AWS S3 Subresources

* S3 Subresources provides support to store, and manage the bucket configuration information
* S3 subresources only exist in the context of a specific bucket or object
* S3 defines a set of subresources associated with buckets and objects.
* S3 Subresources are subordinates to objects; that is, they do not exist on their own, they are always associated with some other entity, such as an object or a bucket
* S3 supports various options to configure a bucket for e.g., bucket can be configured for website hosting, configuration added to manage lifecycle of objects in the bucket, and to log all access to the bucket.

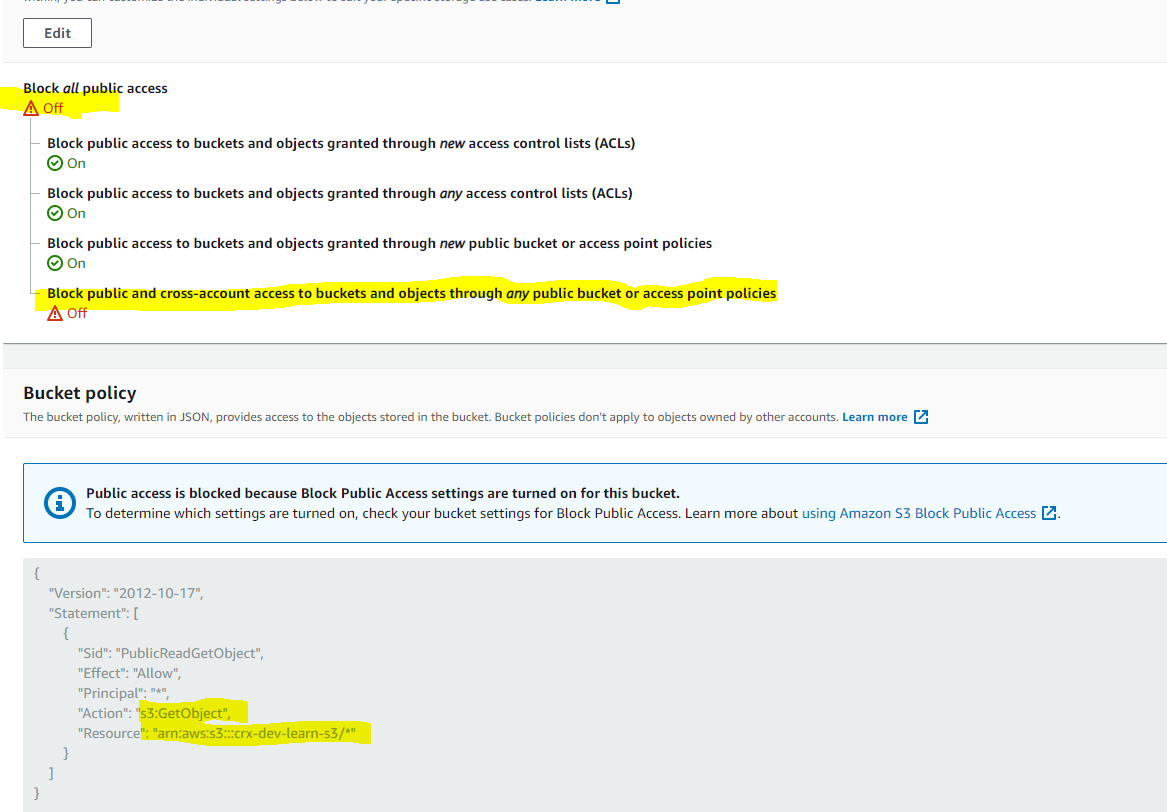
## Bucket Subresources

### Object Lifecycle

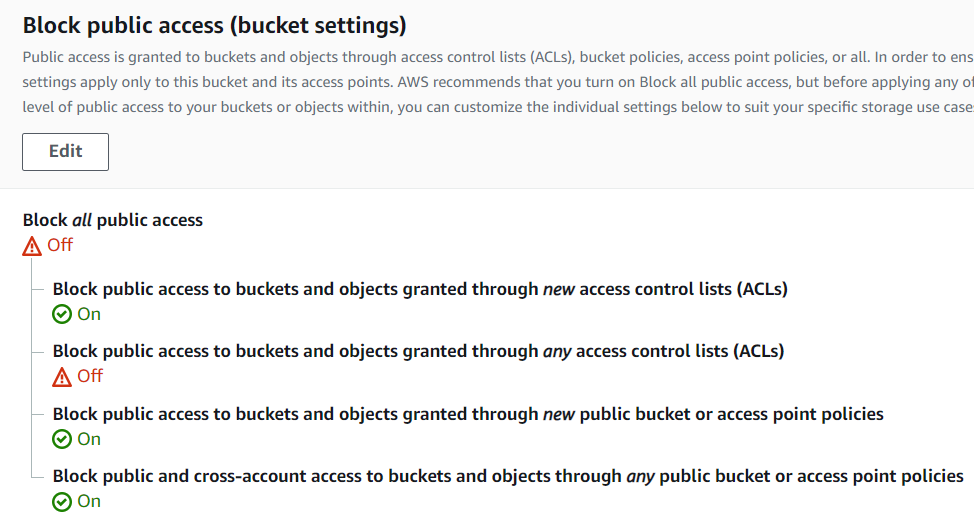
Refer Blog Post [S3 Object Lifecycle Management](https://jayendrapatil.com/aws-s3-object-lifecycle-management/)

### Static Website hosting

* S3 can be used for Static Website hosting with Client side scripts.
* S3 does not support server-side scripting
* S3, in conjunction with Route 53, supports hosting a website at the root domain which can point to the S3 website endpoint
* S3 website endpoints do not support HTTPS. i.e. (http://crx-dev-learn.s3-**website**-us-east-1.amazonaws.com/index.html)
* For S3 website hosting the content should be made **publicly readable** which can be provided using a bucket policy or an ACL on an object
  + - Bucket Policy:



* + - Object ACL & grant read access to static files individually:



* User can configure the index, error document as well as configure the conditional routing of on object name
* **Bucket policy applies only to objects owned by the bucket owner. If the bucket contains objects not owned by the bucket owner, then public READ permission on those objects should be granted using the object ACL.**
* Requester Pays buckets or DevPay buckets do not allow access through the website endpoint. Any request to such a bucket will receive a 403 -Access Denied response

### Versioning

Refer Blog Post [S3 Object Versioning](https://jayendrapatil.com/aws-s3-object-versioning/)

### Policy & Access Control List (ACL)

Refer Blog Post [S3 Permissions](https://jayendrapatil.com/aws-s3-permisions/)

### CORS (Cross Origin Resource Sharing)

* **Cross-origin resource sharing, or CORS, is a security feature of modern web browsers. It enables web browsers to negotiate which domains can make requests of external websites or services.**
* CORS is an important consideration when developing browser applications with the AWS SDK for JavaScript because most requests to resources are sent to an external domain, such as the endpoint for a web service.
* **All browsers implement the Same-Origin policy, for security reasons, where the web page from a domain can only request resources from the same domain.**
* **CORS allow client web applications loaded in one domain access to the restricted resources to be requested from another domain**
* **With CORS support, S3 allows cross-origin access to S3 resources**
* CORS configuration rules identify the origins allowed to access the bucket, the operations (HTTP methods) that would be supported for each origin, and other operation-specific information.

## How CORS Works

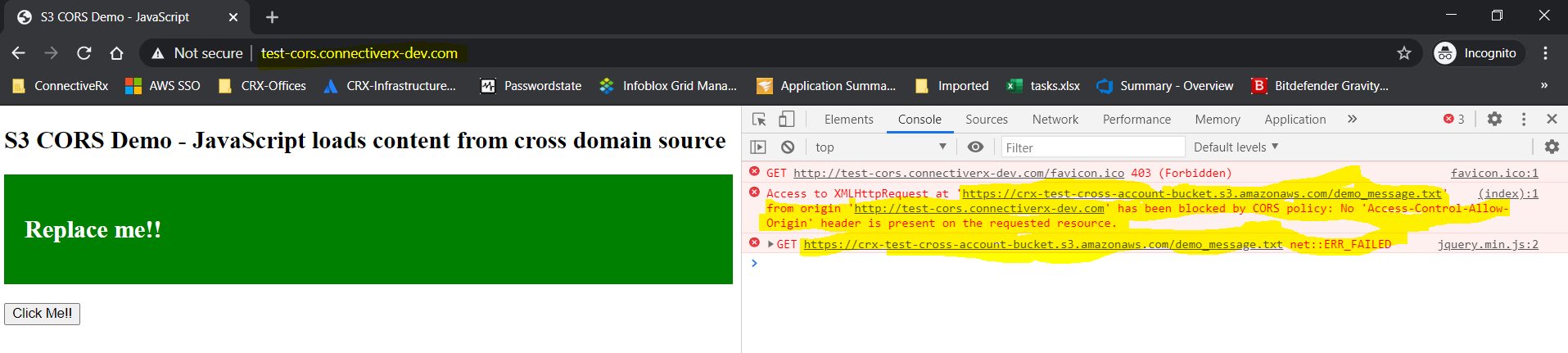
In the simplest case, your browser script makes a GET request for a resource from a server in another domain. Depending on the CORS configuration of that server, if the request is from a domain that's authorized to submit GET requests, the cross-origin server responds by returning the requested resource.

If either the requesting domain or the type of HTTP request is not authorized, the request is denied. However, CORS makes it possible to preflight the request before actually submitting it. In this case, a preflight request is made in which the OPTIONS access request operation is sent. If the cross-origin server's CORS configuration grants access to the requesting domain, the server sends back a preflight response that lists all the HTTP request types that the requesting domain can make on the requested resource.

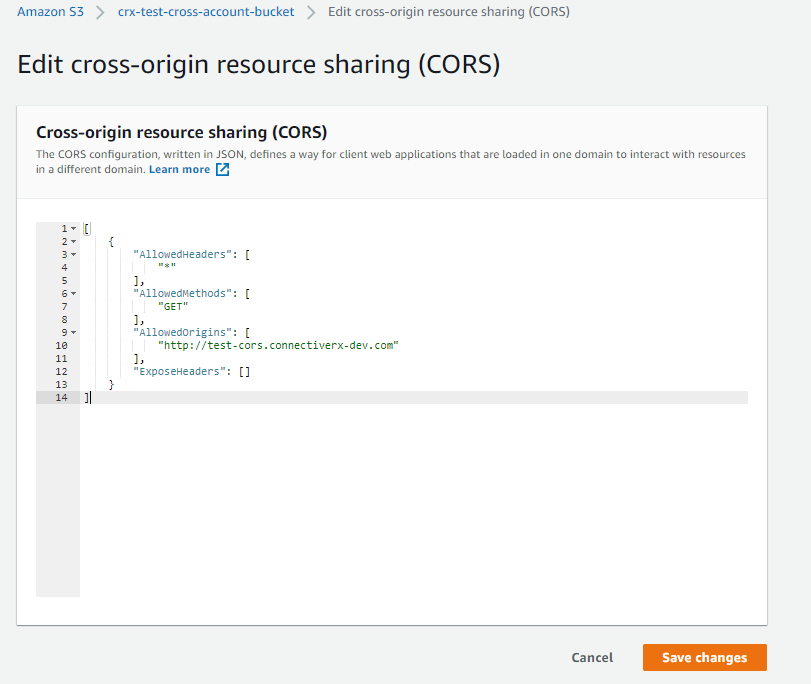

                    Process flow for CORS requests
                

**Demo** :

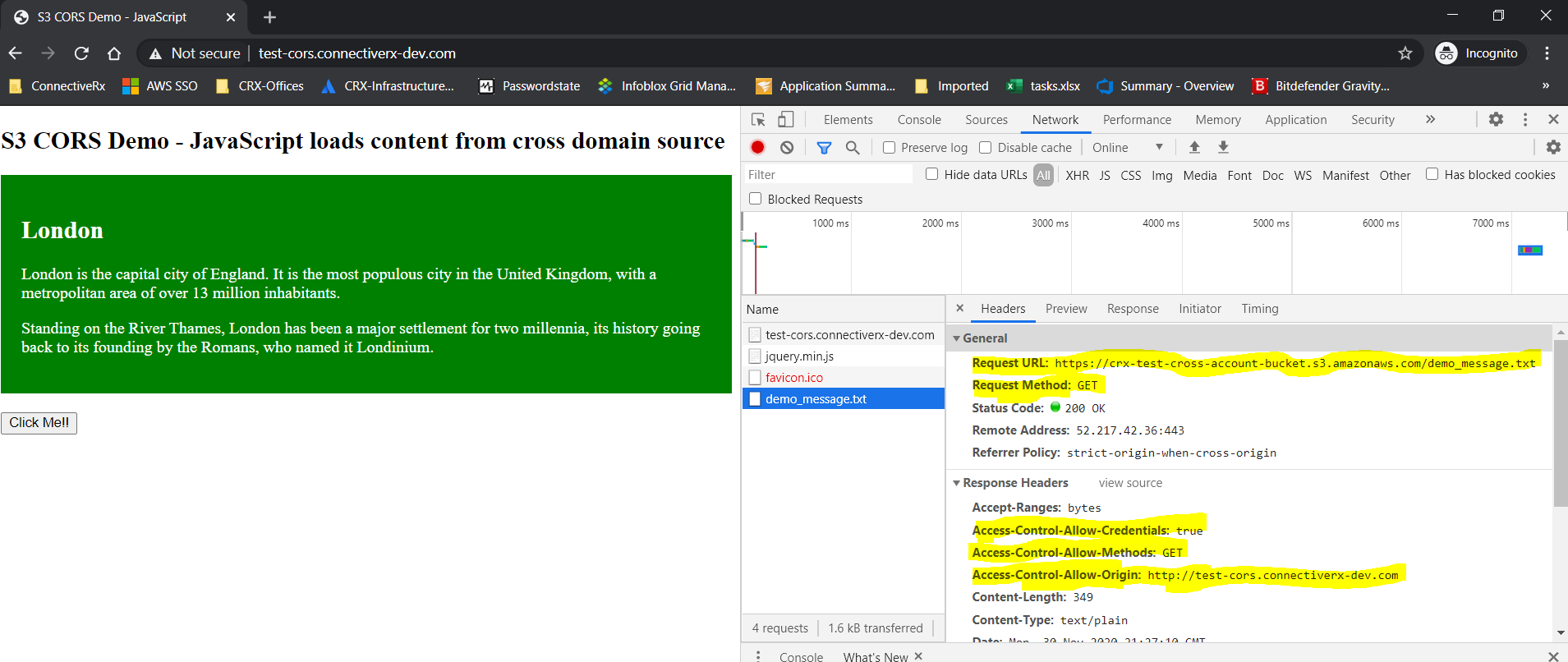
1. Website <http://test-cors.connectiverx-dev.com/> have javascript that tries to load page content from a s3 bucket (i.e. https://crx-test-cross-account-bucket.s3.amazonaws.com/demo\_message.txt).
2. By default since S3 is not configured to support CORS, It doesn’t sends AccessControl Headers (Access-Control-Allow-Origin & Access-Control-Allow-Method ) in response header hence browser returns CORS errors.



1. Update S3 CORS config as below to allow CORS.

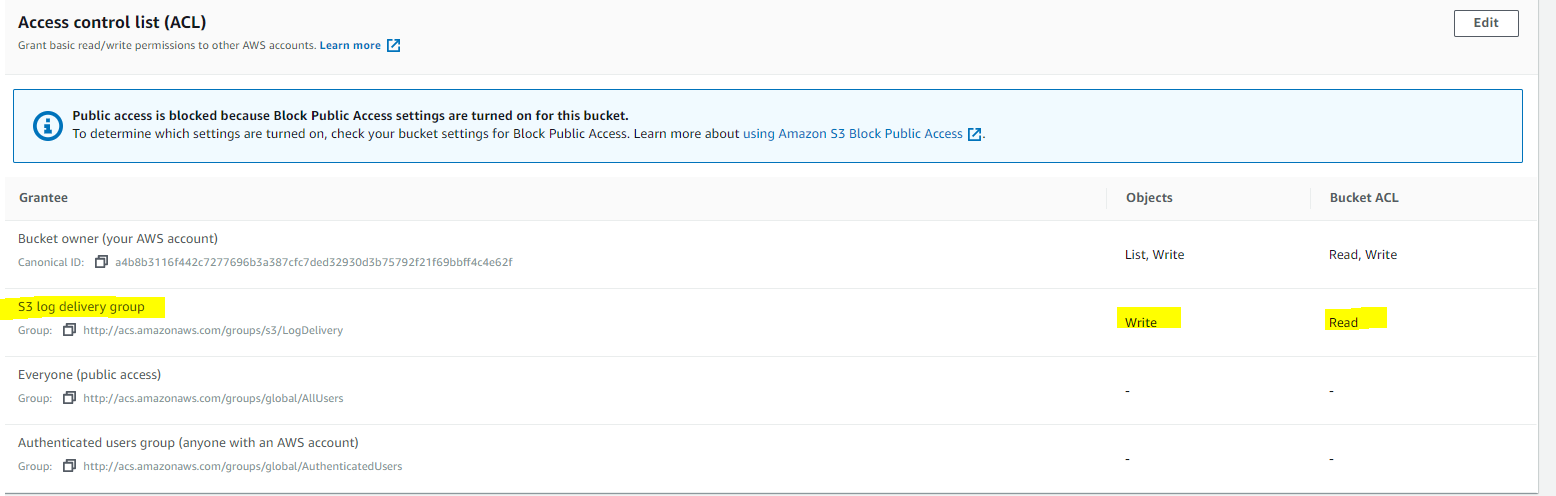


1. Once CORS config is updated, S3 sends AccessControl Headers (Access-Control-Allow-Origin & Access-Control-Allow-Method ) in response header which helps browser to proceed further & load content from other bucket.



### Logging – S3 Access Logs

* Logging, disabled by default, enables tracking access requests to S3 bucket
* Each access log record provides details about a single access request, such as the requester, bucket name, request time, request action, response status, and error code, if any.
* Access log information can be useful in security and access audits and also help learn about the customer base and understand the S3 bill
* **S3 periodically collects access log records, consolidates the records in log files, and then uploads log files to a target bucket as log objects.**
* If logging is enabled on multiple source buckets with same target bucket, the target bucket will have access logs for all those source buckets, but each log object will report access log records for a specific source bucket



### Tagging

* S3 provides the tagging subresource to store and manage tags on a bucket
* Cost allocation tags can be added to the bucket to categorize and track AWS costs
* AWS can generate a cost allocation report with usage and costs aggregated by the tags applied to the buckets

### Location

* When you create a bucket, AWS region needs to be specified where the S3 bucket will be created
* S3 stores this information in the location subresource and provides an API for retrieving this information

### Notification

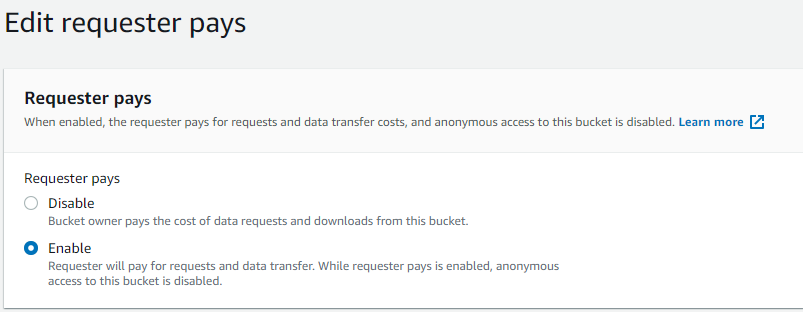
* S3 notification feature enables notifications to be triggered when certain events happen in your bucket
* Notifications are enabled at Bucket level
* Notifications can be configured to be filtered by the prefix and suffix of the key name of objects. However, filtering rules cannot be defined with  overlapping prefixes, overlapping suffixes, or prefix and suffix overlapping
* S3 can publish the following events
  + New Objects created event
    - Can be enabled for PUT, POST or COPY operations
    - You will not receive event notifications from failed operations
  + Object Removal event
    - Can public delete events for object deletion, version object deletion or insertion of delete marker
    - **You will not receive event notifications from automatic deletes from lifecycle policies or from failed operations.**
  + Reduced Redundancy Storage (RRS) object lost event
    - Can be used to reproduce/recreate the Object
* S3 can publish events to the following destination
  + **SNS topic**
  + **SQS queue**
  + **AWS Lambda**
* **For S3 to be able to publish events to the destination, S3 principal should be granted necessary permissions**

### Cross Region Replication

* Cross-region replication is a bucket-level feature that enables automatic, asynchronous copying of objects across buckets in different AWS regions
* S3 can replicate all or a subset of objects with specific key name prefixes
* S3 encrypts all data in transit across AWS regions using SSL
* Object replicas in the destination bucket are exact replicas of the objects in the source bucket with the same key names and the same metadata.
* Cross Region Replication can be useful for the following scenarios :-
  + **Compliance requirement** to have data backed up across regions
  + **Minimize latency** to allow users across geography to access objects
  + **Operational reasons** compute clusters in two different regions that analyze the same set of objects
* Requirements
  + **source and destination buckets must be versioning-enabled**
  + **source and destination buckets must be in different AWS regions**
  + objects can be replicated from a source bucket to only one destination bucket
  + S3 must have permission to replicate objects from that source bucket to the destination bucket on your behalf.
  + If the source bucket owner also owns the object, the bucket owner has full permissions to replicate the object. If not, the source bucket owner must have permission for the S3 actions s3:GetObjectVersion and s3:GetObjectVersionACL to read the object and object ACL
  + Setting up cross-region replication in a cross-account scenario (where the source and destination buckets are owned by different AWS accounts), the source bucket owner must have permission to replicate objects in the destination bucket.
* Replicated & Not Replicated
  + S3 **does not** retroactively replicate objects that existed before you added replication configuration. Only new objects created after you add a replication configuration **are** replicated
  + Only Objects created with **SSE-S3** **are** replicated using server-side encryption using the S3-managed encryption key.
  + Objects created with server-side encryption using either customer-provided (SSE-C) are replicated
  + Objects created with server-side encryption using AWS KMS–managed encryption (SSE-KMS) keys **are not replicated, by default. It requires**[**additional handling**](https://docs.aws.amazon.com/AmazonS3/latest/dev/crr-replication-config-for-kms-objects.html)**.**
  + S3 replicates **only** objects in the source bucket for which the bucket owner has permission to read objects and read ACLs
  + S3 **does not** replicate objects in the source bucket for which the bucket owner does not have permissions.
  + Any object ACL updates **are** replicated, although there can be some delay before Amazon S3 can bring the two in sync. This applies only to objects created after you add a replication configuration to the bucket.
  + Updates to bucket-level S3 subresources **are not** replicated, allowing different bucket configurations on the source and destination buckets
  + Only customer actions **are** replicated & actions performed by lifecycle configuration **are not** replicated
  + Objects in the source bucket that are replicas, created by another cross-region replication, **are not** replicated.

### Requester Pays

* By default, buckets are owned by the AWS account that created it (the bucket owner) and the AWS account pays for storage costs, downloads and data transfer charges associated with the bucket.
* Using Requester Pays subresource :-
  + Bucket owner specifies that the requester requesting the download will be charged for the download
  + However, the bucket owner still pays the storage costs
* Enabling Requester Pays on a bucket
  + disables anonymous access to that bucket
  + does not support BitTorrent
  + does not support SOAP requests
  + cannot be enabled for end user logging bucket



### Torrent

* Default distribution mechanism for S3 data is via client/server download
* Bucket owner bears the cost of Storage as well as the request and transfer charges which can increase linearly for an popular object
* S3 also supports the BitTorrent protocol
  + BitTorrent is an open source Internet distribution protocol
  + BitTorrent addresses this problem by recruiting the very clients that are downloading the object as distributors themselves
  + S3 bandwidth rates are inexpensive, but BitTorrent allows developers to further save on bandwidth costs for a popular piece of data by letting users download from Amazon and other users simultaneously
* Benefit for publisher is that for large, popular files the amount of data actually supplied by S3 can be substantially lower than what it would have been serving the same clients via client/server download
* Any object in S3 that is publicly available and can be read anonymously can be downloaded via BitTorrent
* torrent file can be retrieved for any publicly available object by simply adding a “?torrent” query string parameter at the end of the REST GET request for the object
* Generating the .torrent for an object takes time proportional to the size of that object, so its recommended to make a first torrent request yourself to generate the file so that subsequent requests are faster
* Torrent are enabled only for objects that are less than 5 GB in size.
* Torrent subresource can only be retrieve, and cannot be created, updated or deleted

## AWS Certification Exam Practice Questions

1. An organization’s security policy requires multiple copies of all critical data to be replicated across at least a primary and backup data center. The organization has decided to store some critical data on Amazon S3. Which option should you implement to ensure this requirement is met?
   1. Use the S3 copy API to replicate data between two S3 buckets in different regions
   2. You do not need to implement anything since S3 data is automatically replicated between regions
   3. Use the S3 copy API to replicate data between two S3 buckets in different facilities within an AWS Region
   4. **You do not need to implement anything since S3 data is automatically replicated between multiple facilities within an AWS Region**
2. A customer wants to track access to their Amazon Simple Storage Service (S3) buckets and also use this information for their internal security and access audits. Which of the following will meet the Customer requirement?
   1. Enable AWS CloudTrail to audit all Amazon S3 bucket access.
   2. **Enable server access logging for all required Amazon S3 buckets**
   3. Enable the Requester Pays option to track access via AWS Billing
   4. Enable Amazon S3 event notifications for Put and Post.
3. A user is enabling a static website hosting on an S3 bucket. Which of the below mentioned parameters cannot be configured by the user?
   1. Error document
   2. **Conditional error on object name**
   3. Index document
   4. Conditional redirection on object name
4. Company ABCD is running their corporate website on Amazon S3 accessed from http//www.companyabcd.com. Their marketing team has published new web fonts to a separate S3 bucket accessed by the S3 endpoint: https://s3-us-west1.amazonaws.com/abcdfonts. While testing the new web fonts, Company ABCD recognized the web fonts are being blocked by the browser. What should Company ABCD do to prevent the web fonts from being blocked by the browser?
   1. Enable versioning on the abcdfonts bucket for each web font
   2. Create a policy on the abcdfonts bucket to enable access to everyone
   3. Add the Content-MD5 header to the request for webfonts in the abcdfonts bucket from the website
   4. **Configure the abcdfonts bucket to allow cross-origin requests by creating a CORS configuration**
5. Company ABCD is currently hosting their corporate site in an Amazon S3 bucket with Static Website Hosting enabled. Currently, when visitors go to http://www.companyabcd.com the index.html page is returned. Company C now would like a new page welcome.html to be returned when a visitor enters http://www.companyabcd.com in the browser. Which of the following steps will allow Company ABCD to meet this requirement? Choose 2 answers.
   1. **Upload an html page named welcome.html to their S3 bucket**
   2. Create a welcome subfolder in their S3 bucket
   3. **Set the Index Document property to welcome.html**
   4. Move the index.html page to a welcome subfolder
   5. Set the Error Document property to welcome.html