   4. Store the files in Amazon EFS and create a Lifecycle Policy to remove the files after 6 months.
3. Your firm has uploaded a large amount of aerial image data to S3. In the past, in your on-premises environment, you used a dedicated group of servers to oaten process this data and used Rabbit MQ, an open source messaging system, to get job information to the servers. Once processed the data would go to tape and be shipped offsite. Your manager told you to stay with the current design, and leverage AWS archival storage and messaging services to minimize cost. Which is correct?
   1. Use SQS for passing job messages, use Cloud Watch alarms to terminate EC2 worker instances when they become idle. Once data is processed, change the storage class of the S3 objects to Reduced Redundancy Storage (Need to replace On-Premises Tape functionality)
   2. Setup Auto-Scaled workers triggered by queue depth that use spot instances to process messages in SQS. Once data is processed, change the storage class of the S3 objects to Reduced Redundancy Storage (Need to replace On-Premises Tape functionality)
   3. **Setup Auto-Scaled workers triggered by queue depth that use spot instances to process messages in SQS. Once data is processed, change the storage class of the S3 objects to Glacier**(Glacier suitable for Tape backup)
   4. Use SNS to pass job messages use Cloud Watch alarms to terminate spot worker instances when they become idle. Once data is processed, change the storage class of the S3 object to Glacier.
4. You have a proprietary data store on-premises that must be backed up daily by dumping the data store contents to a single compressed 50GB file and sending the file to AWS. Your SLAs state that any dump file backed up within the past 7 days can be retrieved within 2 hours. Your compliance department has stated that all data must be held indefinitely. The time required to restore the data store from a backup is approximately 1 hour. Your on-premise network connection is capable of sustaining 1gbps to AWS. Which backup methods to AWS would be most cost-effective while still meeting all of your requirements?
   1. Send the daily backup files to Glacier immediately after being generated (will not meet the RTO)
   2. Transfer the daily backup files to an EBS volume in AWS and take daily snapshots of the volume (Not cost effective)
   3. Transfer the daily backup files to S3 and use appropriate bucket lifecycle policies to send to Glacier (Store in S3 for seven days and then archive to Glacier)
   4. Host the backup files on a Storage Gateway with Gateway-Cached Volumes and take daily snapshots (Not Cost effective as local storage as well as S3 storage)