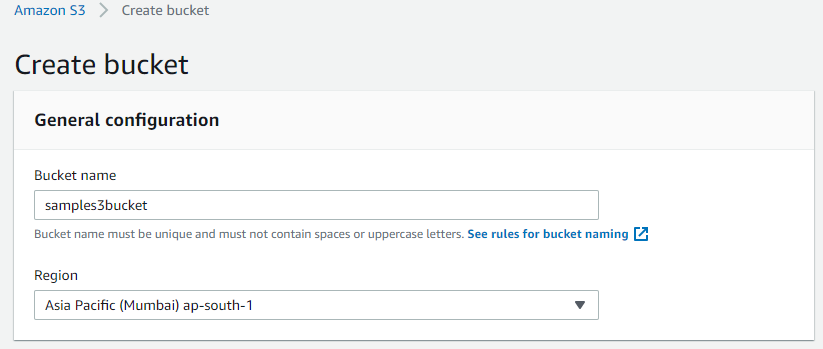
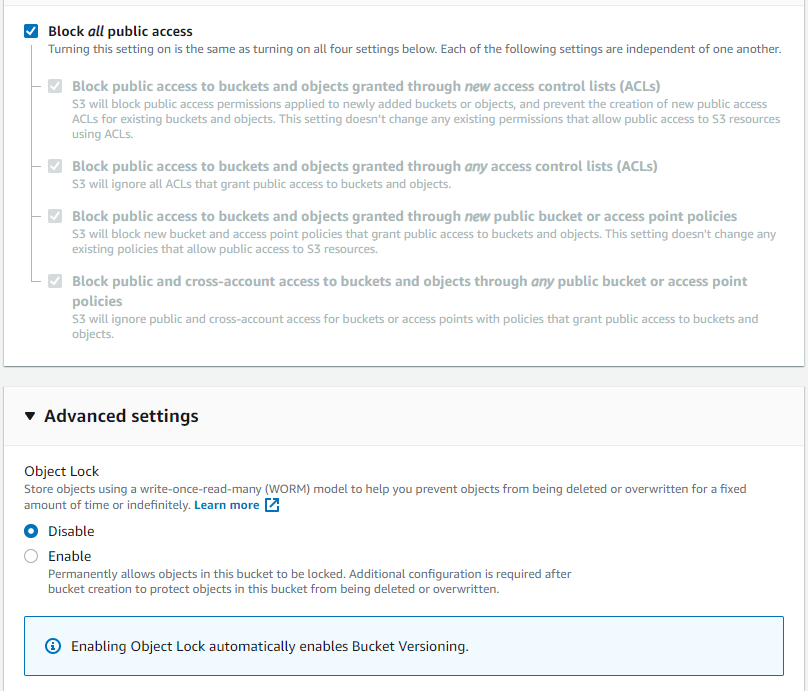
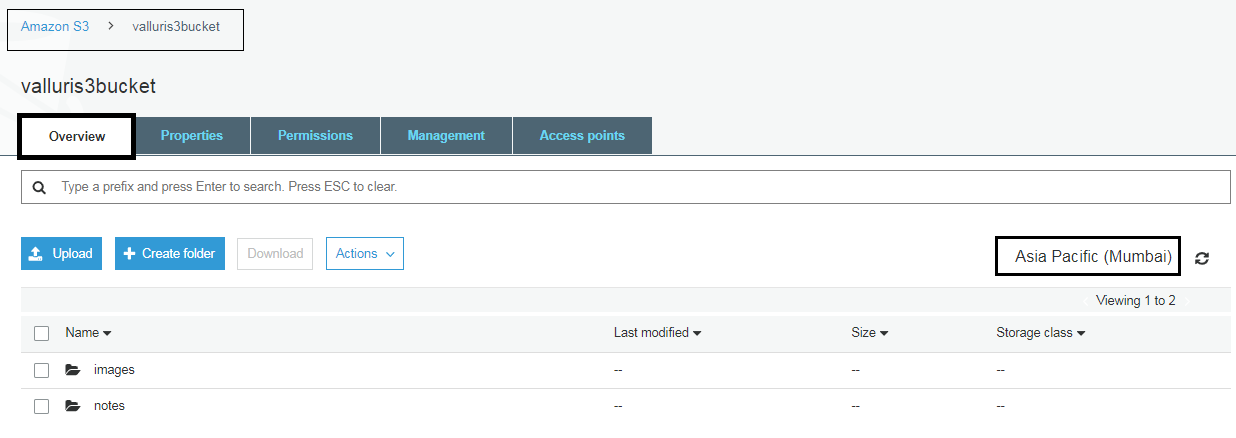
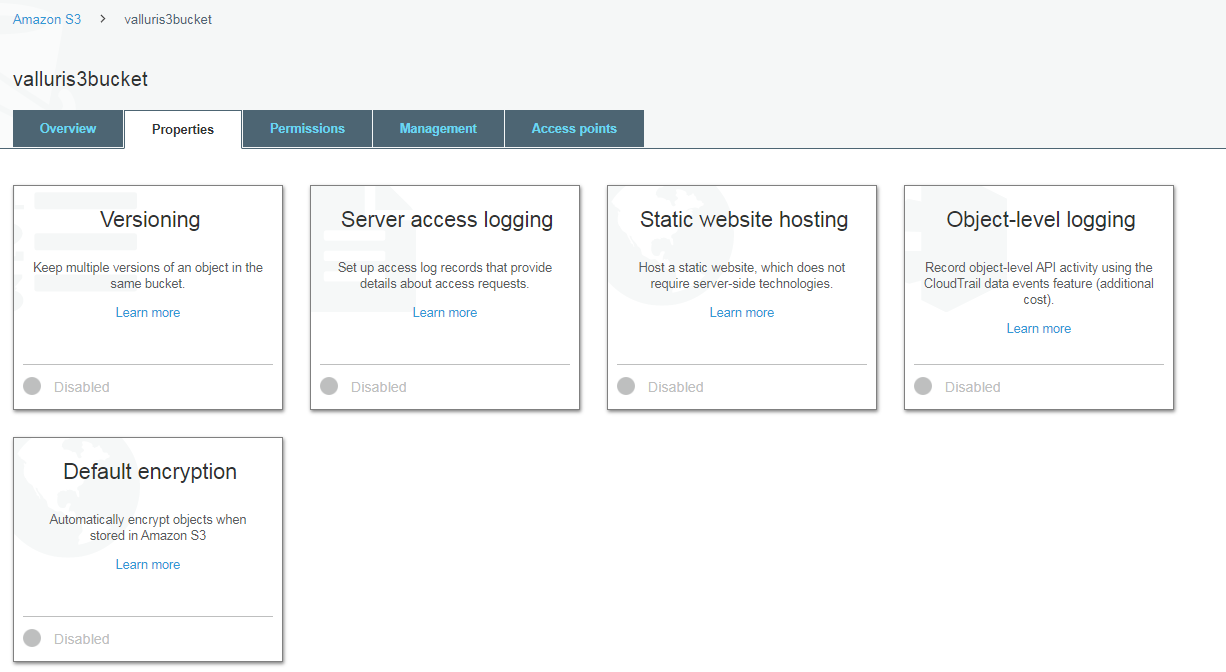
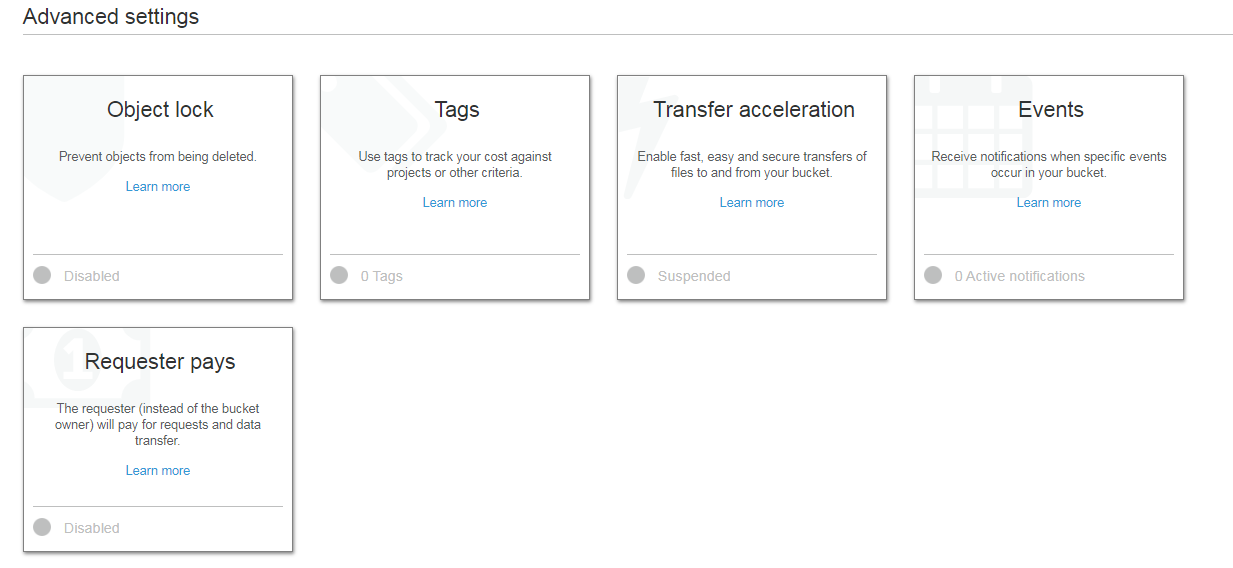
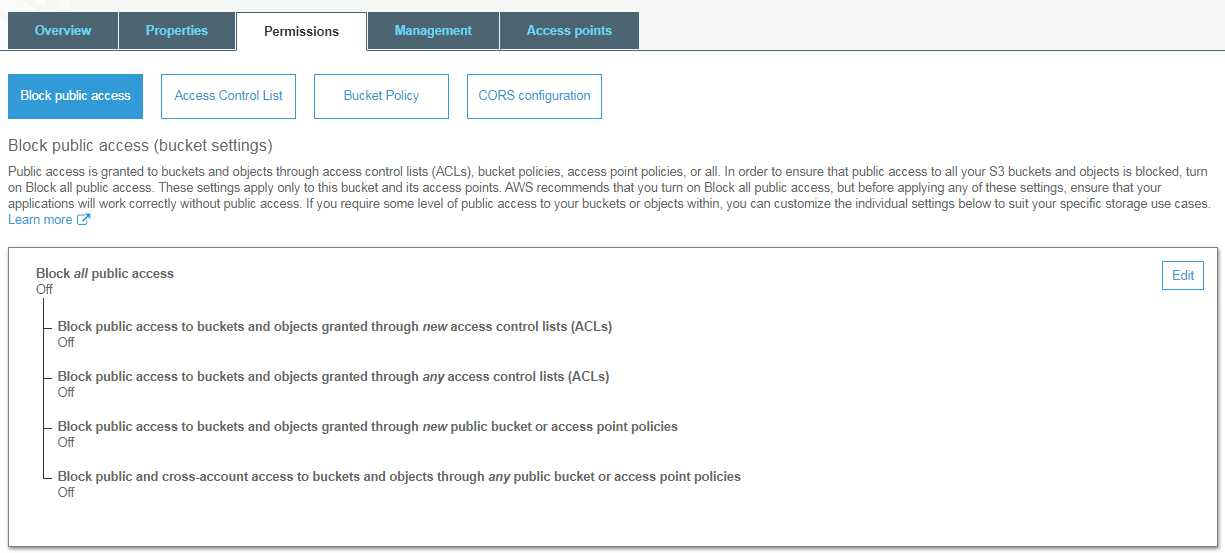
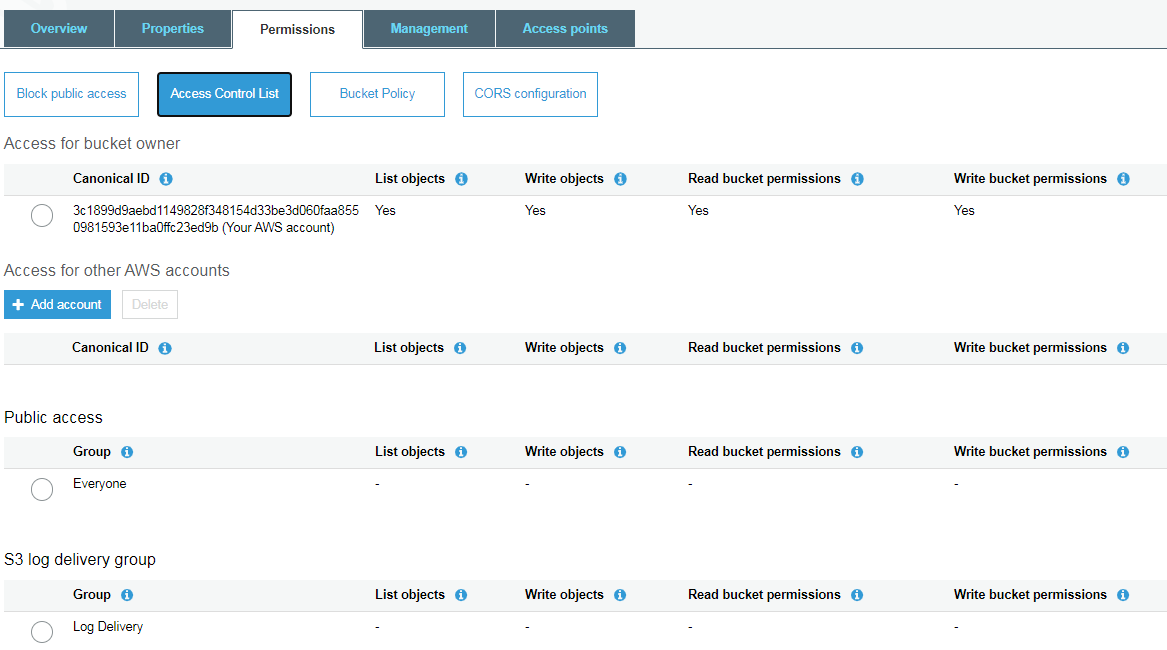
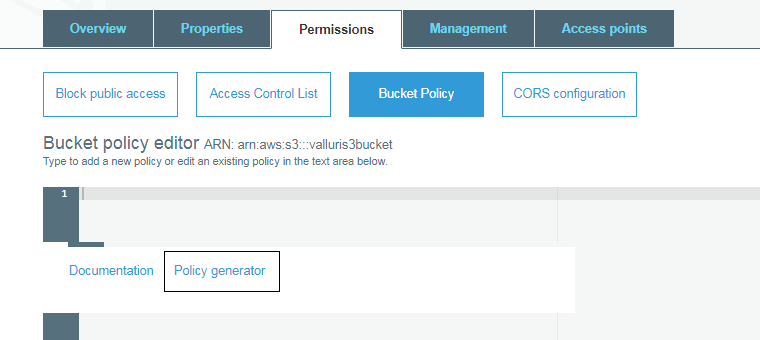
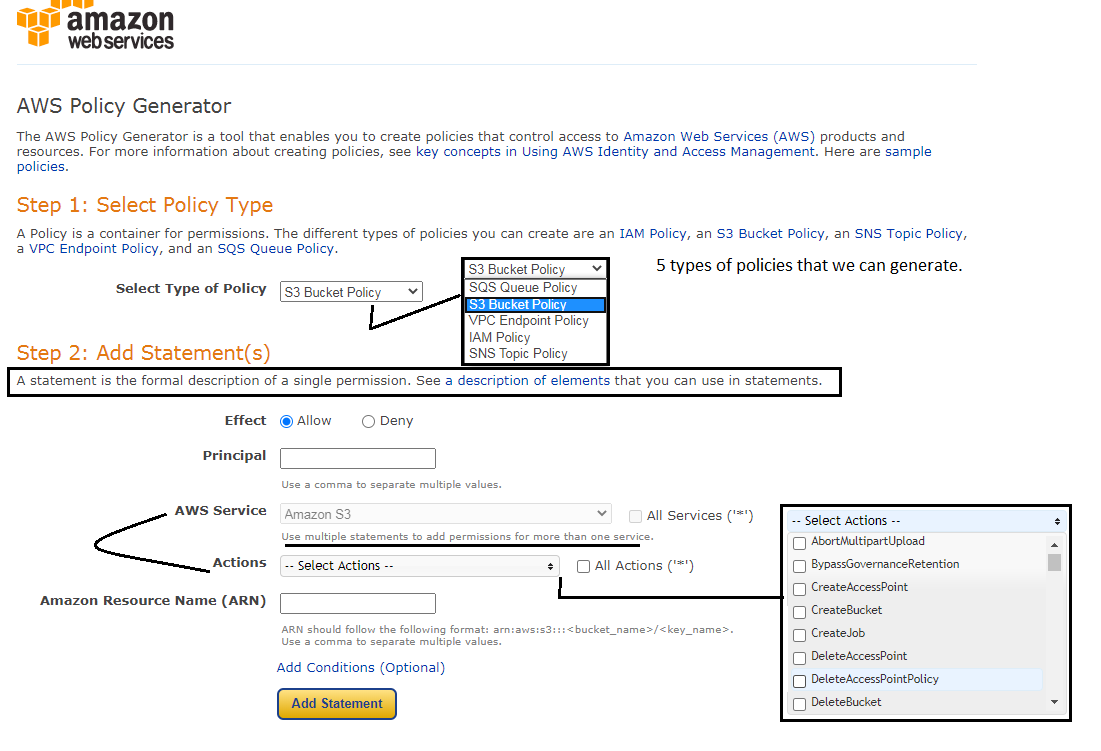
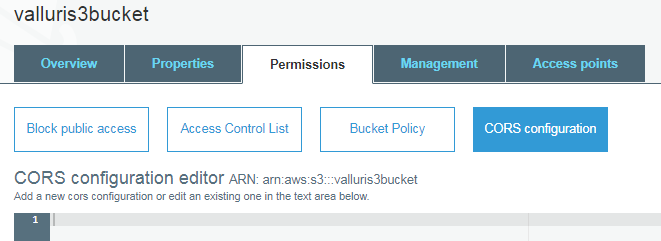
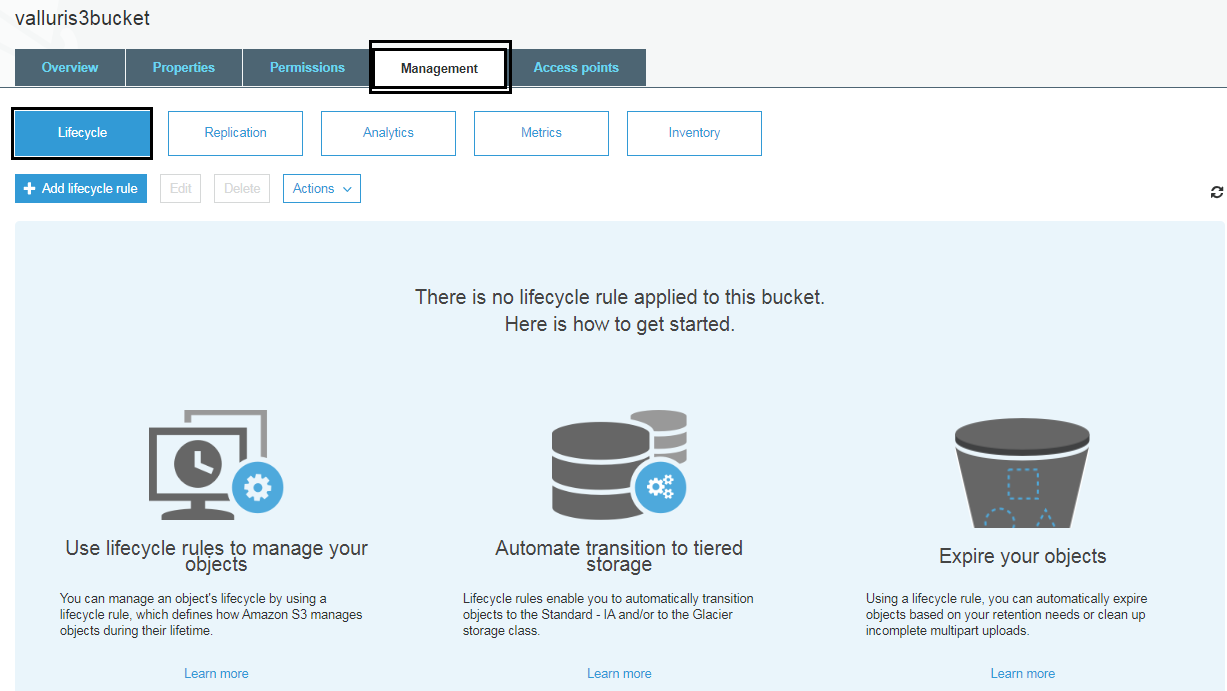
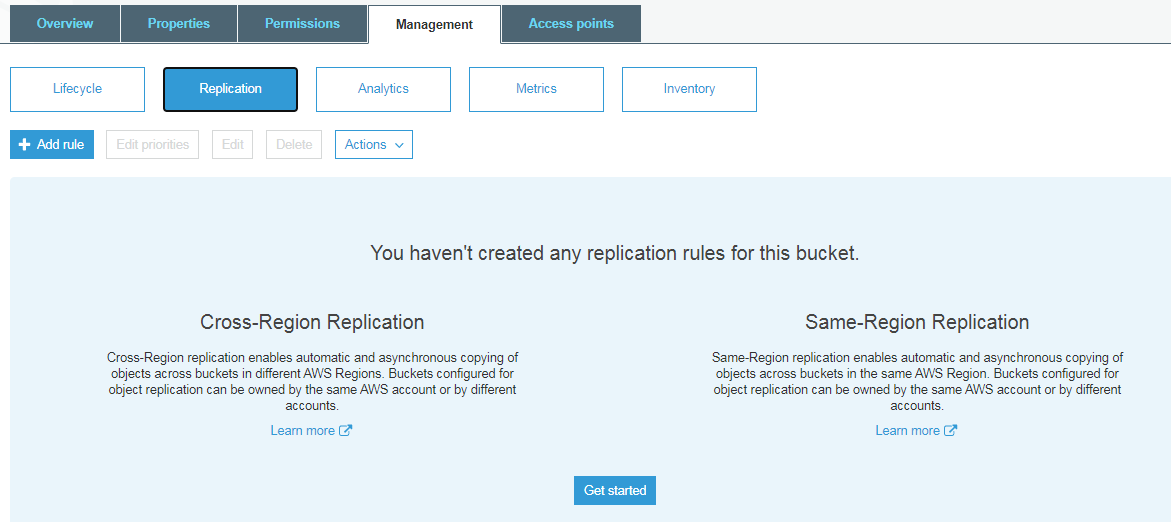
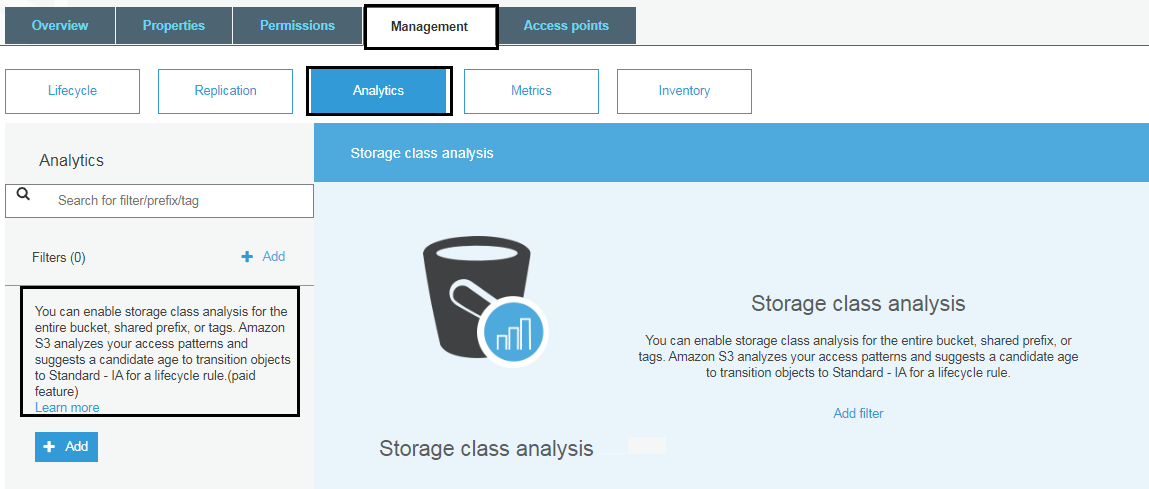
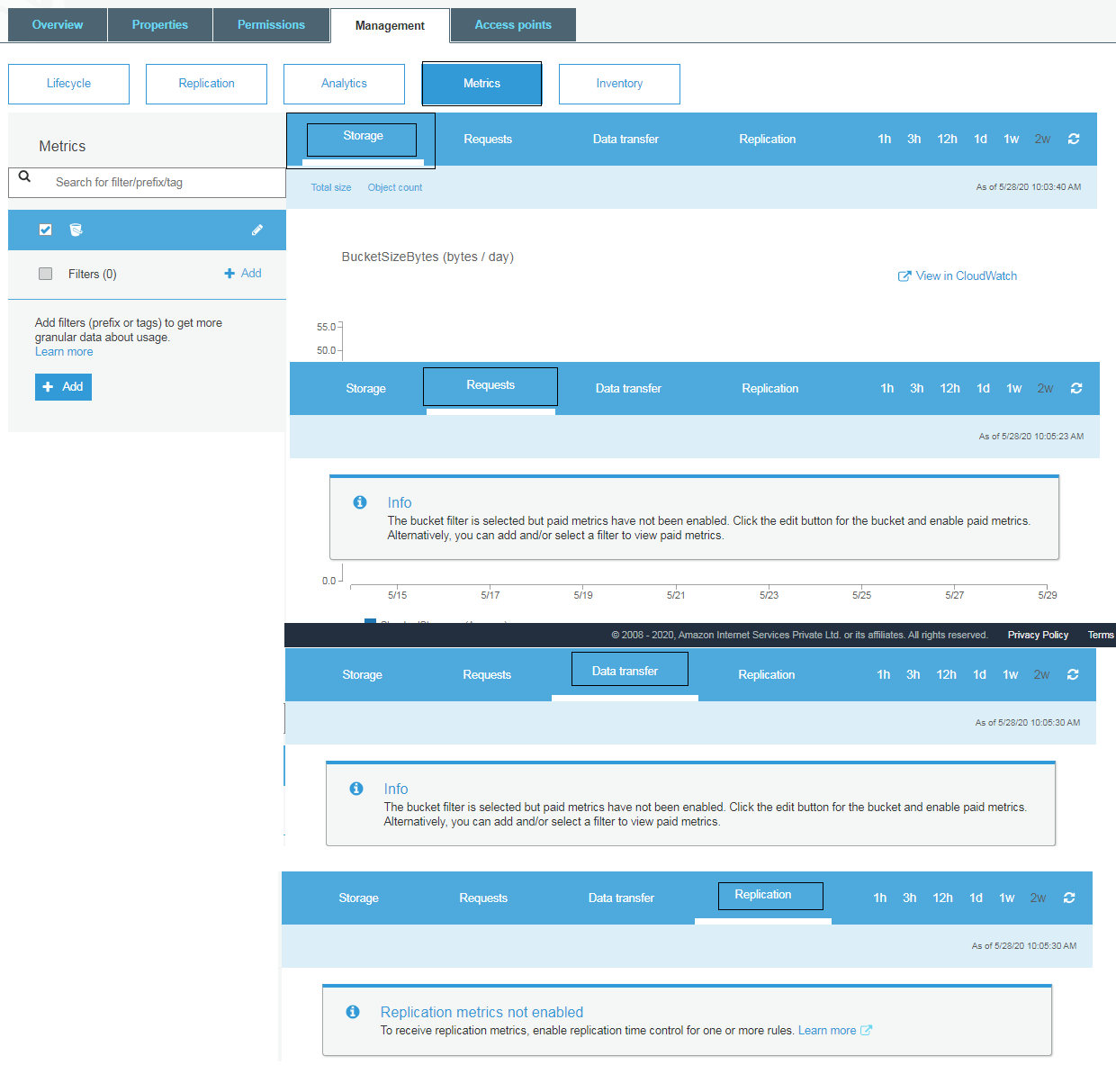
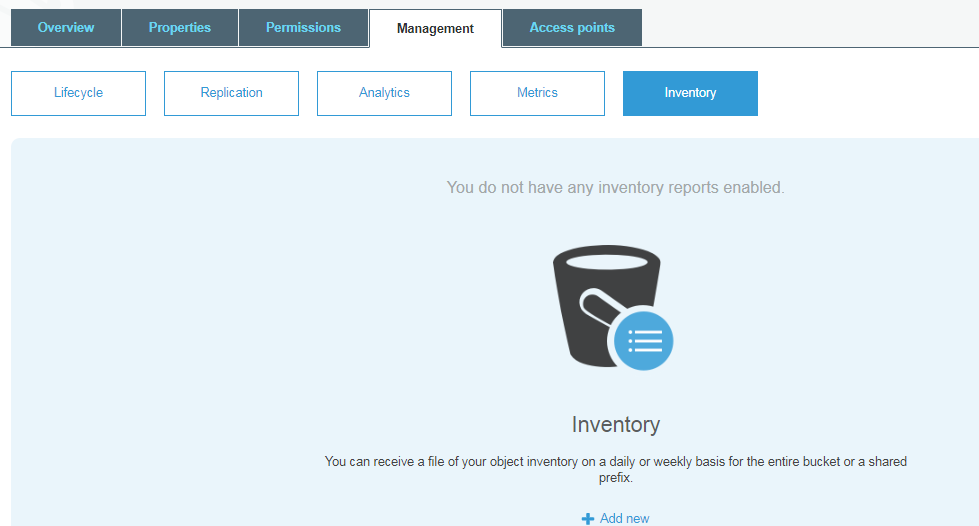
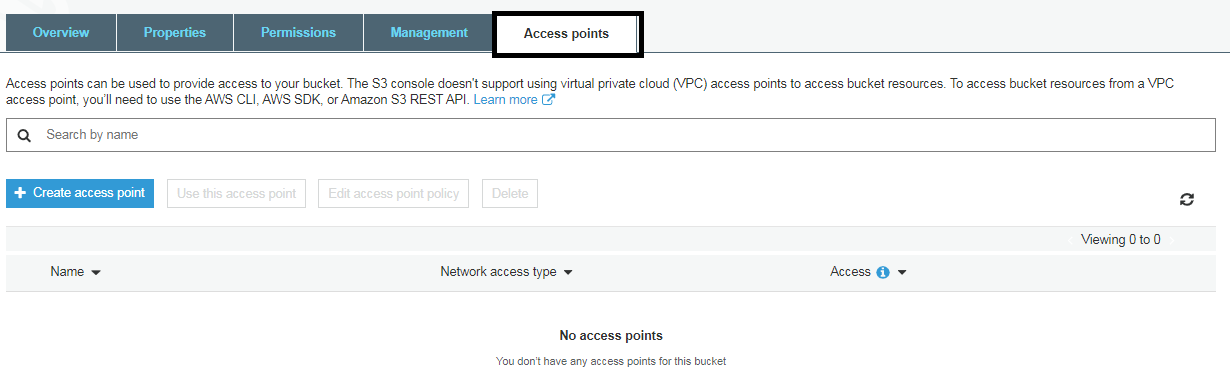
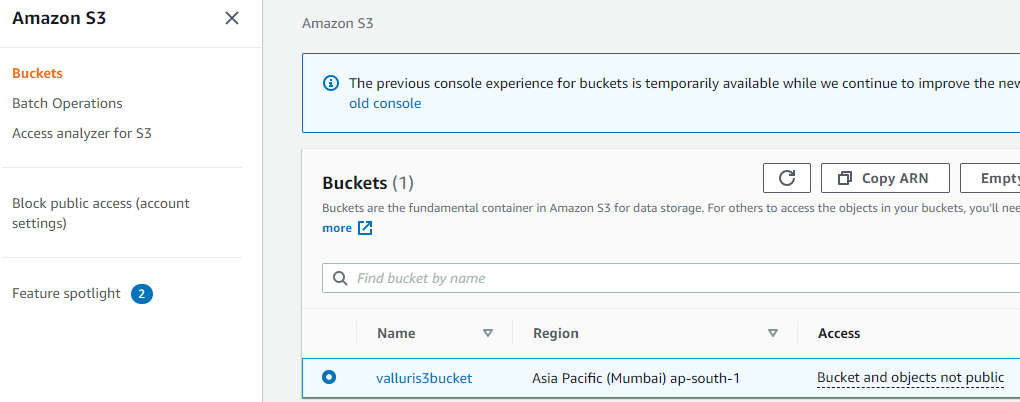
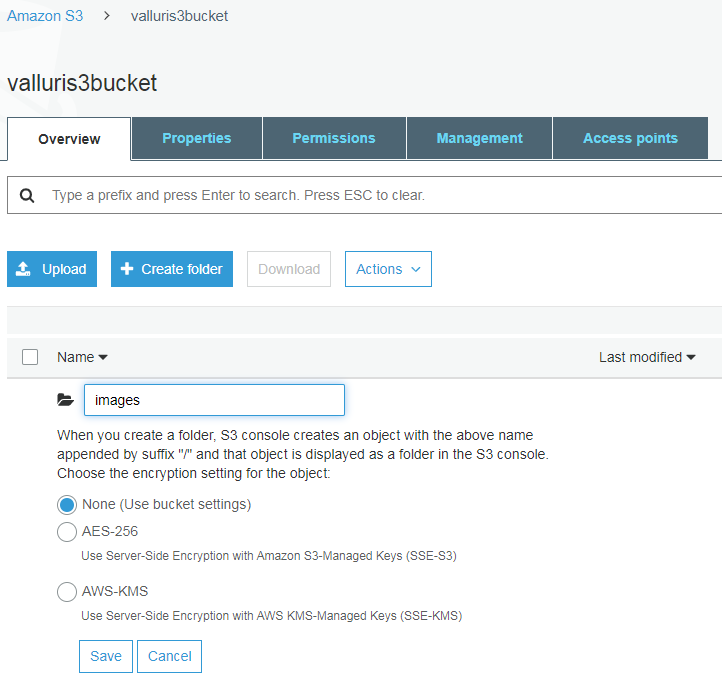
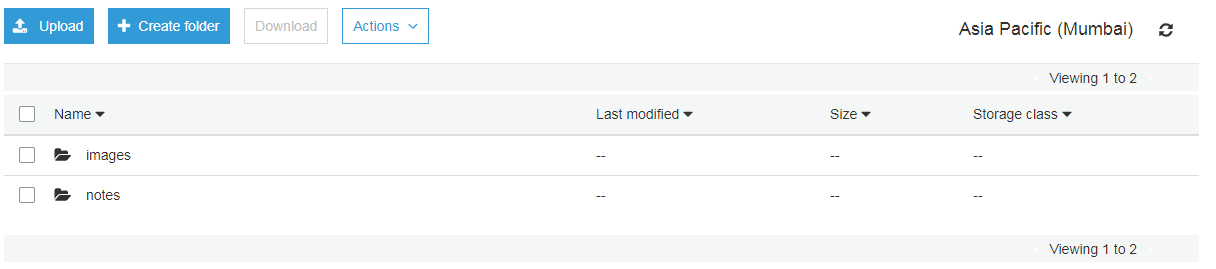
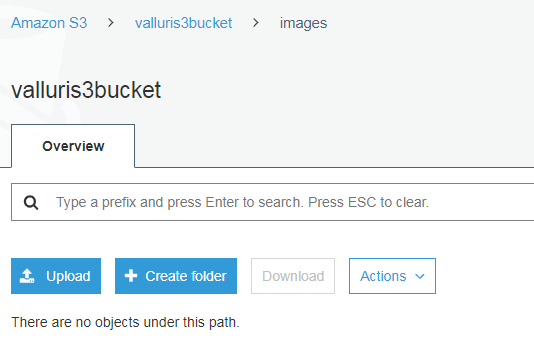
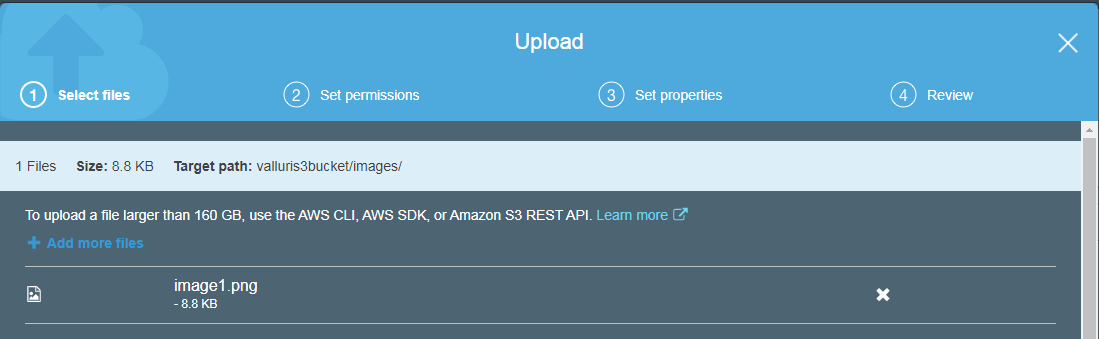
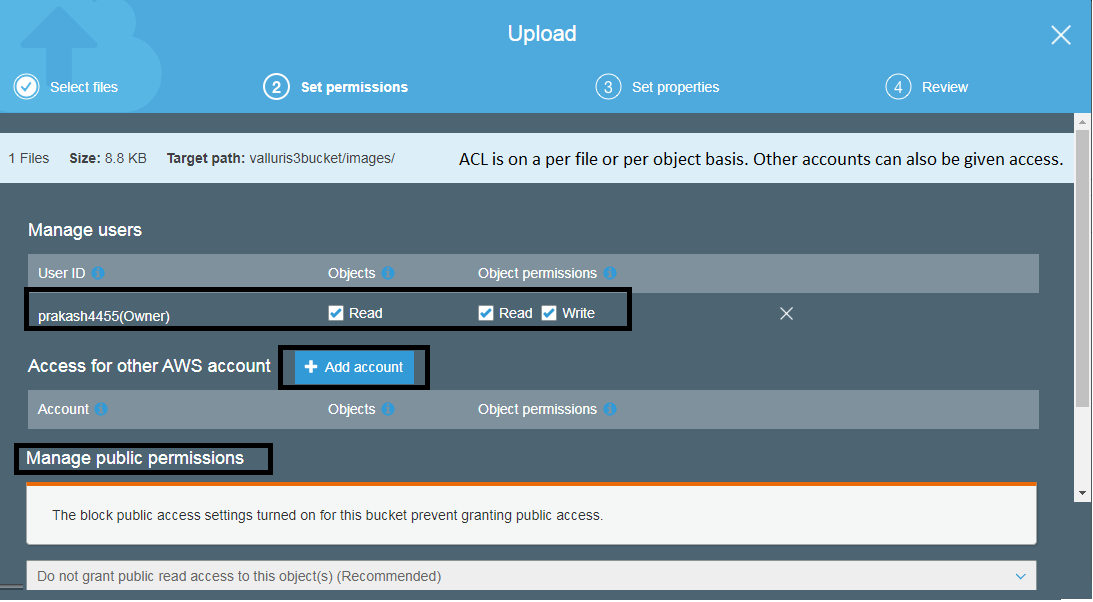
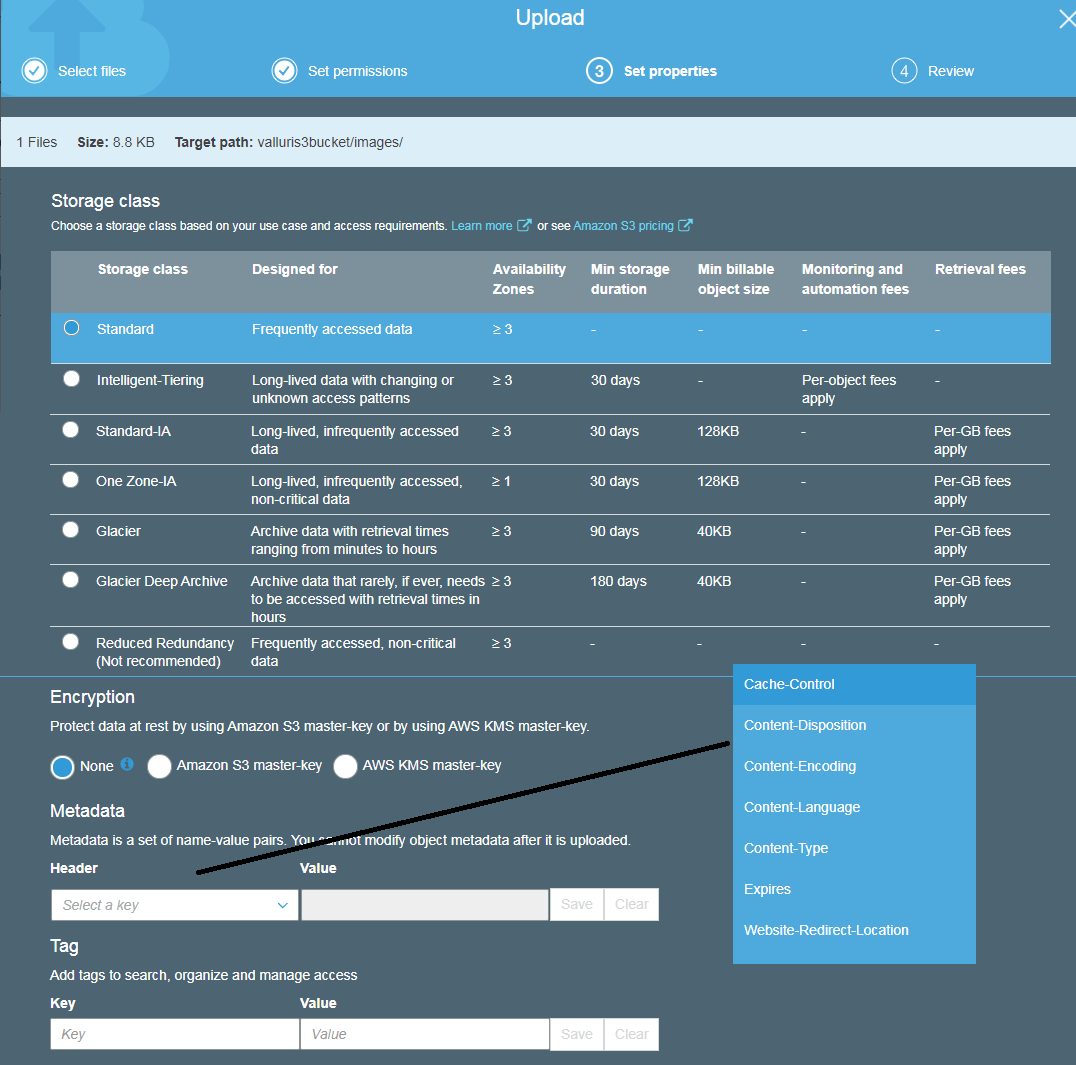
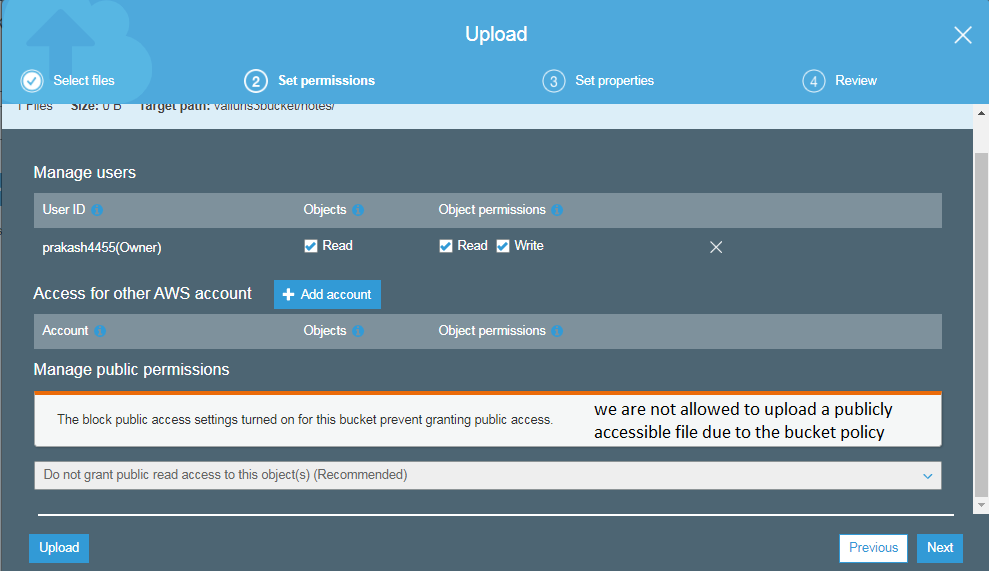
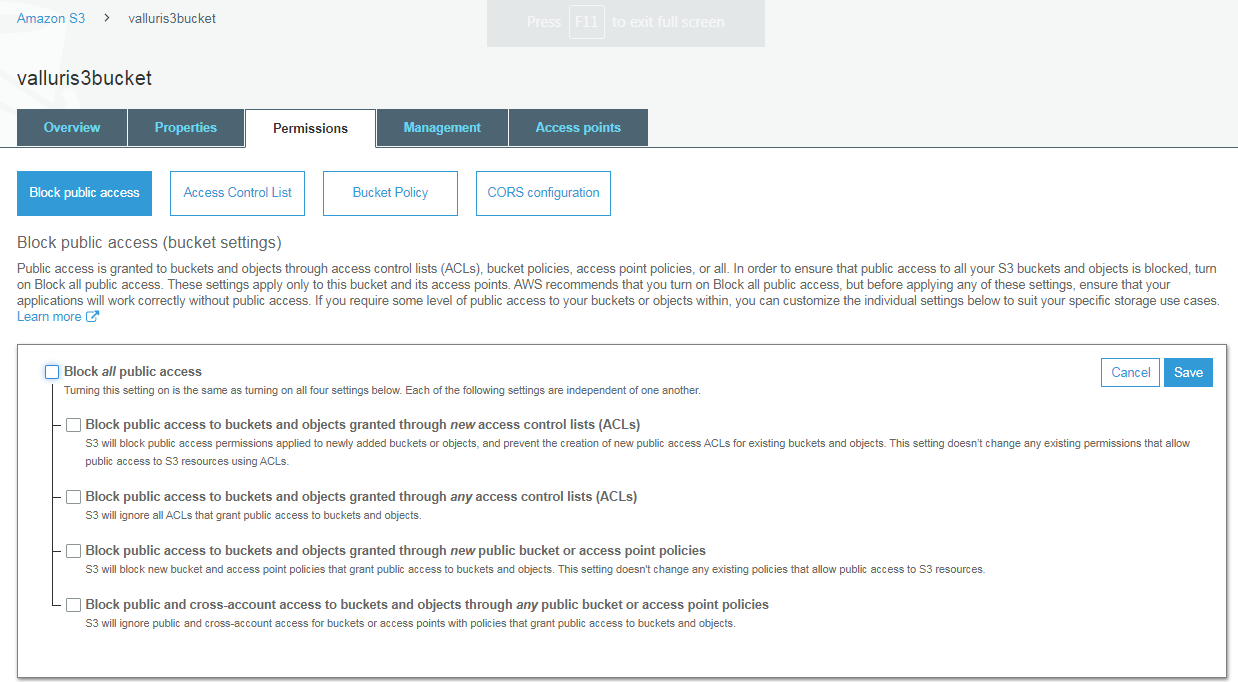
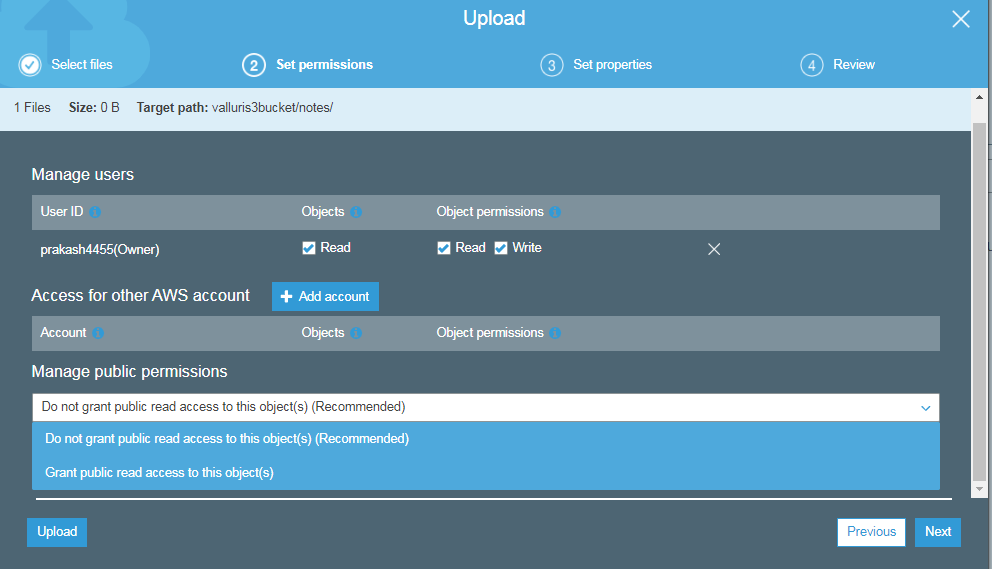
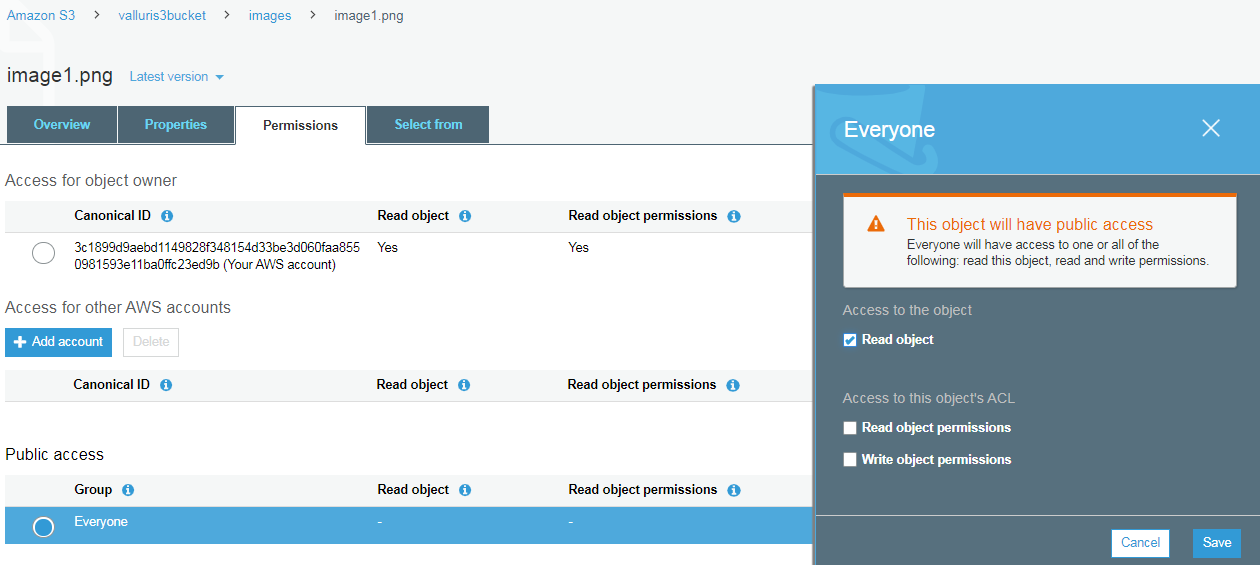
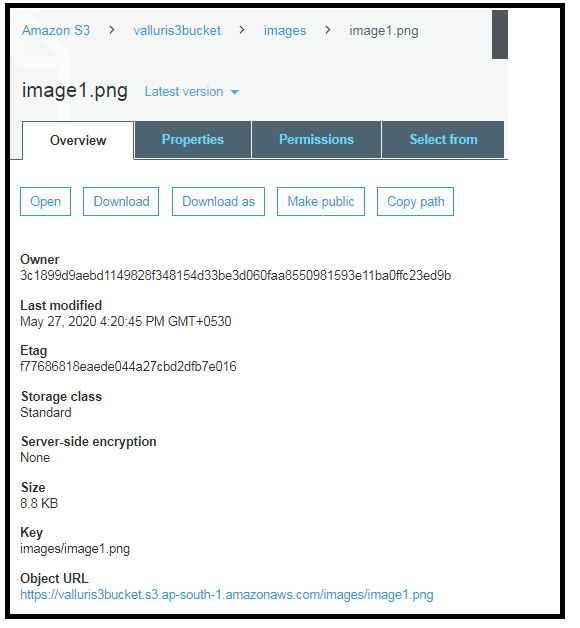
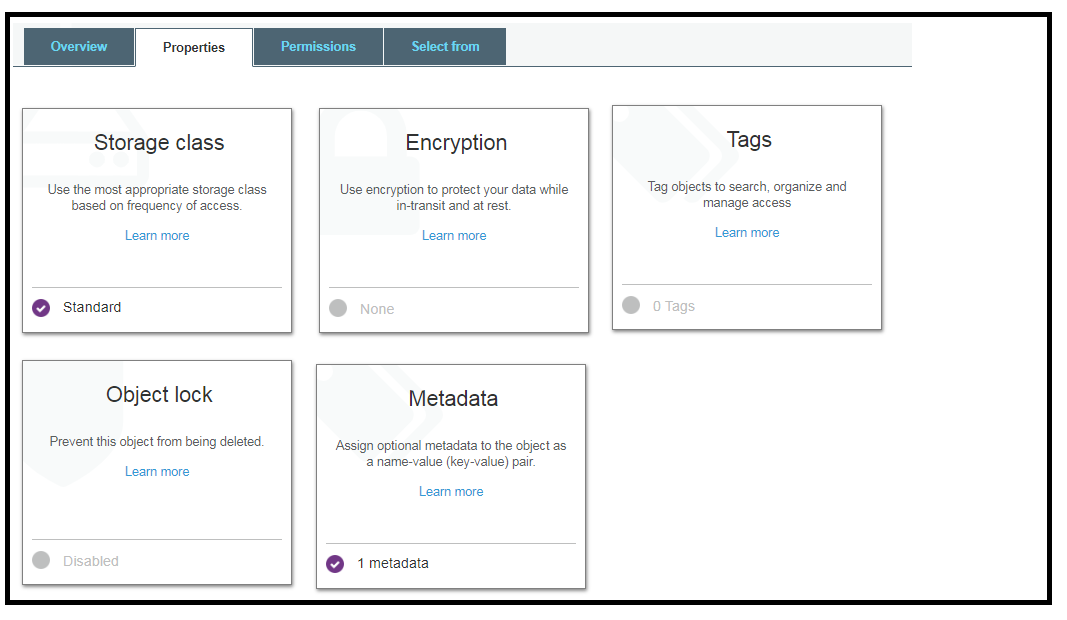
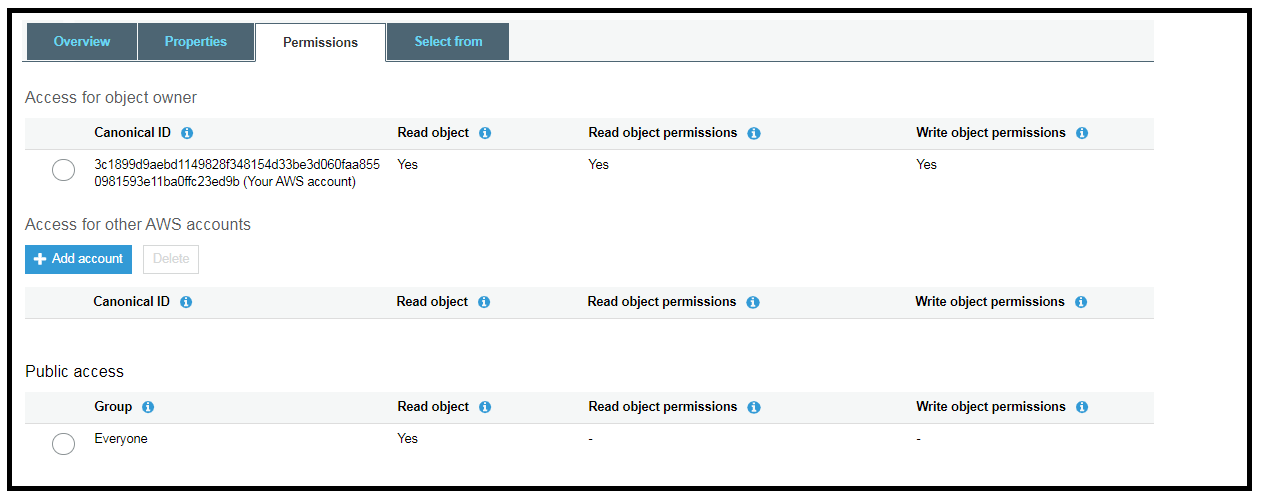
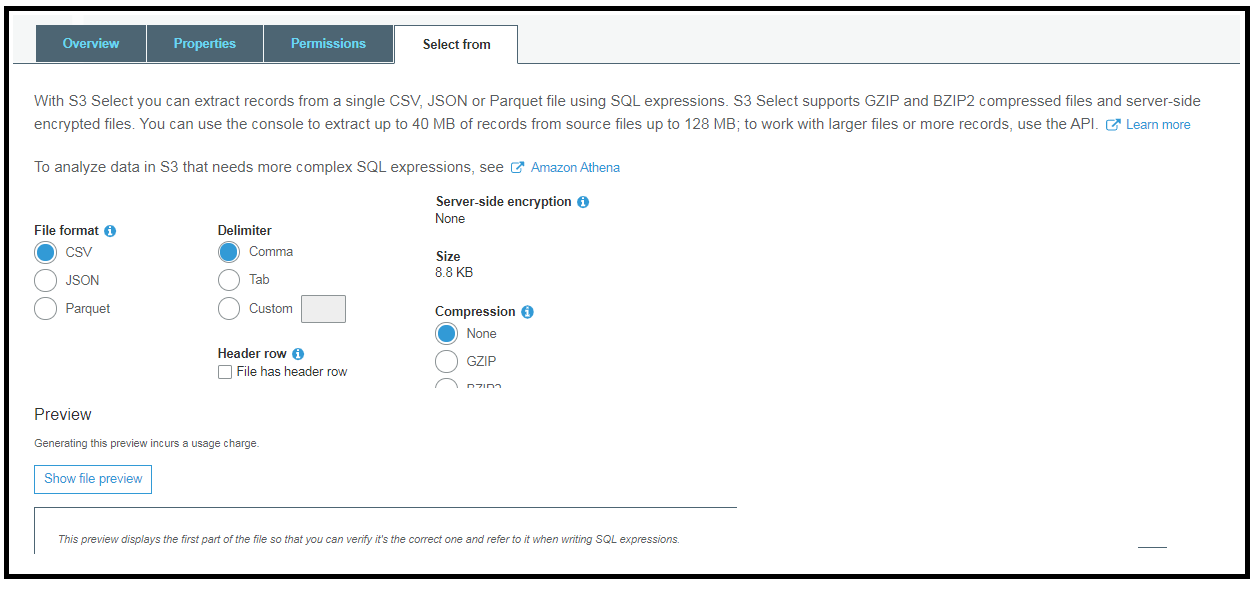
1 . Simple storage service – S3  
- This service is not specific to region and is global.  
- Unlimited storage + it’s a key-value store.  
- it provides a secure,durable, highly scalable object storage(not block storage).  
- Used for files(0 to 5tb), images, webpages etc. NOT for OS and DB’s.  
- You can store any retreive any amount of data from anywhere on the web.  
- The data is spread across multiple devices and multiple facilities.  
- i.e there is a high element of high availability / disaster recovery – which is already built into aws service. i.e if s3 service will be still available even if amazon looses one of their devices or one of their facilities.  
- A faulty device / data centre/ availability zone/ : you will still be able to access your files.  
- Bucket is similar to folder, and its name must be unique *globally*.  
- Upon upload successful : we get a HTTP200 code.  
  
DATA CONSISTENCY MODEL FOR S3

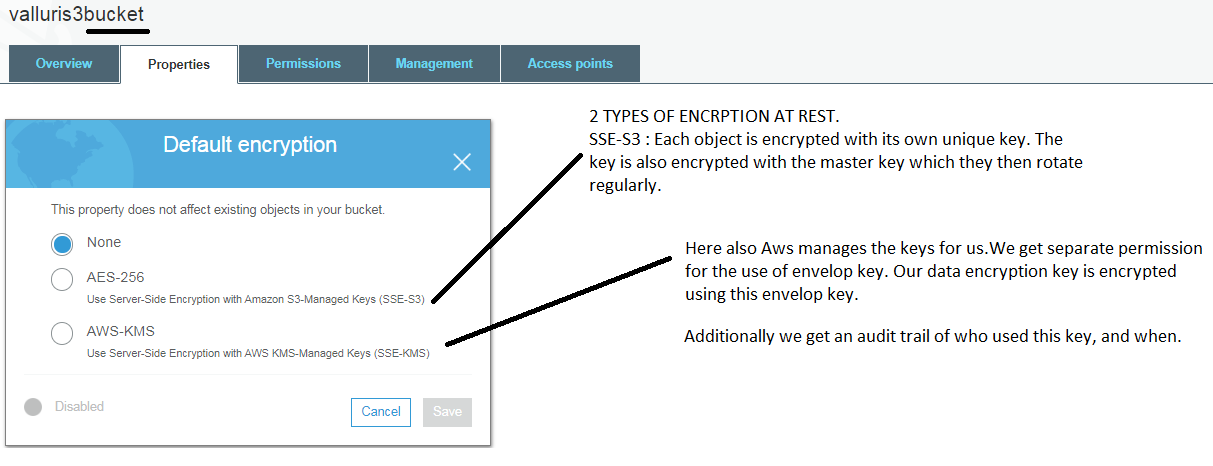
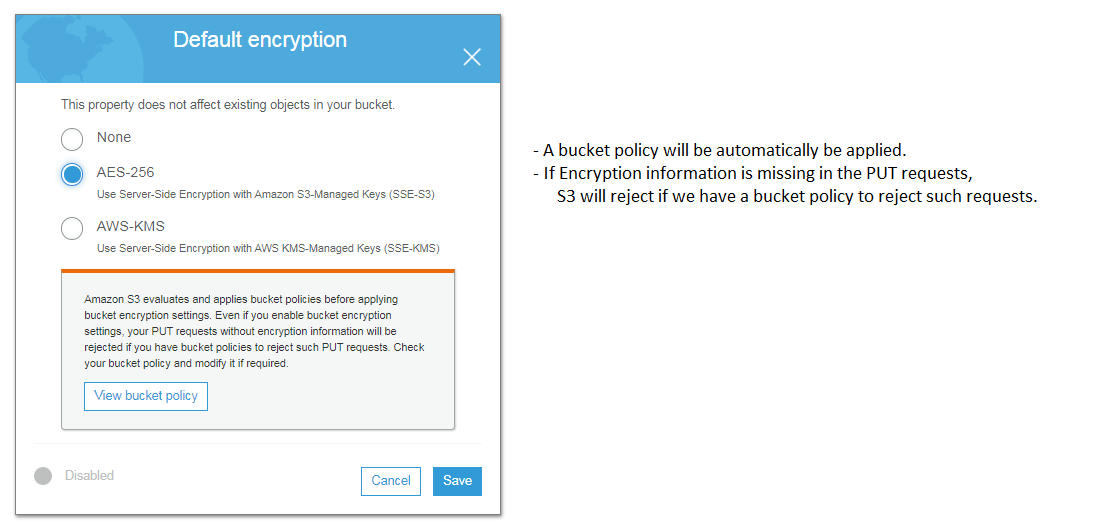
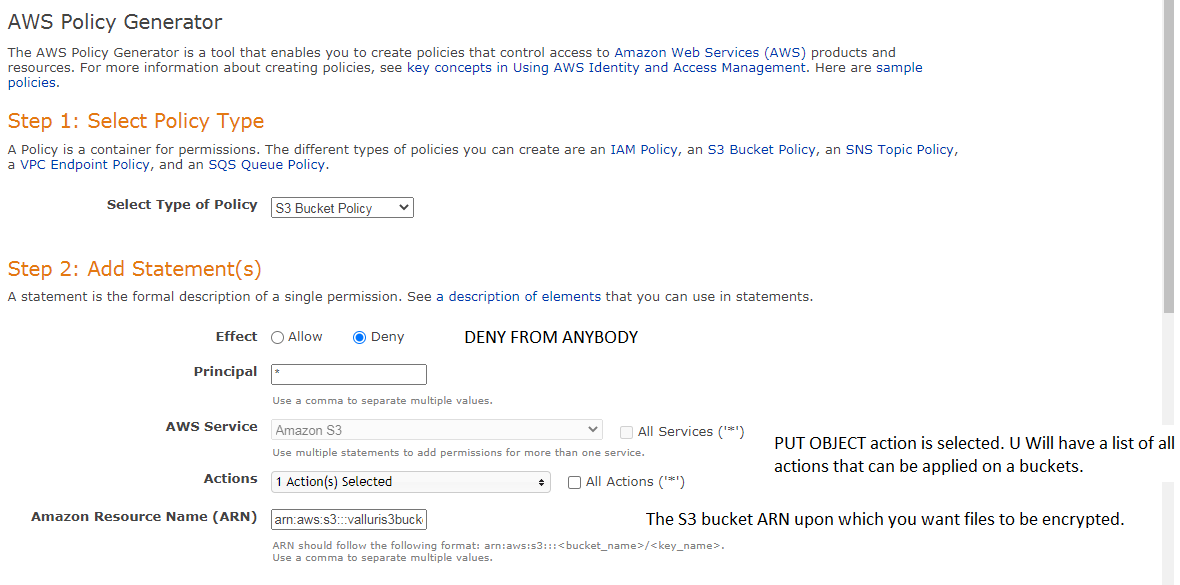
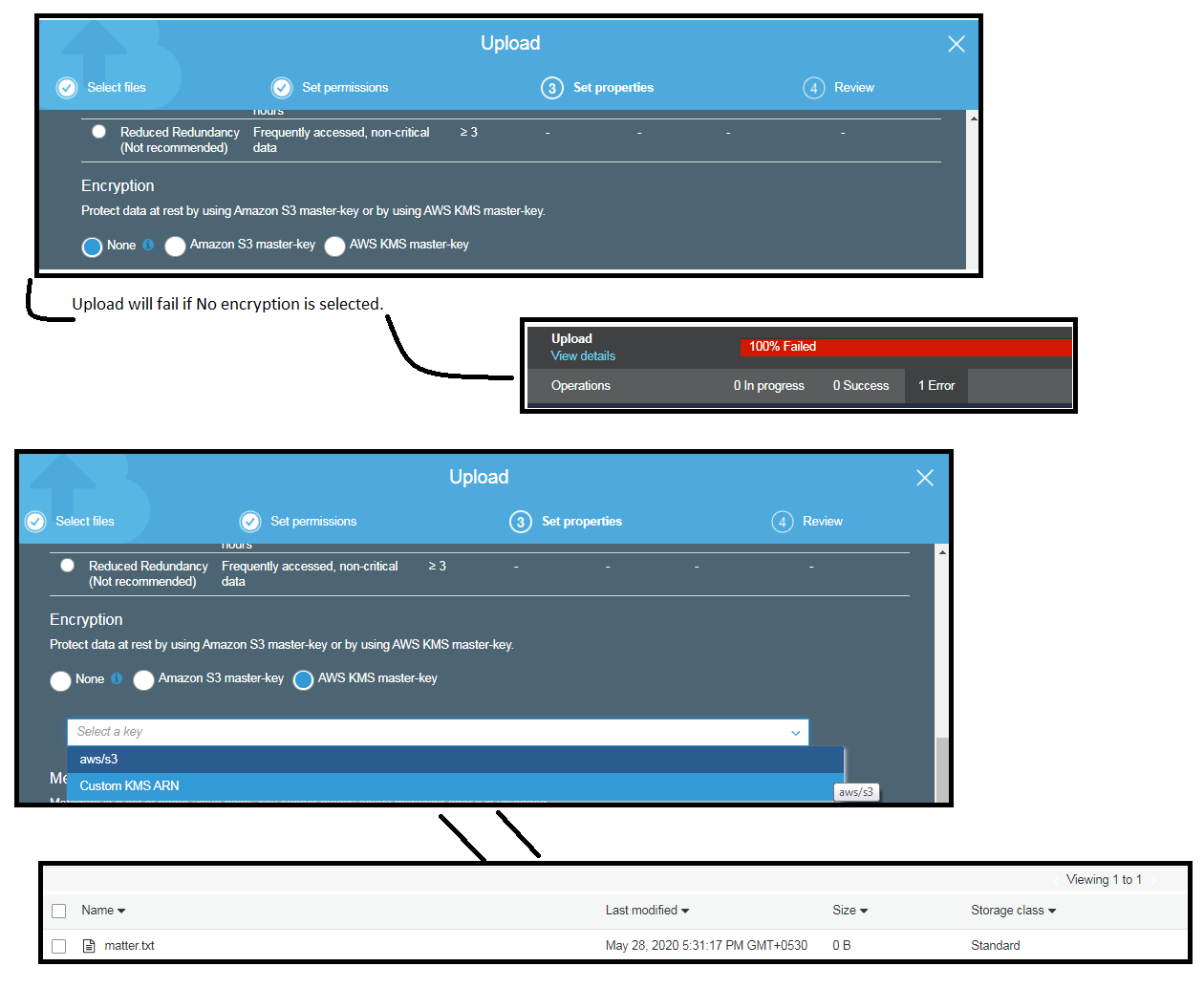
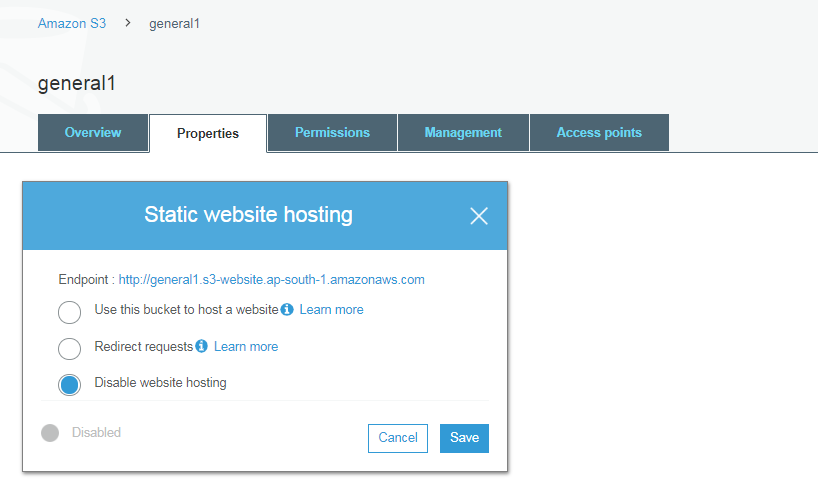
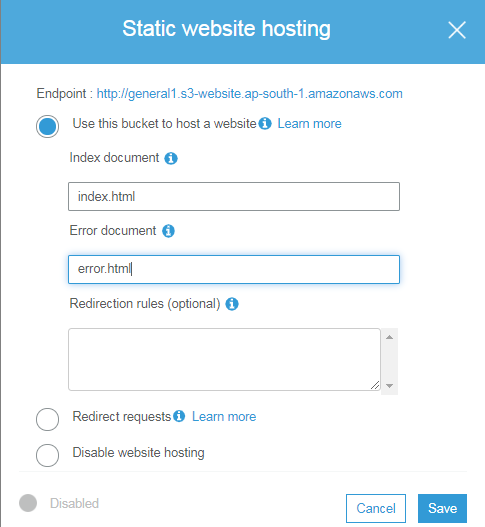
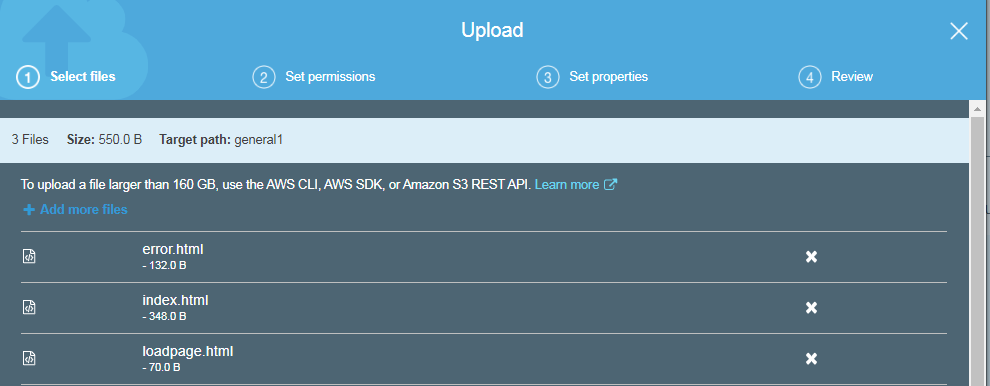
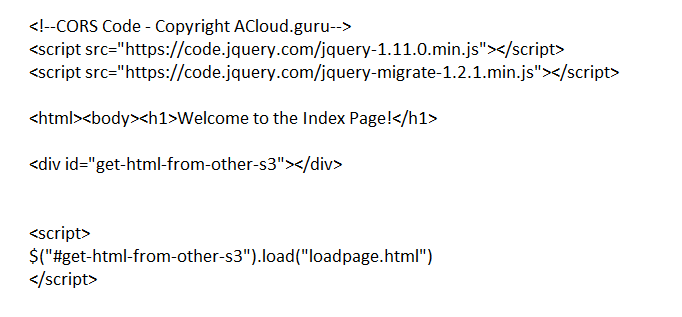
For puts of new objects : We have read after write consistency. i.e As soon as the file is uploaded to S3 the file is available to read.  
Eventual Consistency : If we are going to change a file / overwrite the file with a **new version** / delete the file – It can take time to propagate around S3. It is not going to be immediate.  
  
Basics of S3.  
Built for 99.99% availability : i.e up time.  
Amazon guarentees 99.9% availability.  
Amazon also guarentees 99.999 999 999% durability : The amout of data that you can expect to loose over a given year.  
By maintaing versions of your file, you can role back if you lost your file + you can also replicate your data.  
  
S3 Storage tiers/classes.  
Regular S3 : Suitable for most work loads.99.99% availability + eleven 9’s durability.  
 Stores the data redundantly across multiple (more than 2) devices in multiple facilities.  
S3 – IA : Infrequently accessed. Data that is acessed less frequently, but requires rapid access when needed. We pay less than Regular S3 but we are charged per retreival.  
S3 One Zone – IA : It is the same as ‘S3-IA’, but data is stored only in a single availability zone.It is still eleven 9’s durability but is only 99.5% availability.20% cheaper than S3-IA.  
Reduced redundancy storage : 99.99 durable + 99.99% avialibility.Phase out.  
Glacier : Used for archiving.Cheaper than other.Optimized for data that is in-frequently accessed, As it takes between 3-5 hours to restore data.

S3-Intelligent Tiering : This is used for data that has unpreditible access patters.  
 Two different tiers : Frequent access tiering (expensive than later)   
 & In-frequent access tiering .  
 This mechanism will automatically move your data to most   
 cost effective tier based on how frequently you access each object.  
 How does this work : If an object is not access in the last 30 days, it gets moved on to the in-frequent access tiering and vice versa.  
Note that we are charged very less for monitoring objects.  
  
On What basic are you charged.  
Storage per GB, very minimal  
Num of get,put,copy requests..  
Amount of data transferred out of S3.  
charged for Transfer Accellaration(used to accelarate file transfer speeds, by using cloud front.)  
  
S3 Security.  
Buckets : No public access by default.  
Bucket policies : Applied at bucket level and applies to all objects, Written in JSON with policiy generator tool.  
Access Control lists: Applied at object level in the bucket.  
Access logs : To track the different access requests that are being made to objects, they are written to another s3 bucket.  
  
S3 ACL and Bucket policies.  
  
Creating a bucket screenshot.  
  
  
  
  
  
After creating the bucket, How do all the page views look like and what you can do with them.  
  
Overview.  
  
  
Bucket Properties + Advanced Settings.  
  
  
  
  
Bucket permissions / Public Access.  
  
  
Bucket permissions / ACL  
  
  
Bucket permissions / Bucket Policy.  
  
  
Policy Generator for bucket policy.  
  
Principal : The entity to which you are going to ‘Allow’ access to.It could be another IAM user (user arn : arn:aws:iam::395037328646:user/Walluri ) or another S3 bucket etc.   
  
ARN : The arn of the resource to which we are applying the bucket policy to.  
  
  
  


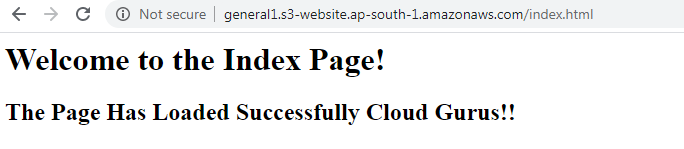
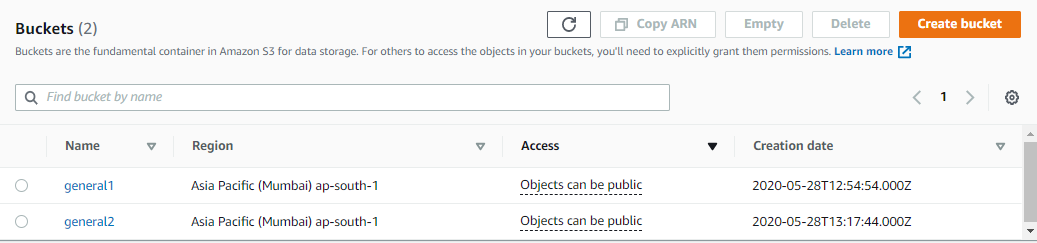
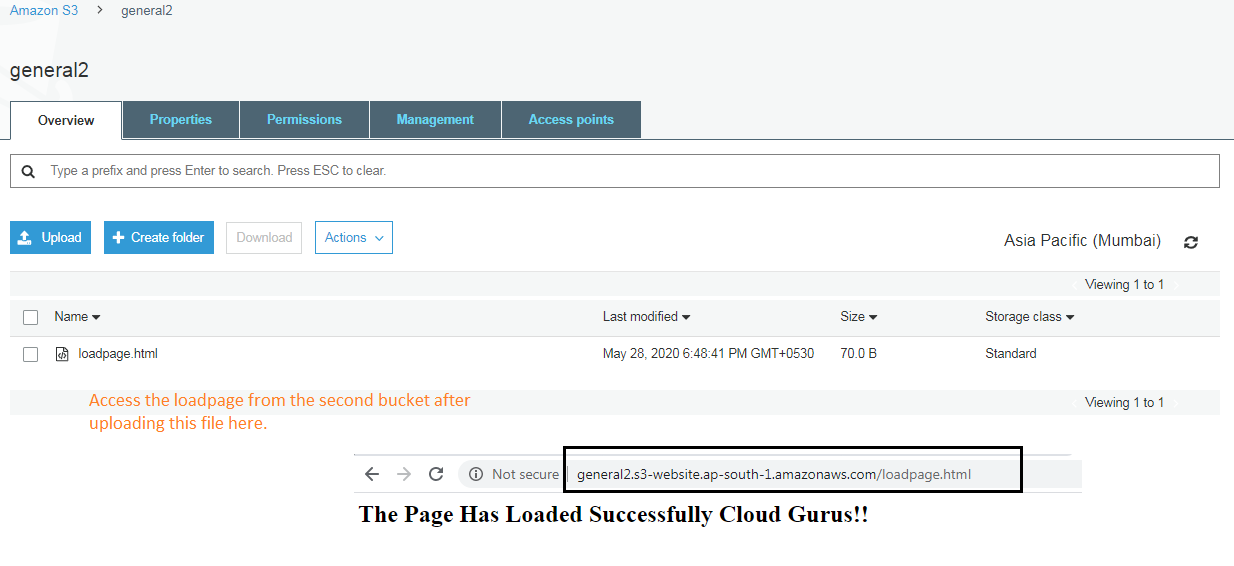
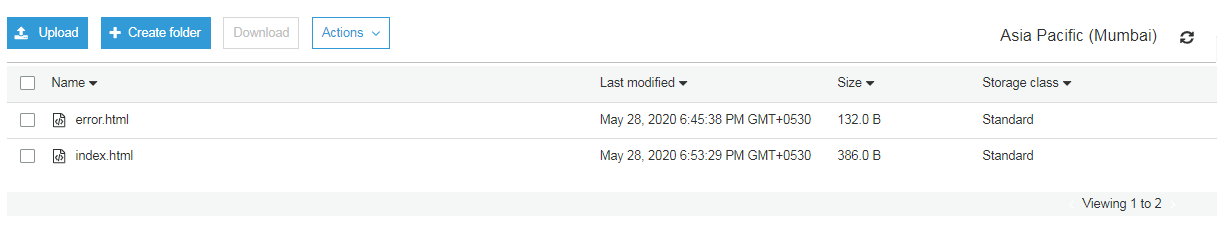
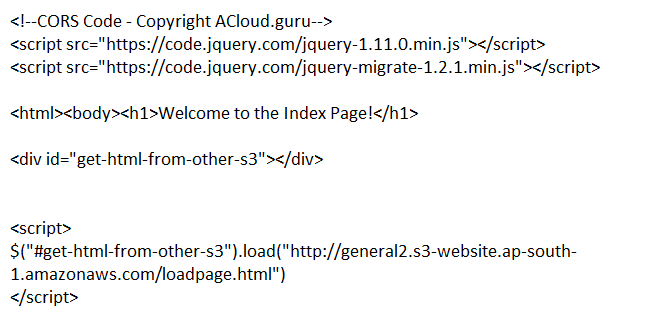
Bucket Management / Lifecycle.  
  
  
Bucket Management /Replication.  
  
  
  
Bucket Management /Analytics.  
  
  
Bucket Management /Metrics.  
  
Bucket Management / Inventory  
  
  
Access Points.  


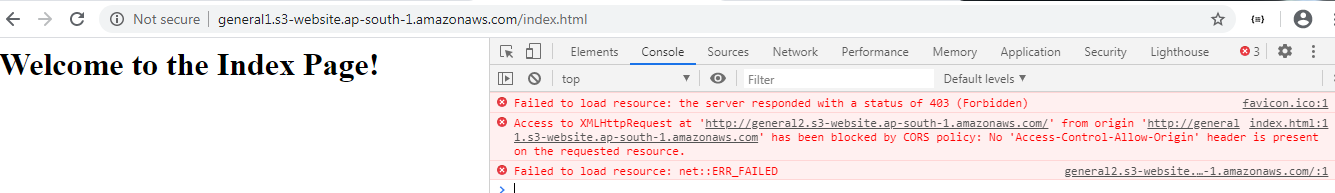
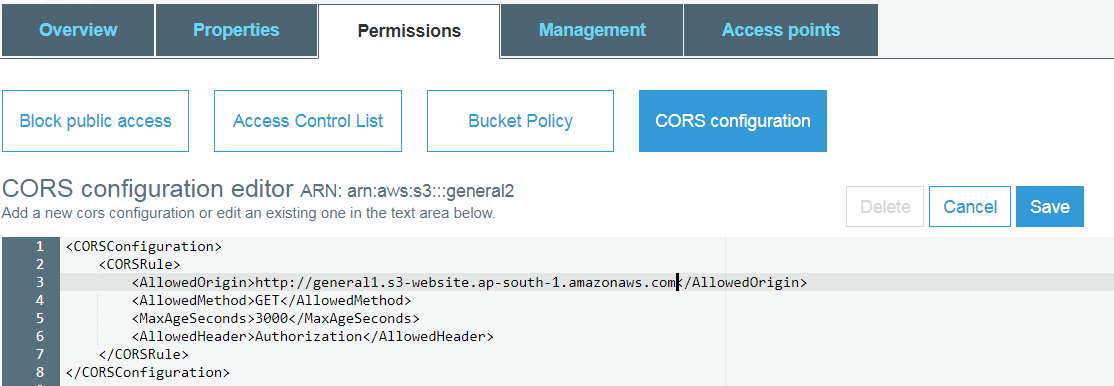
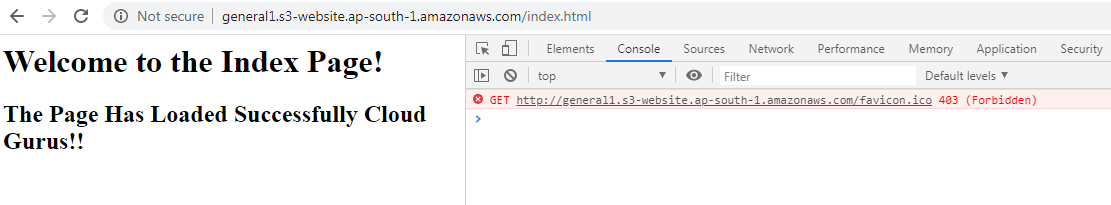
How many buckets do you have ?

  
  
Create folders in a bucket.  
  
  
  
  
Select the folder and upload files.  
  
step 1 : select file.  
  
  
  
step 2 : Manage Permissions.  
  
  
Step 3 : Storage class and Encryption with metadata and tags.  
  
  
  
  
  
  
Click on the uploaded file and see the different ways you can access it.  
  
How to make the file publicly accessible ?  
When I try to upload another file I get the below greyed text.  
  
  
So let us change the bucket permissions. Thus we are updating the bucket policy to allow public access to the objects.  
  
  
Now when uploading a file to this bucket we can decied if we want to grant public access or not.  
  
  
  
  
What about the image file uploaded above ? you can grant access in the below way.  
  
  
What pages does an uploaded file have?  
  
  
  
  
  
  
  
  
  
S3 – ENCRYPTION. (Bucket level)

  
  
How do you ENFORCE encryption every time a put request in intiated to the bucket.  
  
S3 checks for the ‘Expect: 100-continue’ key value in the header When a file is being uploaded to it.  
This means that S3 can reject or acknowledge based on the above key value.  
i.e S3 can actually reject your message based on contents of the header, And then the body of the message will not be sent.  
  
If the file is to be encrypted at upload time, the ‘x-amz-server-side-encryption’ parameter will be included in the request header.  
For SSE-S3 : we have to pass a value of ‘AES256’  
For SSE-KMS : We have to pass ‘ams:kms’  
i.e when this parameter is present in the header of the PUT request, it tells S3 to encrypt the file at the time of upload, using the specified encryption method.  
Now to reject requests that do not have this header, WE just need to define a bucket policy that rejects a denies a PUT request, which does not include a ‘x-amz-server-side-encryption’ paramter.  
  
  
S3 – ENCRYPTION - LAB.  
You can enable encryption of all objects in a bucket by selecting the below bucket ‘properties’.  
  
  
The above still could be NULL, And we can still Enforce encryption using bucket policy.  
  
Policy generator – first half.  
  
  
Policy generator – second half.  
  
  
Now lets try to upload without encryption and with encrytion.  
  
  
  
 CORS configuration lab  
- It a mechanism of allowing code that is in one S3 bucket to access code that is in an another S3 bucket.   
  
Create a bucket and note the static website url.   
  
  
  
Upload the two html files and a error.html file in the bucket1.  
  
  
Index.html contents. Loadpage.html is a simple html file.  
  


Try to access the index.html in the bucket1. [The load page conmtent is also loaded]

  
  
Create another bucket.  
  
  
Upload the loadpage.html to second bucket.  
  
Change the index.html (content as shown below) and upload it again after deleting it.  
Delete the loadpage.html file from the first bucket.  
Bucket 1 contents are also below.  
  


When you try to access the page we get cors error.  
  
  
  
  
Change the cors configuration as below for the second bucket.  
  
  
The page will work.  


S3 optimization performance.  
For GET-intensive workloads : Use Cloud front.  
Mixed workloads : Avoid sequential key names for your objects to prevent multiple objects from being stored on the same partition.  
  
Amazon S3 was upgraded and it supports 3500 PUT requests per second  
 and 5500 GET requests per second.  
  
So logical and sequential naming patters can now be used unlike mentioned above.