

Cloud Computing Assignment 3

Details :

- Name : P K Navin Shrinivas
- SRN : PES2UG20CS237
- Section : D

Task 1 :

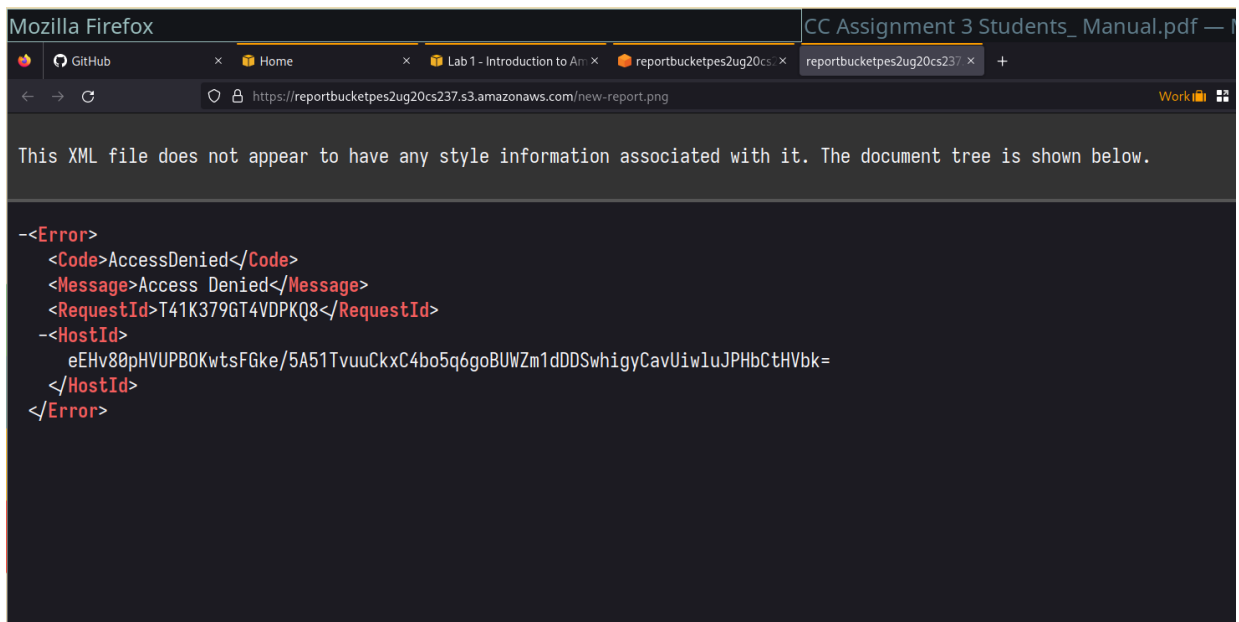
- Create the S3 bucket with `reportbucketSRN` as the name [1a]

The screenshot shows the 'Create bucket' page in the AWS Management Console. The breadcrumb navigation is 'Amazon S3 > Buckets > Create bucket'. The page title is 'Create bucket' with an 'Info' link. A subtitle states 'Buckets are containers for data stored in S3. Learn more'. The 'General configuration' section contains a 'Bucket name' input field with the value 'reportbucketpes2ug20cs237', an 'AWS Region' dropdown menu set to 'US East (N. Virginia) us-east-1', and a 'Choose bucket' button. A note indicates that settings from an existing bucket can be copied.

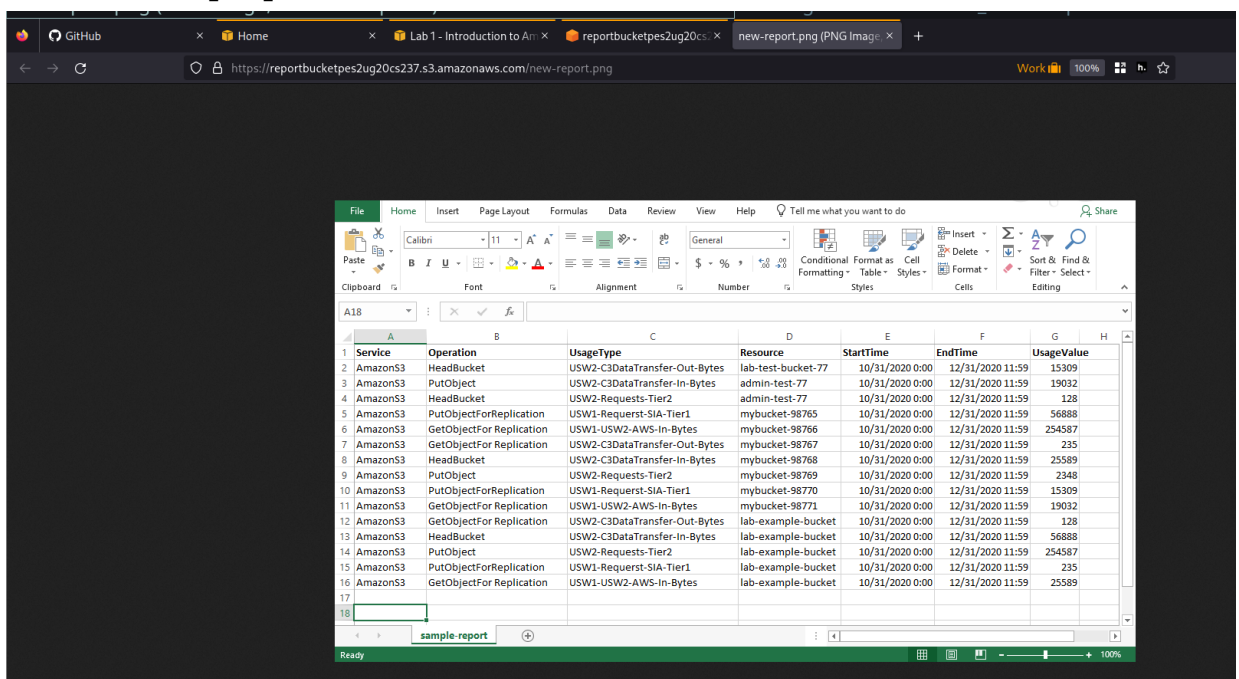
The screenshot shows the 'Buckets' page in the AWS Management Console. A green success banner at the top states 'Successfully created bucket "reportbucketpes2ug20cs237"' with a 'View details' button. Below the banner, the breadcrumb navigation is 'Amazon S3 > Buckets'. There is an 'Account snapshot' section with a 'View Storage Lens dashboard' button. The 'Buckets (1)' section includes a search bar, a table of buckets, and action buttons like 'Refresh', 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. The table lists one bucket: 'reportbucketpes2ug20cs237' in the 'US East (N. Virginia) us-east-1' region, with 'Bucket and objects not public' access and a creation date of 'March 7, 2023, 22:05:17 (UTC+05:30)'.

Name	AWS Region	Access	Creation date
reportbucketpes2ug20cs237	US East (N. Virginia) us-east-1	Bucket and objects not public	March 7, 2023, 22:05:17 (UTC+05:30)

- After uploading `new-report.jpg` try accessing the image (from the link in the object console), access should be denied [1b]



- After changing `object ownership` and `bucket setting` to allow to access throughg `ACL`, when we try accessing the same link [1c]

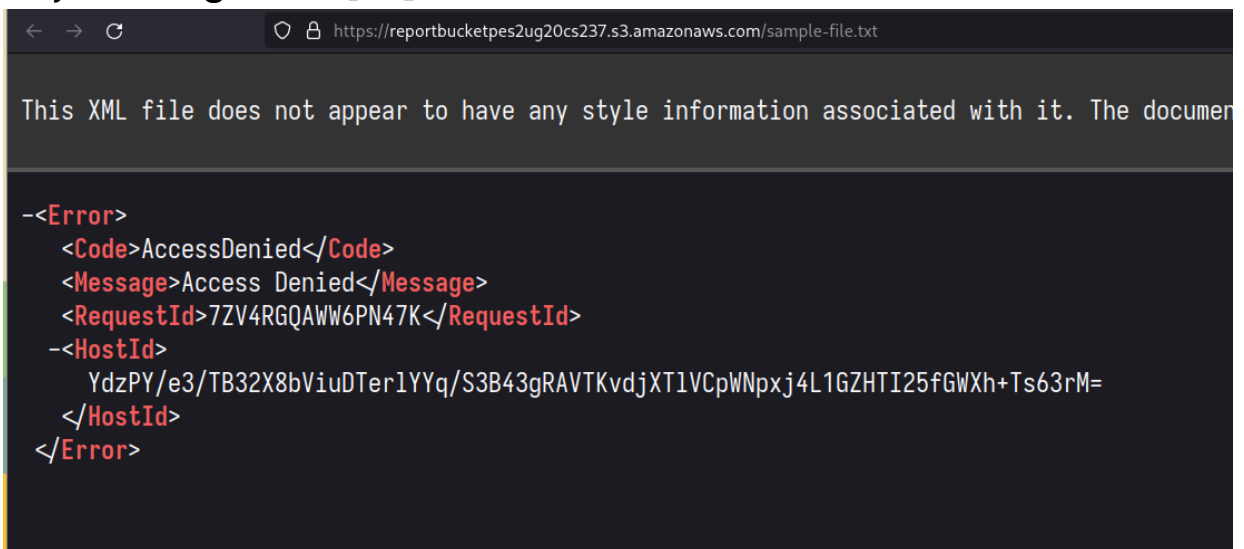


- Now we enter our already running EC2 instance using `session manager` and use the `aws s3` binary to access the

bucket. When we try uploading a file to the bucket [1d]

```
/home/ssm-user
sh-4.2$ ls
reports
sh-4.2$ cd reports
sh-4.2$ ls
dolphins.jpg files.zip report-test1.txt report-test2.txt report-test3.txt whale.jpg
sh-4.2$ aws s3 cp report-test1.txt s3://reportbucketpes2ug20cs237
upload failed: ./report-test1.txt to s3://reportbucketpes2ug20cs237/report-test1.txt An error occurred (AccessDenied) when c
sh-4.2$ aws s3 cp report-test1.txt s3://reportbucketpes2ug20cs237
upload failed: ./report-test1.txt to s3://reportbucketpes2ug20cs237/report-test1.txt An error occurred (AccessDenied) when c
alling the PutObject operation: Access Denied
sh-4.2$
```

- Let's go back and upload `sample-file.txt`, Now when we try accessing the file, we are denied permissions...hence we need to get s3 to change permissions (A common ACL) for all objects together [1e]



The screenshot shows a web browser window with the address bar displaying `https://reportbucketpes2ug20cs237.s3.amazonaws.com/sample-file.txt`. The main content area shows an XML error message:

```
-<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>7ZV4RGQAWW6PN47K</RequestId>
  <HostId>
    YdzPY/e3/TB32X8bViuDTer1YYq/S3B43gRAVTKvdjXT1VCpWNpxj4L1GZHTI25fGWXh+Ts63rM=
  </HostId>
</Error>
```

- We can use IAM to make this happen, let's first copy our `ARN` from `EC2InstanceProfileRole` (this is the role EC uses to access s3 objects)

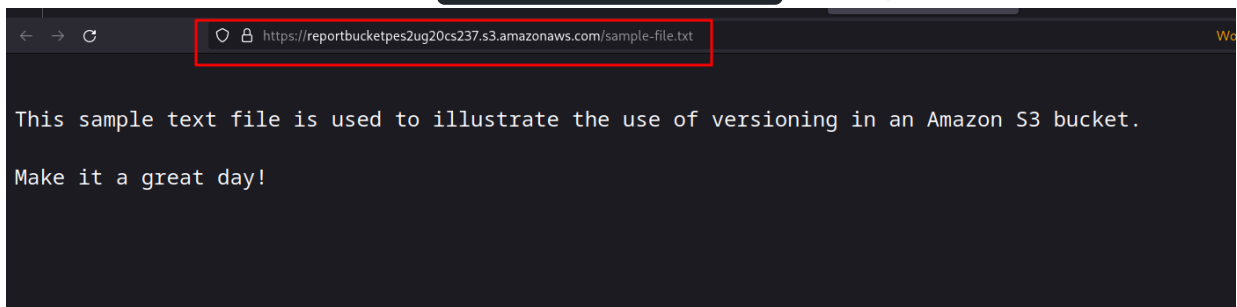
: `arn:aws:iam::438965201338:role/EC2InstanceProfileRole`.

We can also make note of the bucket ARN present in bucket policy : `arn:aws:s3:::reportbucketpes2ug20cs237`.

- let's now create the policies using `policy generator` :
- ARN is going to be `s3arn/*` and principal is going to be EC2 role ARN. Click on generate policy copy the config and paste in bucket policy!
- Now we can upload objects from EC2 instance :

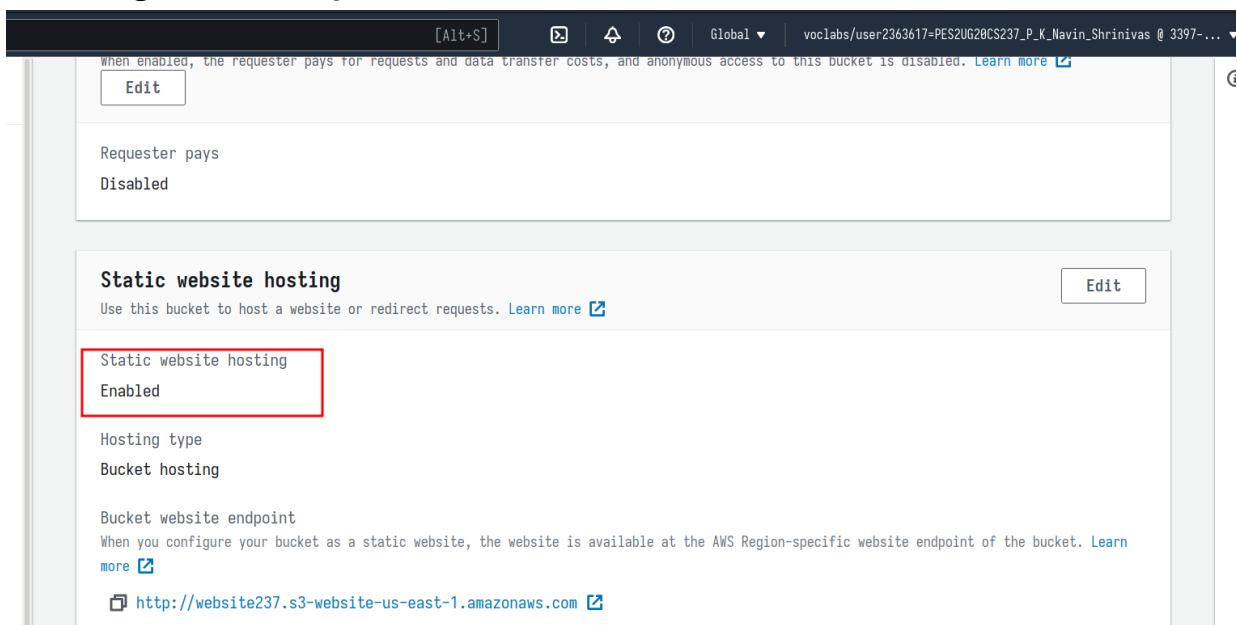
```
sh-4.2$ aws s3 cp report-test1.txt s3://reportbucketpes2ug20cs237
upload: ./report-test1.txt to s3://reportbucketpes2ug20cs237/report-test1.txt
sh-4.2$
```

- To give access to the browser, we need to add another bucket policy, hence use the same **policy generator** and principal as * and allow everything. Now when we go access the same website for **sample-file.txt**, we get this [1f]

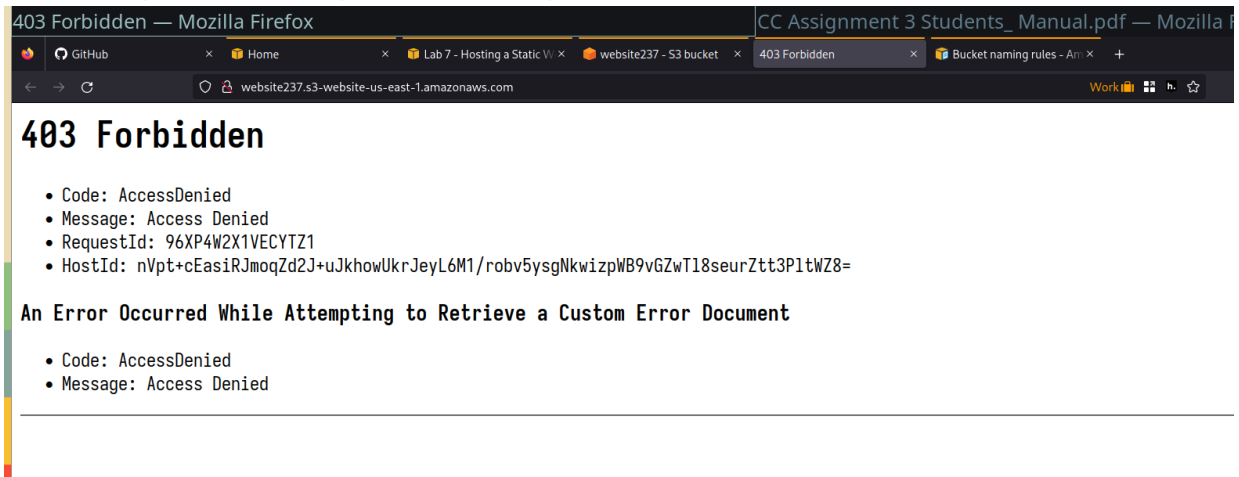


Task 2 : Serving a site using S3

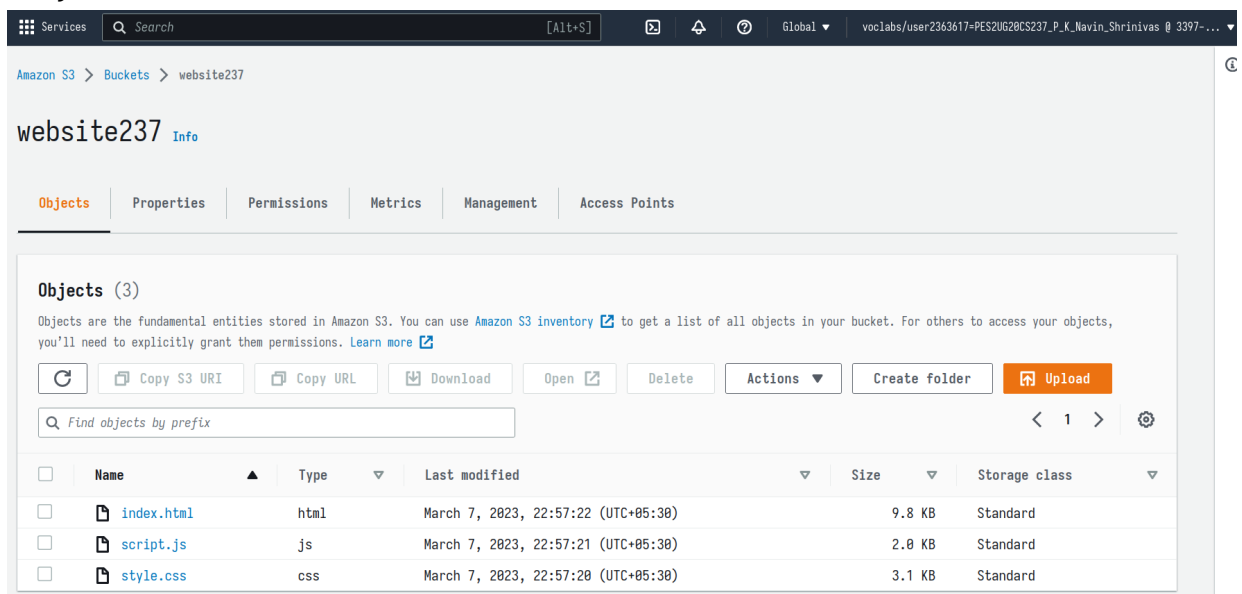
- Create a new S3 service with public access while creating itself, now move to S3 setting and scroll down to website hosting. Manually enter all the defaults and save [1a]



- Although opening that link gives us an 403 [2b]

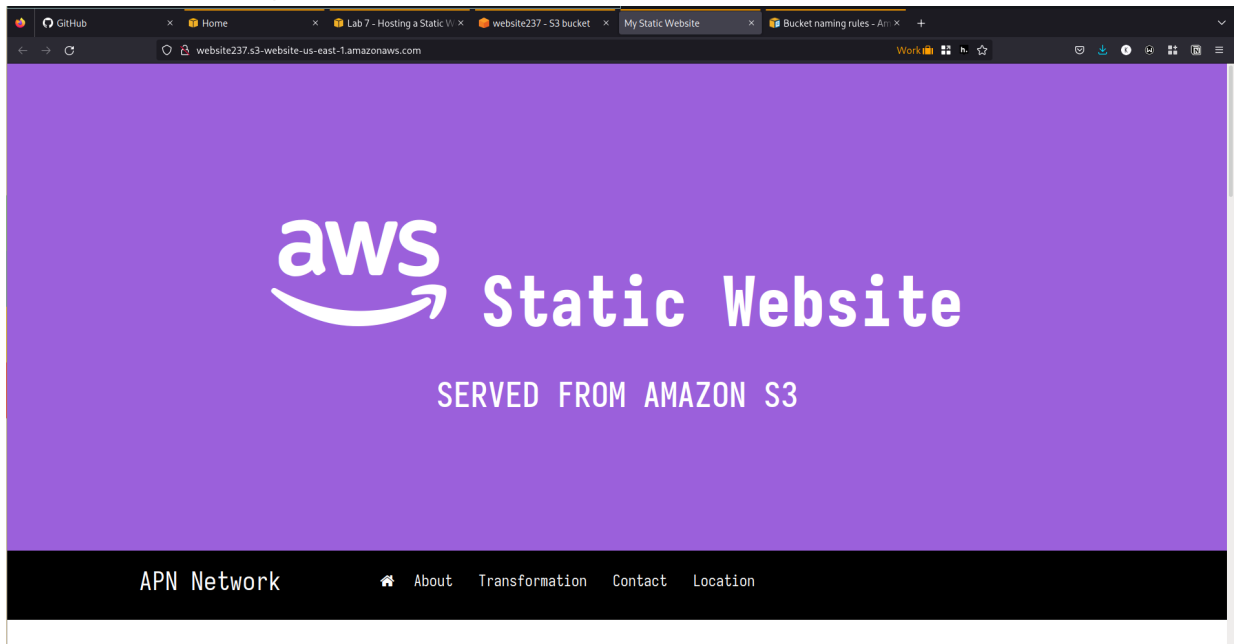


- To solve this, let's upload the files that s3 should serve as objects [2c]

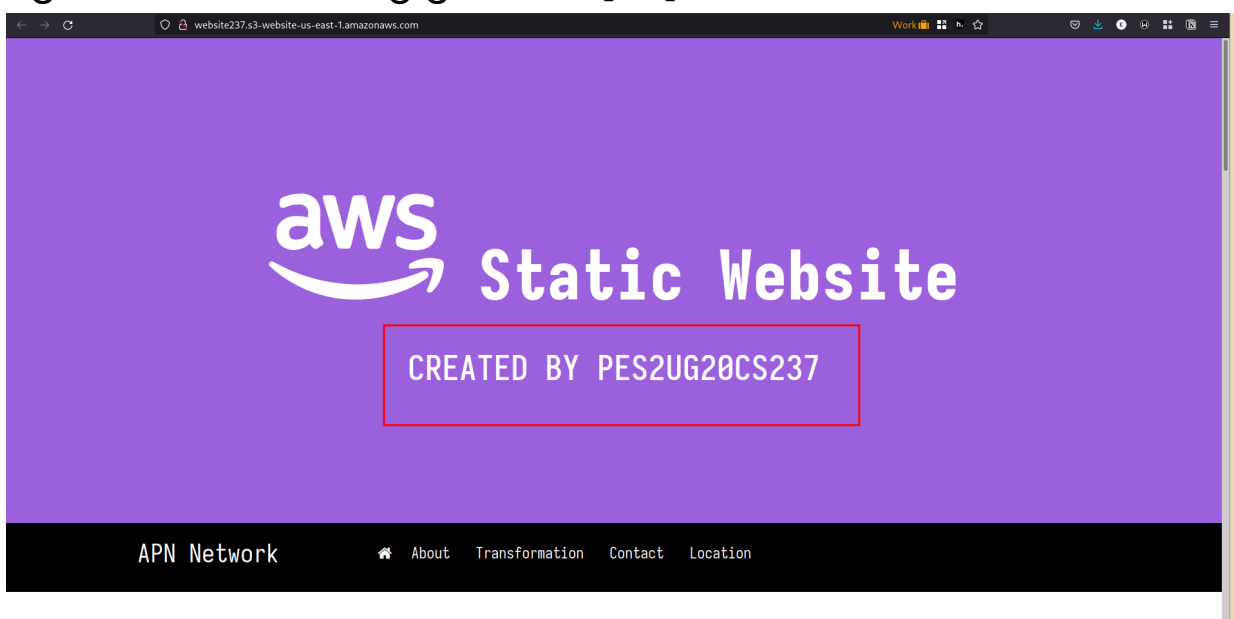


- This still gives us 403 as the objects are private, we can select all three and make them **public using ACL!**

- And now we get this [2d]



- Now we can edit the file locally and upload again, it replace the old name with the same name. We now make it public again and refreshing gives us [2e]



-----END OF LAB-----