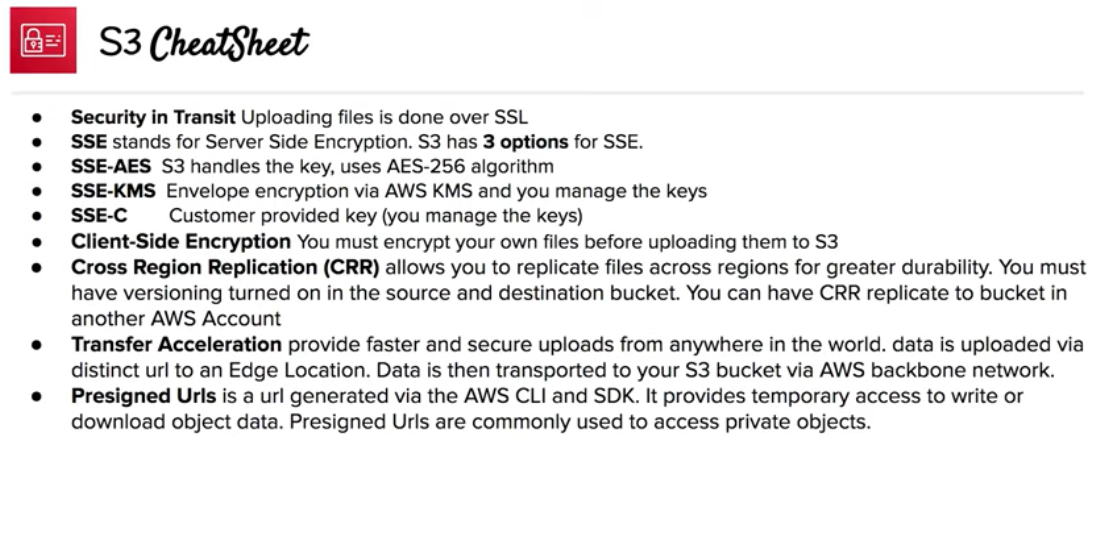
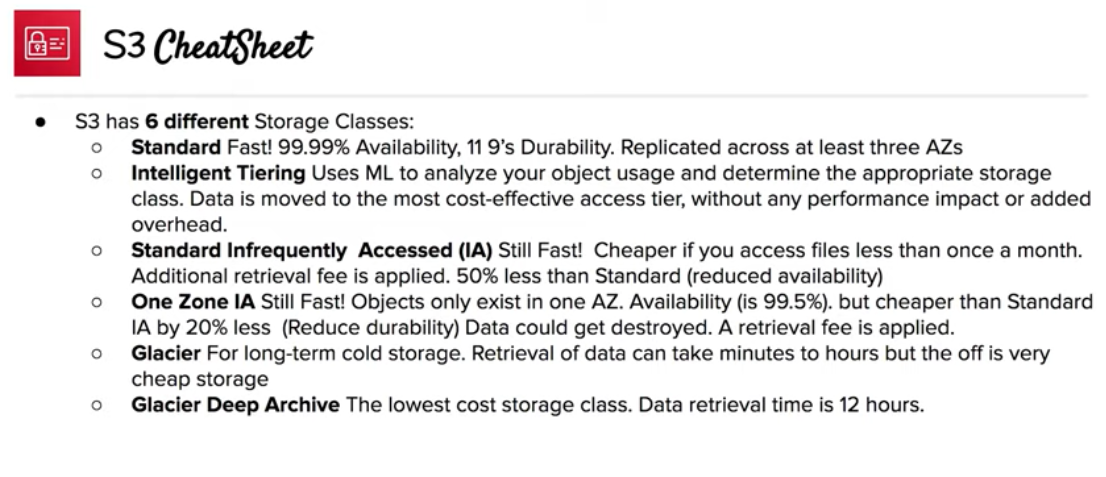
AWS S3 Best Practices

# 





# GET-intensive Workloads

* CloudFront can be used for performance optimization and can help by
  + distributing content with low latency and high data transfer rate.
  + caching the content and thereby reducing the number of direct requests to S3
  + providing multiple endpoints (Edge locations) for data availability
  + available in two flavors as Web distribution or RTMP distribution
* To fast data transport over long distances between a client and an S3 bucket, use [Amazon S3 Transfer Acceleration](https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html). Transfer Acceleration uses the globally distributed edge locations in CloudFront to accelerate data transport over geographical distances

### PUTs/GETs for Large Objects

* AWS allows Parallelizing the PUTs/GETs request to improve the upload and download performance as well as the ability to recover in case it fails
* For PUTs, **Multipart upload** can help improve the uploads by
  + performing multiple uploads at the same time and maximizing network bandwidth utilization
  + quick recovery from failures, as only the part that failed to upload needs to be re-uploaded
  + ability to pause and resume uploads
  + begin an upload before the Object size is known
* For GETs, **range http header** can help to improve the downloads by
  + allowing the object to be retrieved in parts instead of the whole object
  + quick recovery from failures, as only the part that failed to download needs to be retried.

### List Operations

* Object key names are stored lexicographically in S3 indexes, making it hard to sort and manipulate the contents of LIST
* S3 maintains a single lexicographically sorted list of indexes
* Build and maintain Secondary Index outside of S3 for e.g. DynamoDB or RDS to store, index and query objects metadata rather then performing operations on S3

## Security

* Use **Versioning**
  + can be used to protect from unintended overwrites and deletions
  + allows the ability to retrieve and restore deleted objects or rollback to previous versions
* Enable additional security by configuring a bucket to enable MFA (Multi-Factor Authentication) delete
* **Versioning does not prevent Bucket deletion** and must be backed up, as if accidentally or maliciously deleted the data is lost
* Use **Cross Region replication** feature to backup data to a different region
* When using VPC with S3, use VPC S3 endpoints as
  + are horizontally scaled, redundant, and highly available VPC components
  + help establish a private connection between VPC and S3 and the traffic never leaves the Amazon network

## Cost

* Optimize S3 storage cost by selecting an appropriate storage class for objects
* Configure appropriate lifecycle management rules to move objects to different storage classes and expire them

## Tracking

* Use Event Notifications to be notified for any put or delete request on the S3 objects
* Use CloudTrail, which helps capture specific API calls made to S3 from the AWS account and delivers the log files to an S3 bucket
* Use CloudWatch to monitor the Amazon S3 buckets, tracking metrics such as object counts and bytes stored and configure appropriate actions

## AWS Certification Exam Practice Questions

1. A media company produces new video files on-premises every day with a total size of around 100GB after compression. All files have a size of 1-2 GB and need to be uploaded to Amazon S3 every night in a fixed time window between 3am and 5am. Current upload takes almost 3 hours, although less than half of the available bandwidth is used. What step(s) would ensure that the file uploads are able to complete in the allotted time window?
   1. Increase your network bandwidth to provide faster throughput to S3
   2. **Upload the files in parallel to S3 using multipart upload**
   3. Pack all files into a single archive, upload it to S3, then extract the files in AWS
   4. Use AWS Import/Export to transfer the video files
2. You are designing a web application that stores static assets in an Amazon Simple Storage Service (S3) bucket. You expect this bucket to immediately receive over 150 PUT requests per second. What should you do to ensure optimal performance?
   1. Use multi-part upload.
   2. **Add a random prefix to the key names. (so that sequential naming doesn’t with incoming traffic)**
   3. Amazon S3 will automatically manage performance at this scale.
   4. Use a predictable naming scheme, such as sequential numbers or date time sequences, in the key names
3. You have an application running on an Amazon Elastic Compute Cloud instance, that uploads 5 GB video objects to Amazon Simple Storage Service (S3). Video uploads are taking longer than expected, resulting in poor application performance. Which method will help improve performance of your application?
   1. Enable enhanced networking
   2. **Use Amazon S3 multipart upload**
   3. Leveraging Amazon CloudFront, use the HTTP POST method to reduce latency.
   4. Use Amazon Elastic Block Store Provisioned IOPs and use an Amazon EBS-optimized instance
4. Which of the following methods gives you protection against accidental loss of data stored in Amazon S3? (Choose 2)
   1. **Set bucket policies to restrict deletes, and also enable versioning**
   2. By default, versioning is enabled on a new bucket so you don’t have to worry about it (Not enabled by default)
   3. Build a secondary index of your keys to protect the data (improves performance only)
   4. **Back up your bucket to a bucket owned by another AWS account for redundancy**
5. A startup company hired you to help them build a mobile application that will ultimately store billions of image and videos in Amazon S3. The company is lean on funding, and wants to minimize operational costs, however, they have an aggressive marketing plan, and expect to double their current installation base every six months. Due to the nature of their business, they are expecting sudden and large increases to traffic to and from S3, and need to ensure that it can handle the performance needs of their application. What other information must you gather from this customer in order to determine whether S3 is the right option?
   1. You must know how many customers that company has today, because this is critical in understanding what their customer base will be in two years. (No. of customers do not matter)
   2. **You must find out total number of requests per second at peak usage.**
   3. You must know the size of the individual objects being written to S3 in order to properly design the key namespace. (Size does not relate to the key namespace design but the count does)
   4. In order to build the key namespace correctly, you must understand the total amount of storage needs for each S3 bucket. (S3 provided unlimited storage the key namespace design would depend on the number)
6. A document storage company is deploying their application to AWS and changing their business model to support both free tier and premium tier users. The premium tier users will be allowed to store up to 200GB of data and free tier customers will be allowed to store only 5GB. The customer expects that billions of files will be stored. All users need to be alerted when approaching 75 percent quota utilization and again at 90 percent quota use. To support the free tier and premium tier users, how should they architect their application?
   1. **The company should utilize an amazon simple workflow service activity worker that updates the users data counter in amazon dynamo DB. The activity worker will use simple email service to send an email if the counter increases above the appropriate thresholds.**
   2. The company should deploy an amazon relational data base service relational database with a store objects table that has a row for each stored object along with size of each object. The upload server will query the aggregate consumption of the user in questions (by first determining the files store by the user, and then querying the stored objects table for respective file sizes) and send an email via Amazon Simple Email Service if the thresholds are breached. (Good Approach to use RDS but with so many objects might not be a good option)
   3. The company should write both the content length and the username of the files owner as S3 metadata for the object. They should then create a file watcher to iterate over each object and aggregate the size for each user and send a notification via Amazon Simple Queue Service to an emailing service if the storage threshold is exceeded. (List operations on S3 not feasible)
   4. The company should create two separated amazon simple storage service buckets one for data storage for free tier users and another for data storage for premium tier users. An amazon simple workflow service activity worker will query all objects for a given user based on the bucket the data is stored in and aggregate storage. The activity worker will notify the user via Amazon Simple Notification Service when necessary (List operations on S3 not feasible as well as SNS does not address email requirement)
7. Your company host a social media website for storing and sharing documents. the web application allow users to upload large files while resuming and pausing the upload as needed. Currently, files are uploaded to your php front end backed by Elastic Load Balancing and an autoscaling fleet of amazon elastic compute cloud (EC2) instances that scale upon average of bytes received (NetworkIn) After a file has been uploaded. it is copied to amazon simple storage service(S3). Amazon Ec2 instances use an AWS Identity and Access Management (AMI) role that allows Amazon s3 uploads. Over the last six months, your user base and scale have increased significantly, forcing you to increase the auto scaling groups Max parameter a few times. Your CFO is concerned about the rising costs and has asked you to adjust the architecture where needed to better optimize costs. Which architecture change could you introduce to reduce cost and still keep your web application secure and scalable?
   1. Replace the Autoscaling launch Configuration to include c3.8xlarge instances; those instances can potentially yield a network throughput of 10gbps. (no info of current size and might increase cost)
   2. Re-architect your ingest pattern, have the app authenticate against your identity provider as a broker fetching temporary AWS credentials from AWS Secure token service (GetFederation Token). Securely pass the credentials and s3 endpoint/prefix to your app. Implement client-side logic to directly upload the file to amazon s3 using the given credentials and S3 Prefix. (will not provide the ability to handle pause and restarts)
   3. Re-architect your ingest pattern, and move your web application instances into a VPC public subnet. Attach a public IP address for each EC2 instance (using the auto scaling launch configuration settings). Use Amazon Route 53 round robin records set and http health check to DNS load balance the app request this approach will significantly reduce the cost by bypassing elastic load balancing. (ELB is not the bottleneck)
   4. **Re-architect your ingest pattern, have the app authenticate against your identity provider as a broker fetching temporary AWS credentials from AWS Secure token service (GetFederation Token). Securely pass the credentials and s3 endpoint/prefix to your app. Implement client-side logic that used the S3 multipart upload API to directly upload the file to Amazon s3 using the given credentials and s3 Prefix.**(multipart allows one to start uploading directly to S3 before the actual size is known or complete data is downloaded)
8. If an application is storing hourly log files from thousands of instances from a high traffic web site, which naming scheme would give optimal performance on S3?
   1. Sequential
   2. instanceID\_log-HH-DD-MM-YYYY
   3. instanceID\_log-YYYY-MM-DD-HH
   4. **HH-DD-MM-YYYY-log\_instanceID**(HH will give some randomness to start with instead of instaneId where the first characters would be i-)
   5. YYYY-MM-DD-HH-log\_instanceID