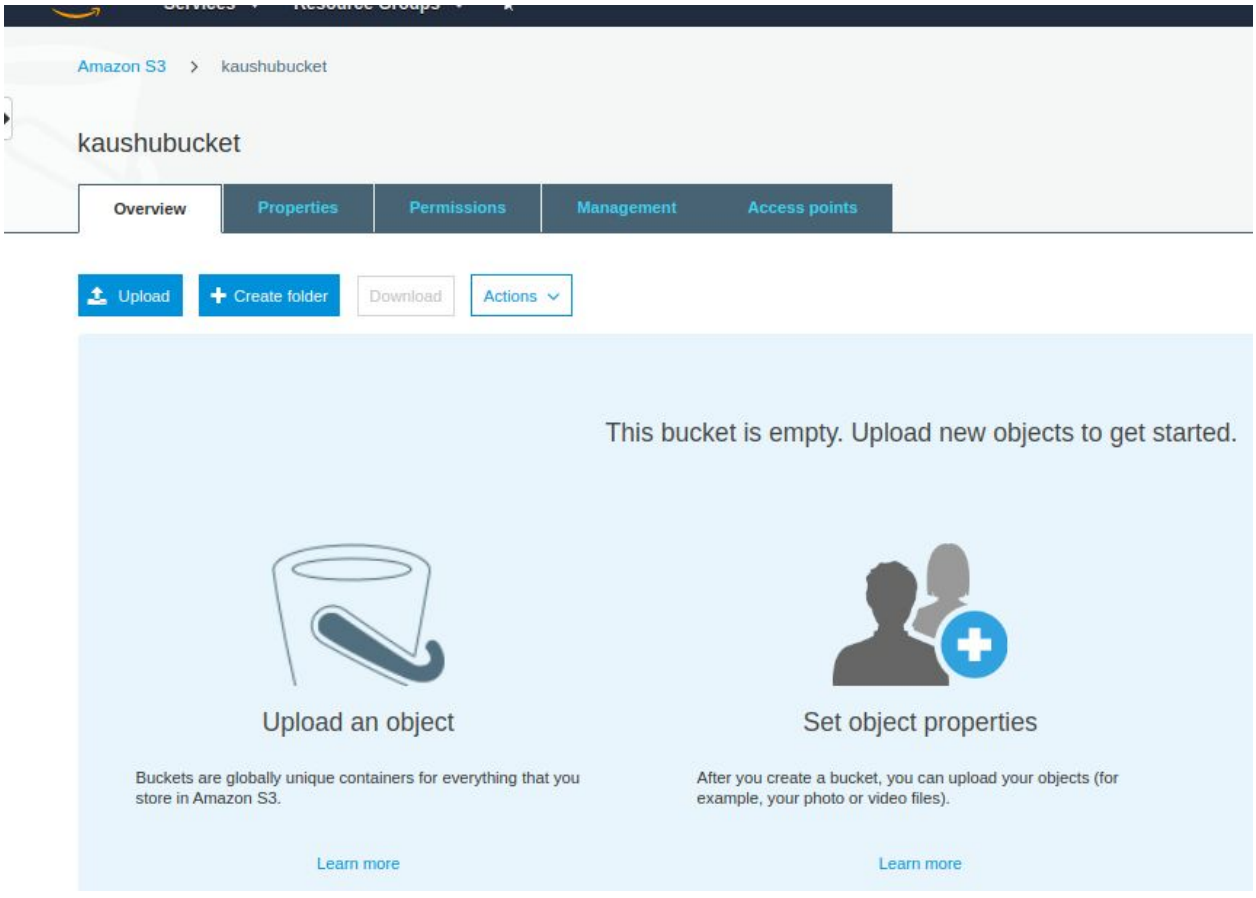
Simple Storage Service (S3)

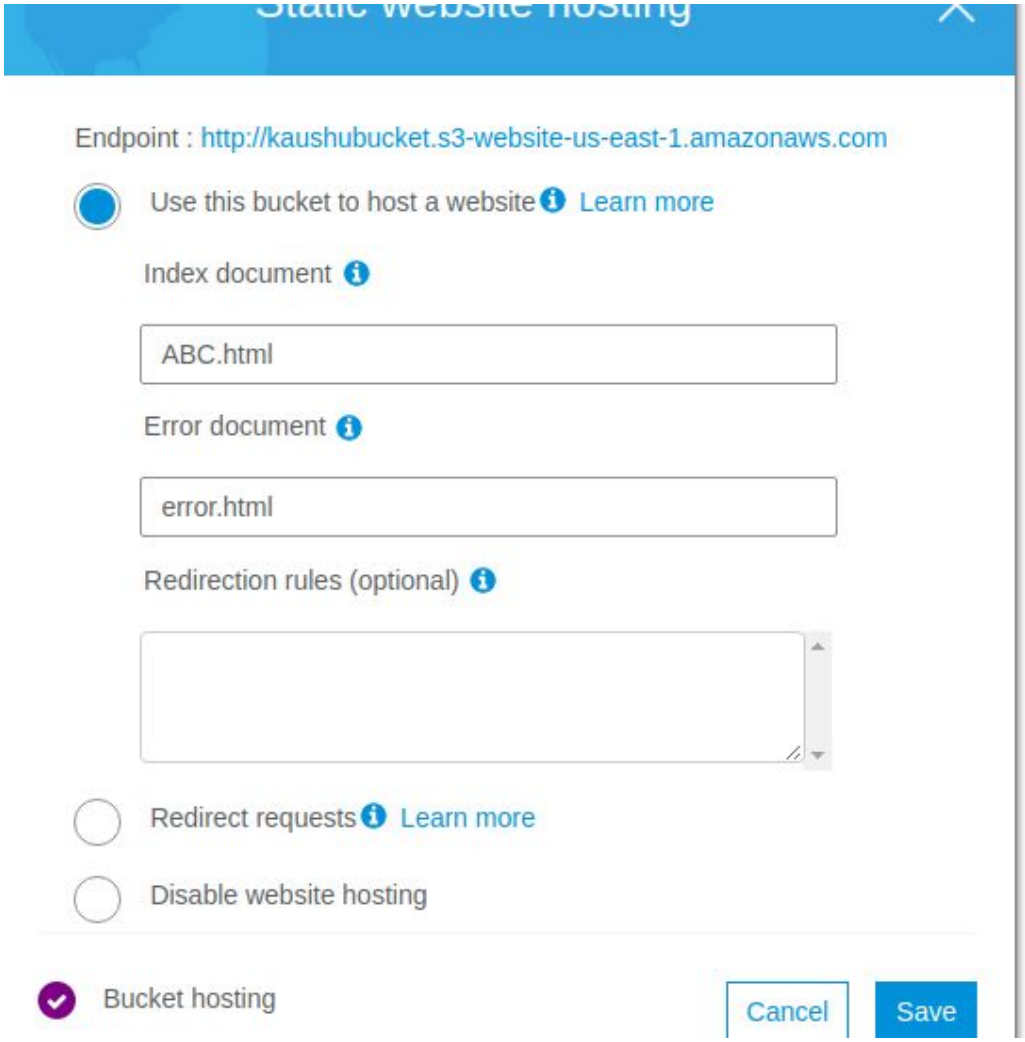
Ques 1. static website hosting using S3 (index page, error page)

Ans 1.

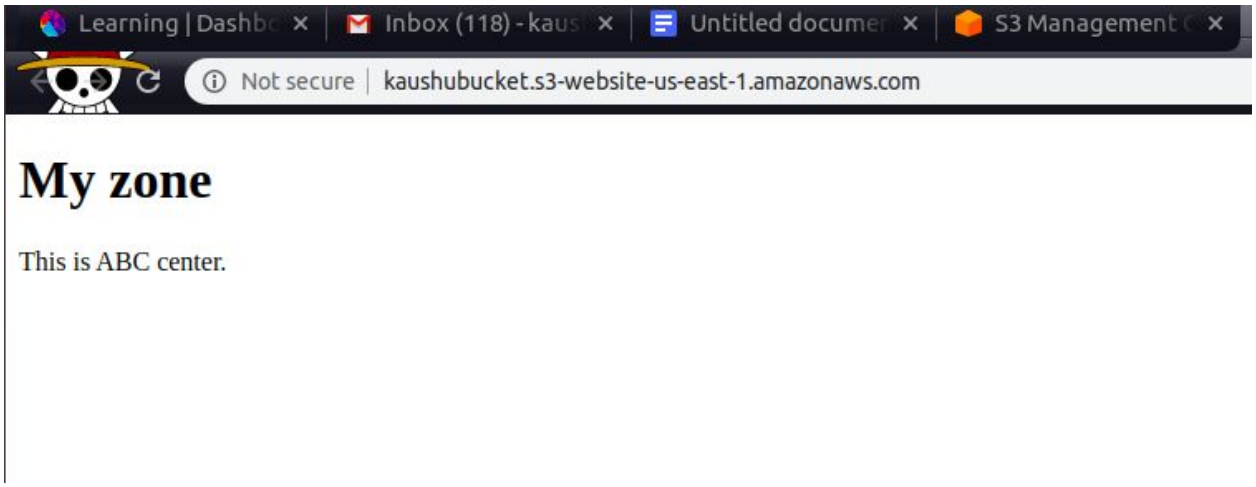
Created a bucket in S3



Hosting a static website



Open the website with the help of end points



Ques 3. . Block s3 access on the basis of

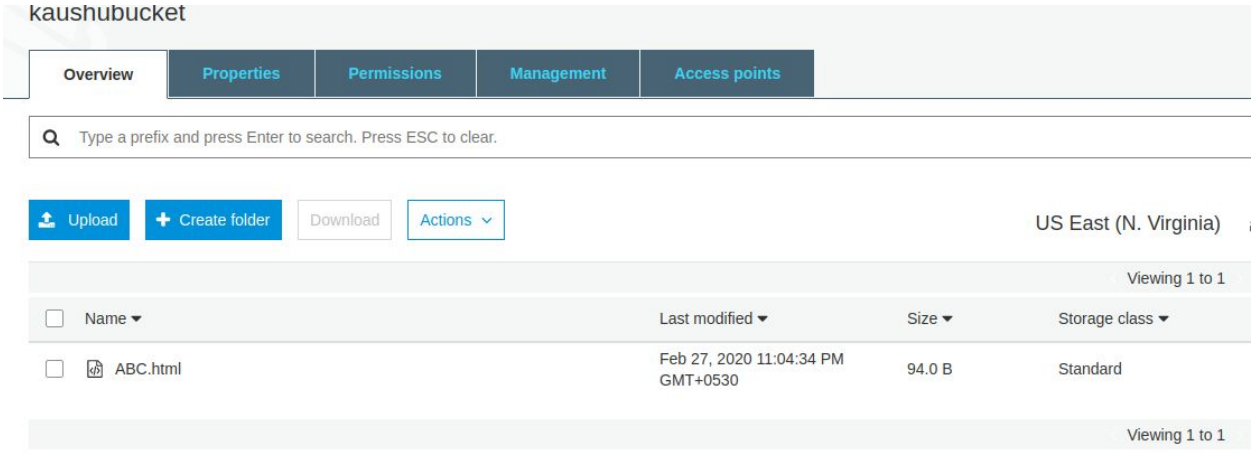
i. IP

ii. Domain

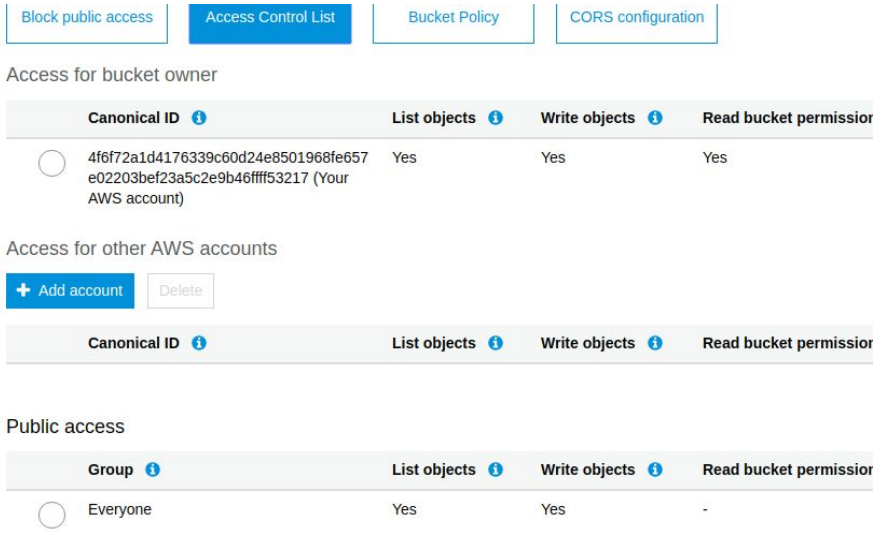
iii. Pre-signed URL(Time based)

Ans 3. Blocking Through ip address

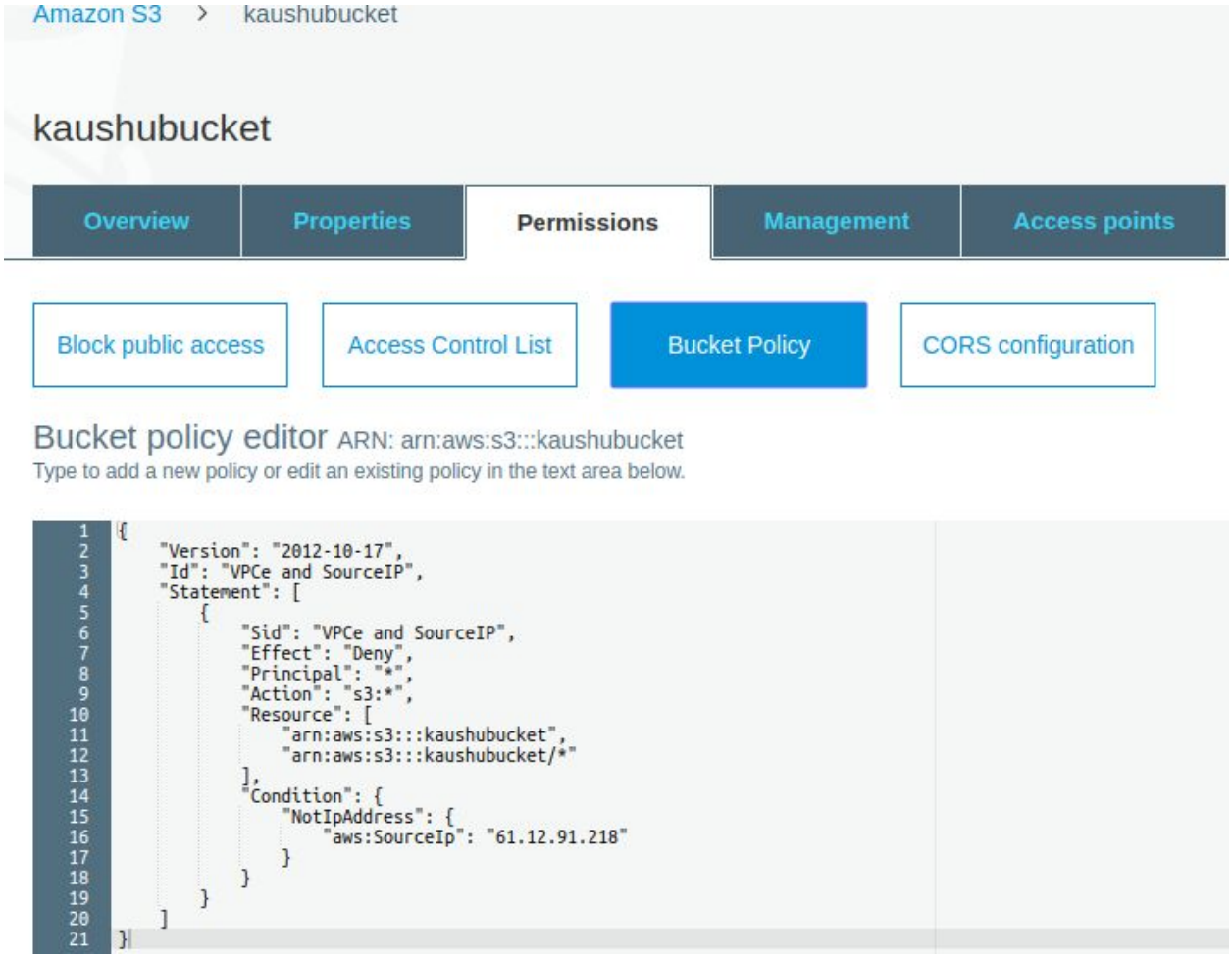
Created a bucket in s3



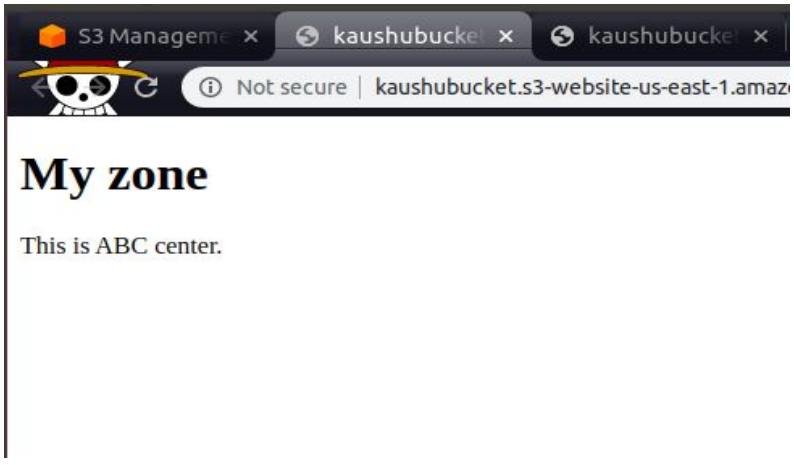
Make the bucket public



Created a policy to block the S3 for particular ips



Open the bucket with allowed ip



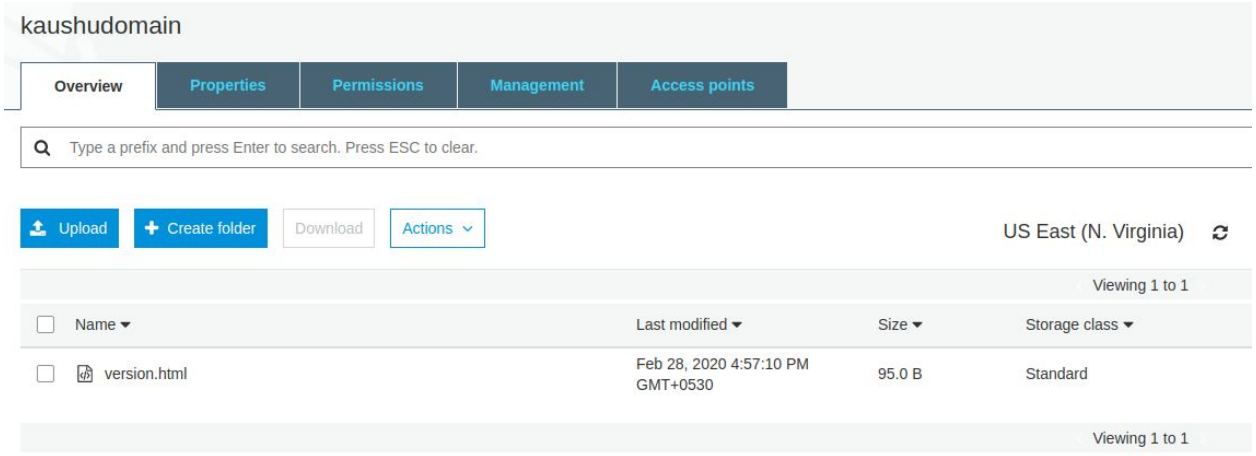
Access denied while opening the S3 with blocked ips

Blocking Through domain

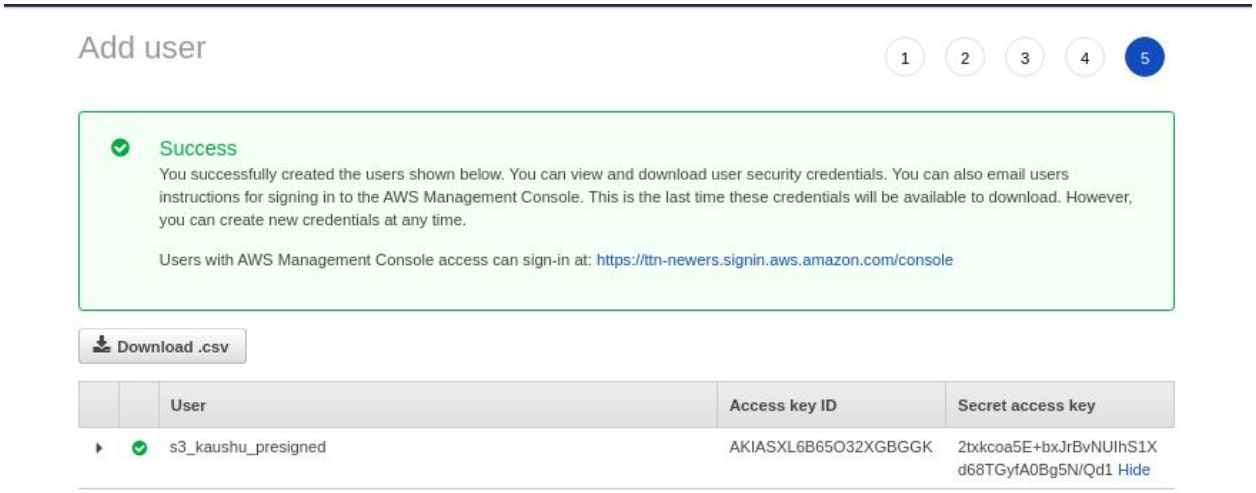
```
{
    "Version": "2012-10-17",
    "Id": "VPCe and domain",
    "Statement": [
        {
            "Sid": "VPCe and domain",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::kaushudomain",
                "arn:aws:s3:::kaushudomain/*"
            ],
            "Condition": {
                "Notdomain": {
                    "aws:SourceIp": "www.example.com"
                }
            }
        }
    ]
}
```

Blocked through presigned url

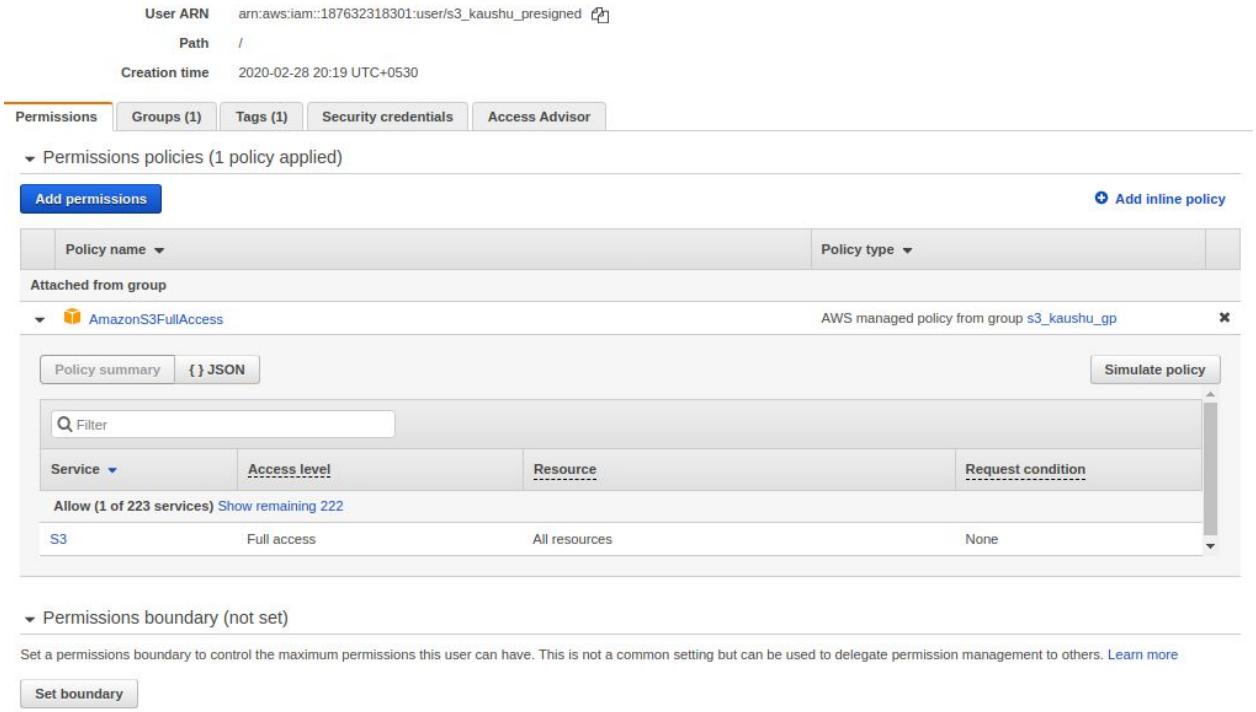
Created a bucket and it is private



Firstly i created the IAM group

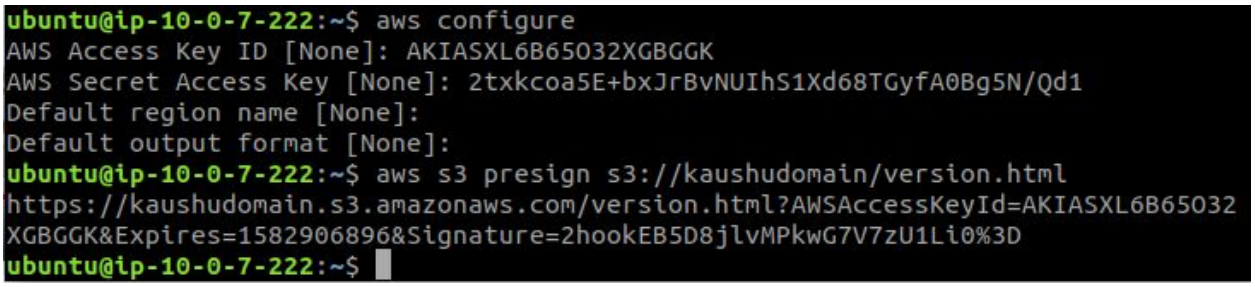


Give all the S3 permission to the user

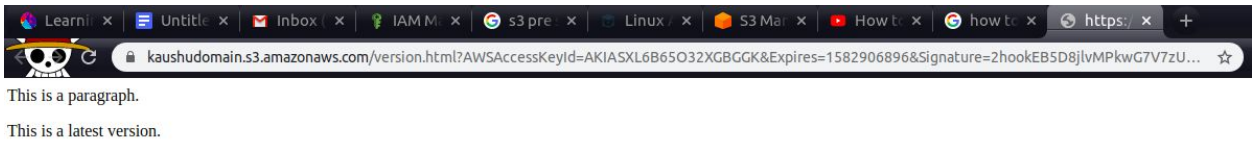


Set the policy for user

Then open the instance in ec2 and generate the presigned url using the below commands



After that access the bucket using pre signed url



Ques 4. Create RDS subnet and launch RDS instance.
what is parameter group and option group

Ans4. Firstly i created the RDS subnet

RDS > Subnet groups > kaushu_rds

kaushu_rds

Subnet group details

VPC ID
t34ak (vpc-01d9bca1ea53fdce9)

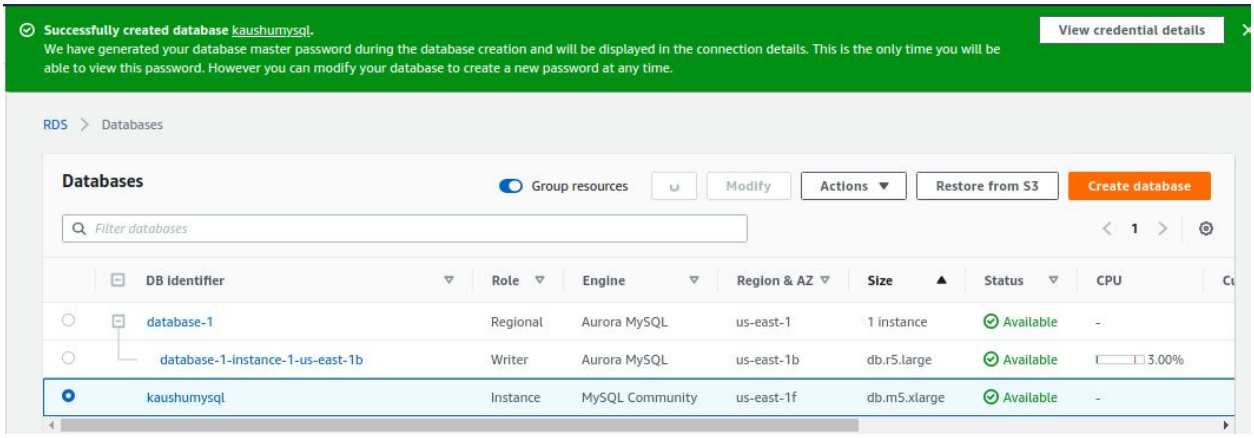
ARN
arn:aws:rds:us-east-1:187632318301:subgrp:kaushu_rds

Description
rds

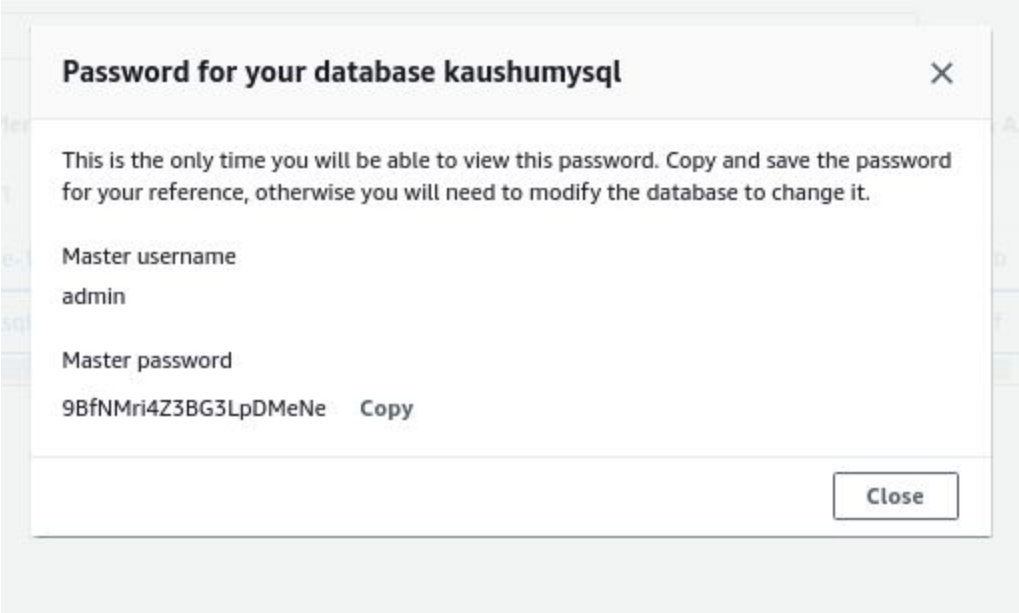
Subnets (9)

Availability zone	Subnet ID	CIDR block
us-east-1f	subnet-0dba4ee75a08a389d	10.0.48.0/20
us-east-1a	subnet-07c3a2631f7eb2d6f	10.0.194.0/24
us-east-1b	subnet-062a1eaac2376ac86	10.0.0.0/20
us-east-1c	subnet-0a976bd6d97e7ce2a	10.0.193.0/24

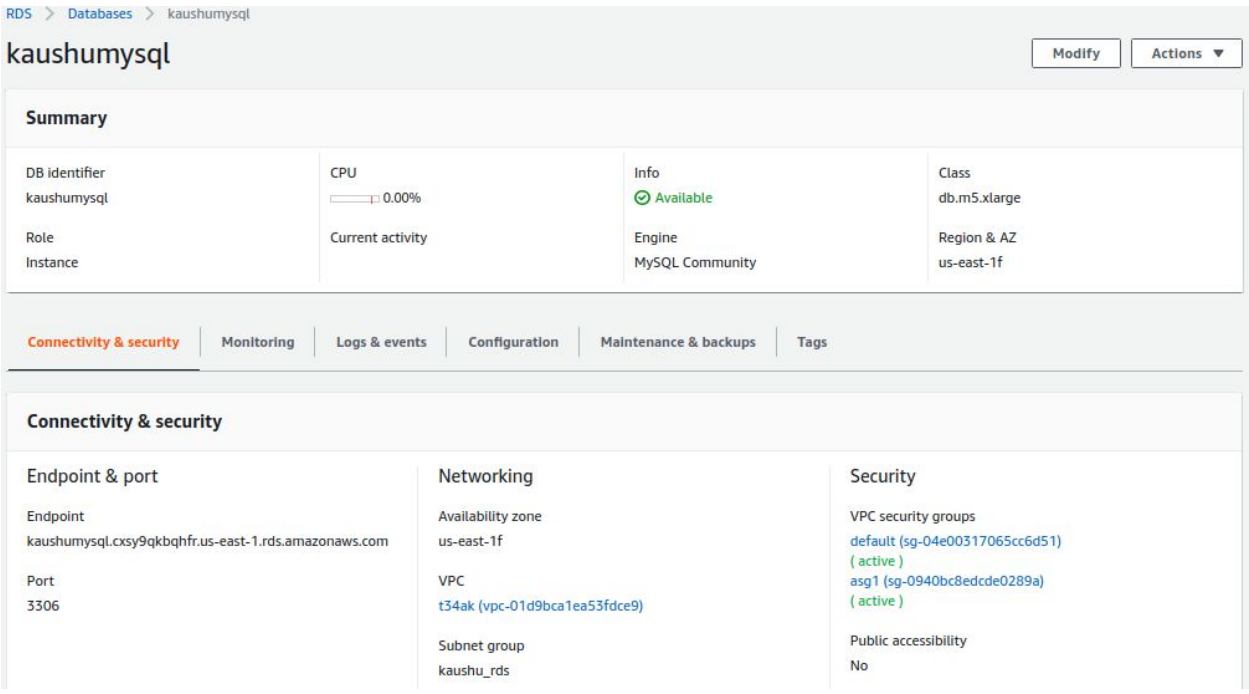
Then i created a mysql database



Database auto generated password given below



Finally database configured



what is parameter group and option group

option group-

Amazon RDS uses option groups to enable and configure these features. An option group can specify features, called options, that are available for a particular Amazon RDS DB instance. ... When you associate a DB instance with an option group, the specified options and option settings are enabled for that DB instance.

Parameter group-

DB parameter groups act as a container for engine configuration values that are applied to one or more DB instances. A default DB parameter group is created if you make a database instance without specifying a custom DB parameter group.

Ques 5. ACL, Bucket policy, IAM Policy.

Ans 5. Use IAM policies if:

- You need to control access to AWS services other than S3. IAM policies will be easier to manage since you can centrally manage all of your permissions in IAM, instead of spreading them between IAM and S3.
- You have numerous S3 buckets each with different permissions requirements. IAM policies will be easier to manage since you don't have to define a large number of S3 bucket policies and can instead rely on fewer, more detailed IAM policies.
- You prefer to keep access control policies in the IAM environment.

Use S3 bucket policies if:

- You want a simple way to grant cross-account access to your S3 environment, without using IAM roles.
- Your IAM policies bump up against the size limit (up to 2 kb for users, 5 kb for groups, and 10 kb for roles). S3 supports bucket policies of up 20 kb.

- You prefer to keep access control policies in the S3 environment.

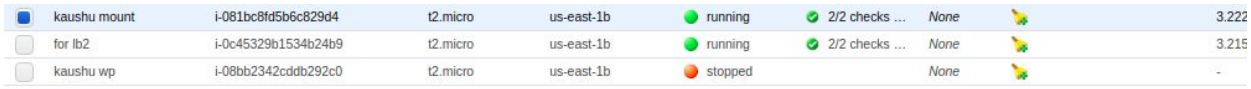
Use S3 bucket policies if:

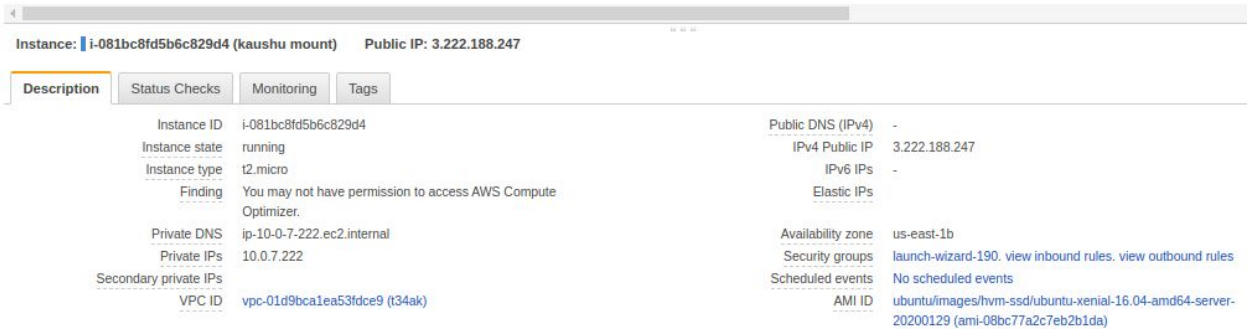
As a general rule, AWS recommends using S3 bucket policies or IAM policies for access control. S3 ACLs is a legacy access control mechanism that predates IAM. However, if you already use S3 ACLs and you find them sufficient, there is no need to change.

An S3 ACL is a sub-resource that's attached to every S3 bucket and object. It defines which AWS accounts or groups are granted access and the type of access. When you create a bucket or an object, Amazon S3 creates a default ACL that grants the resource owner full control over the resource.

Ques 5. Mount S3 to an EC2 instance

Ans 5. First created a instance in ec2

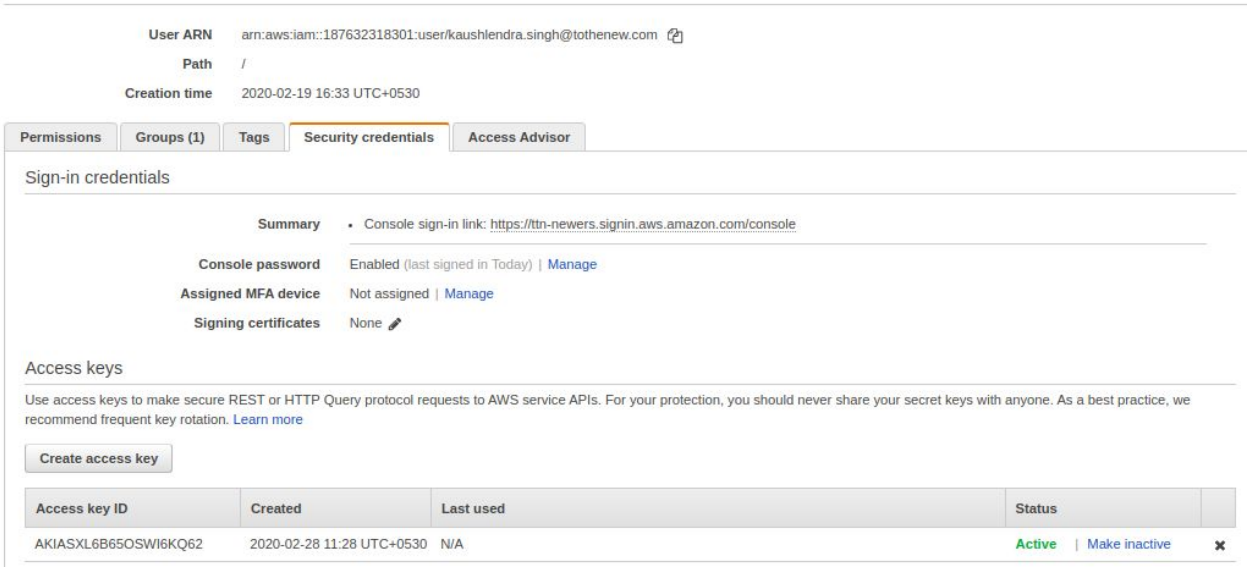




Then run the following below commands to install the repositories and packages

```
ubuntu@ip-10-0-7-222:~/s3fs-fuse$ history
    1  sudo apt-get update
    2  sudo apt-get install automake fuse fuse-devel gcc-c++ git libcurl-devel libxml2-devel make openssl-devel
    3  sudo apt-get install automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config
    4  clear
    5  sudo apt-get update
    6  sudo apt-get install automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config
    7  exit
    8  sudo apt-get install -y automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config
    9  git clone https://github.com/s3fs-fuse/s3fs-fuse.git
   10  cd s3fs-fuse
   11  ./autogen.sh
   12  ./configure --prefix=/usr --with-openssl
   13  make
   14  sudo make install
   15  which s3fs
   16  touch /etc/passwd-s3fs
```

Then create the key pairs to access the s3 bucket



After that run some following below commands

```
17  sudo touch /etc/passwd-s3fs
18  sudo vim /etc/passwd-s3fs
19  sudo chmod 640 /etc/passwd-s3fs
20  sudo vim /etc/passwd-s3fs
21  cd ..
22  cd /etc
23  sudo su
24  cd ~
25  sudo vi /etc/hosts
26  cd usr/bin/s3fs
27  cd /usr/bin/s3fs
28  cd s3fs-fuse
29  sudo vim /etc/passwd-s3fs
30  sudo chmod 640 /etc/passwd-s3fs
31  mkdir /mys3bucket
32  sudo mkdir /mys3bucket
33  ls
34  sudo mkdir mys3bucket
35  ls
36  s3fs your_bucketname -o use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket
37  s3fs kaushubucket -o use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket
38  sudo s3fs kaushubucket -o use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket
39  which s3fs
40  cd /usr/local/bin
41  ls
42  cd /usr/local/bin/s3fs
43  sudo nano /etc/rc.local
44  which s3fs
45  cd s3fs-fuse
46  cd ~
47  cd s3fs-fuse
48  df -TH
49  df -TH /mys3bucket/
```

Check mounted s3 bucket

```
ubuntu@ip-10-0-7-222:~/s3fs-fuse$ df -TH /mys3bucket/
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/xvda1     ext4  8.3G  1.7G  6.6G  20% /
ubuntu@ip-10-0-7-222:~/s3fs-fuse$
```

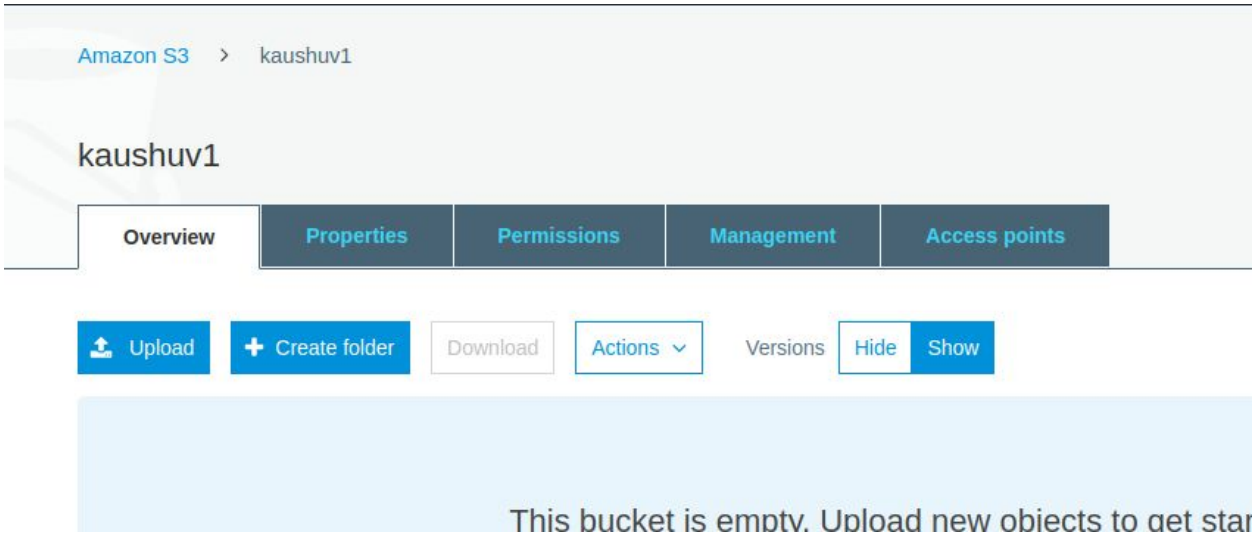
Ques 6. Change content type using s3.

Ans 6.

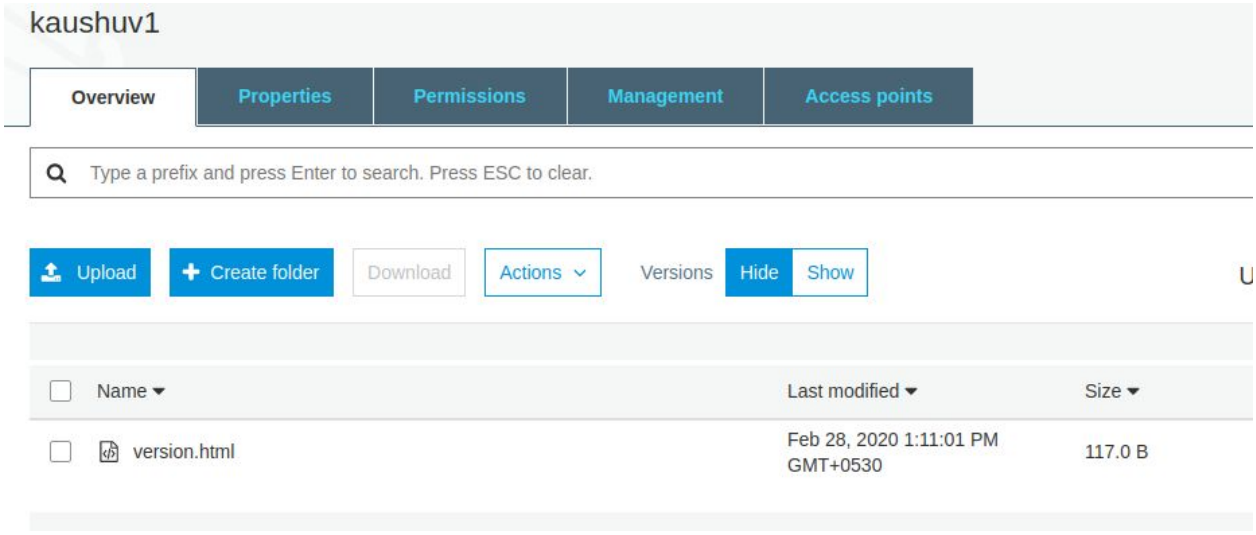
Ques 7. Retrieve previous version of S3(enable versioning).

Ans 7.

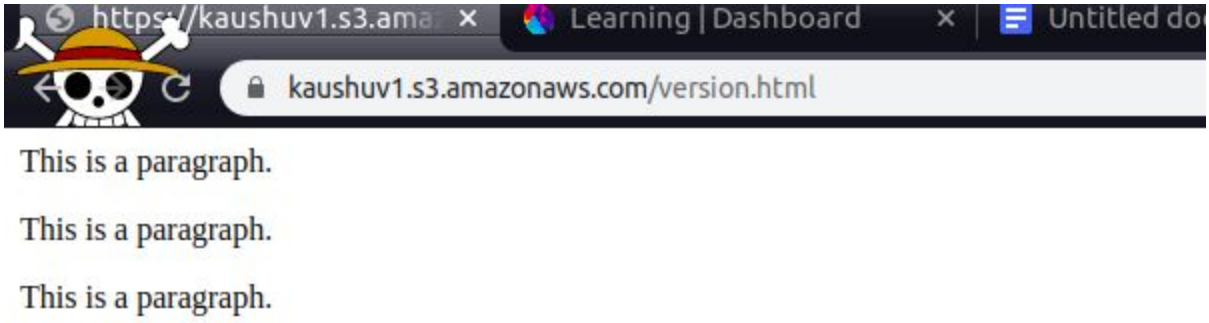
Created a new bucket and enabled the versioning



After that uploaded the html page in bucket

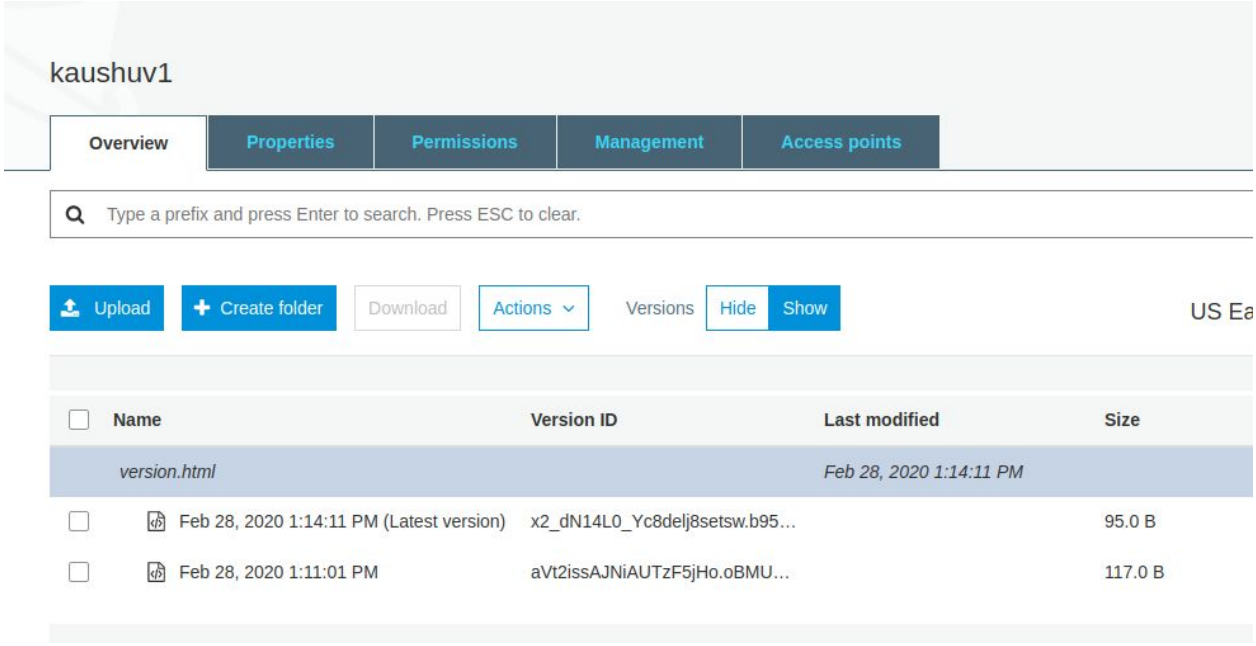


After this open the uploaded page in browser

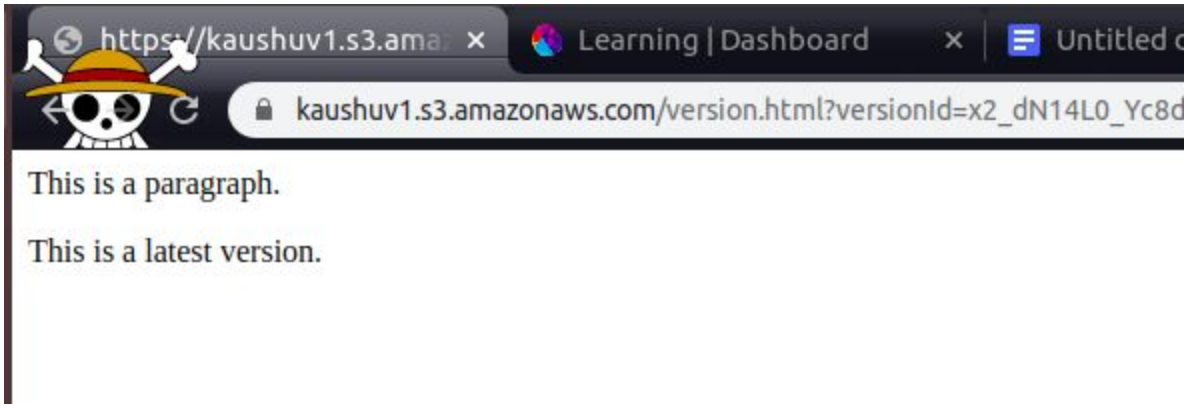


Then edit the html page and upload in the bucket

Versioning created as latest version



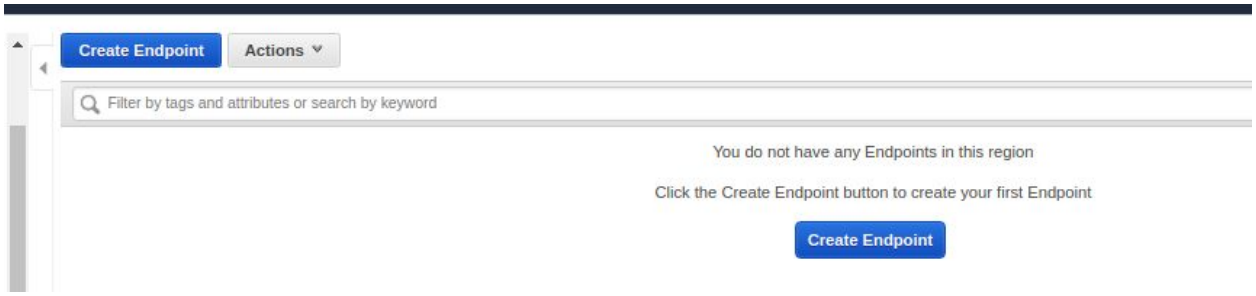
After this open the uploaded page in browser



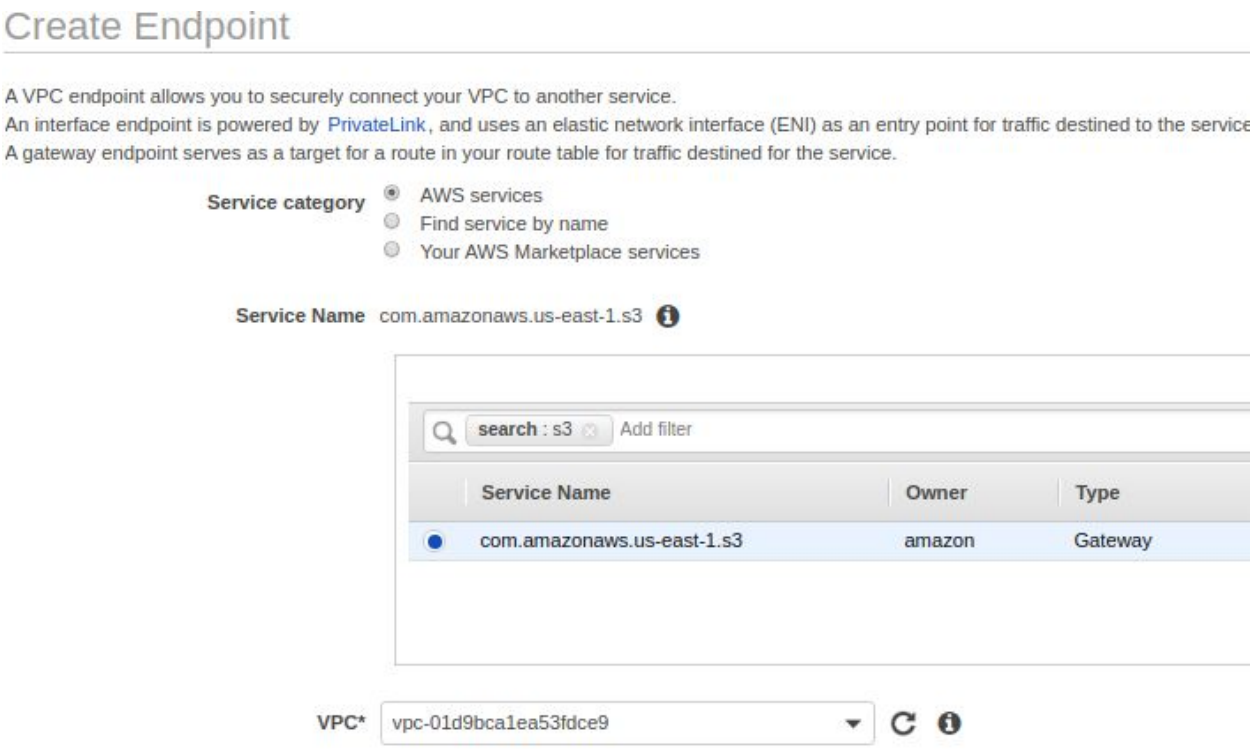
Ques 9. S3 VPC endpoint.

Ans 9.

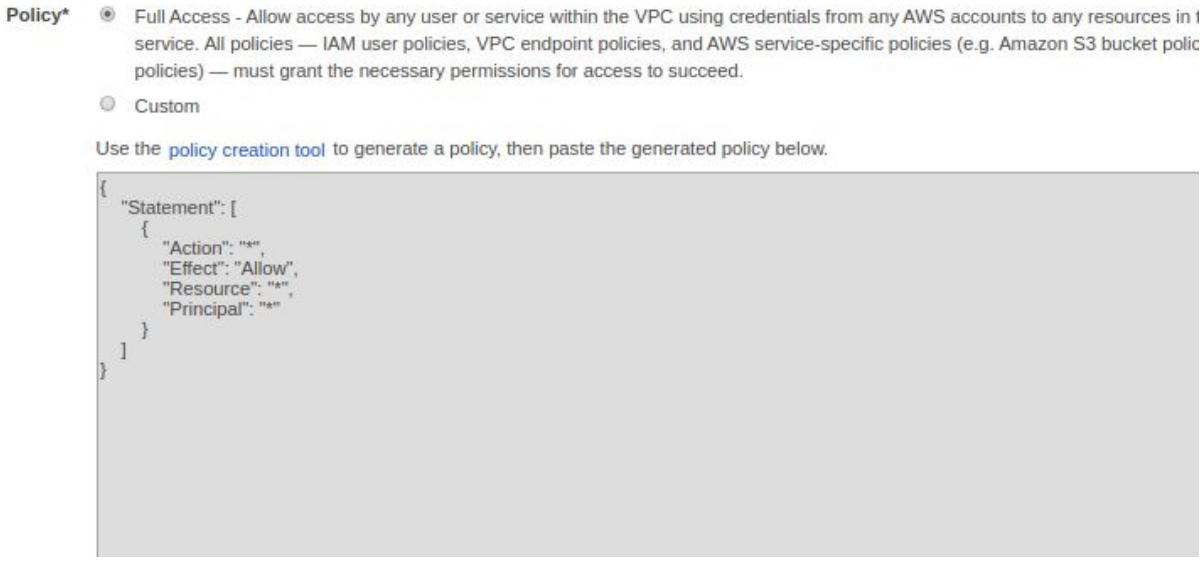
Open the vpc endpoint terminal



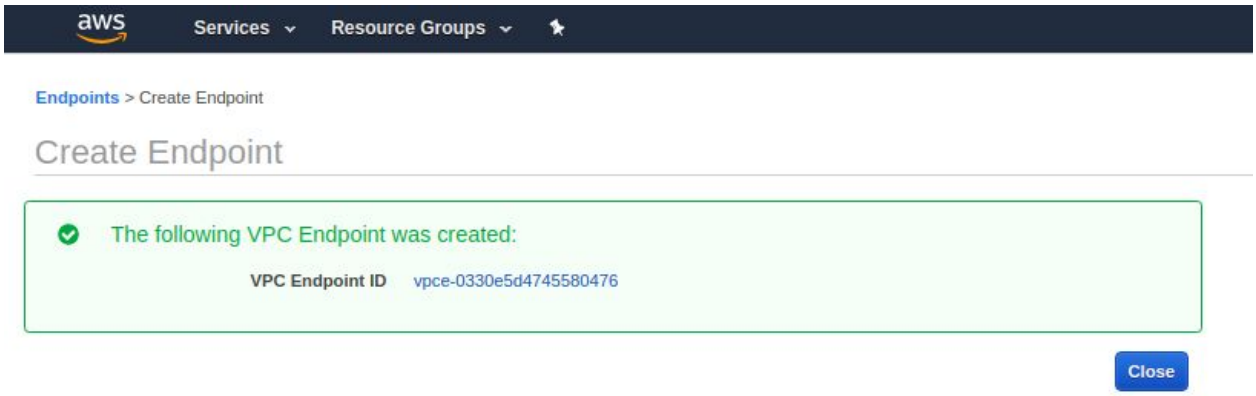
Then create the endpoint for s3 service



Then create the policy for access the s3



The endpoint of S3 has been created ....

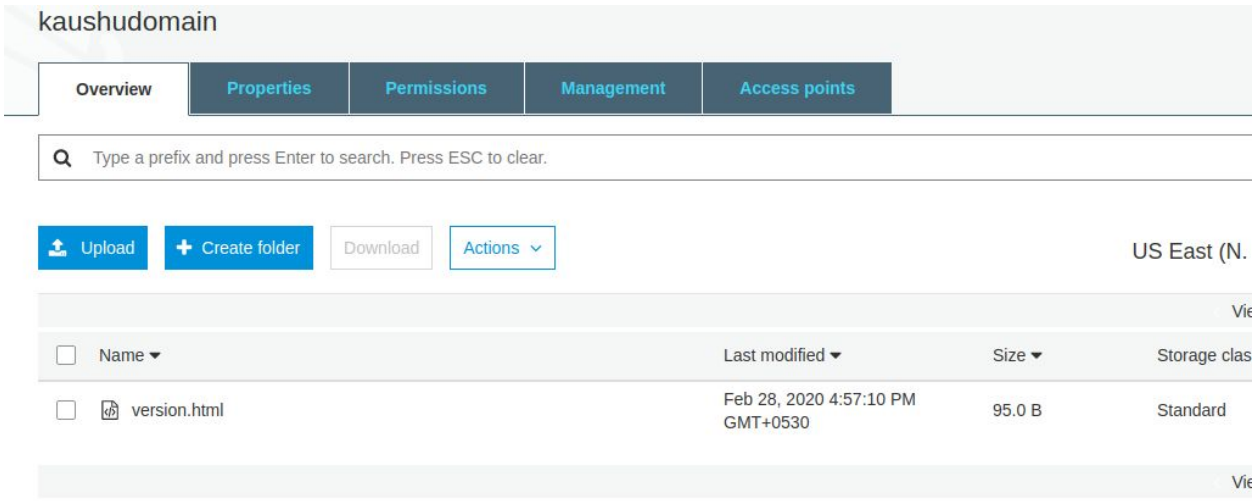


Ques 10. CORS, Enable CORS for 2 specific website.

Ans 10.

Enable CORS in Amazon API Gateway. You can now enable CORS (cross-origin resource sharing) with one click directly in the Amazon API Gateway console. CORS allows methods in API Gateway to request restricted resources from a different domain (e.g., a JavaScript client that calls an API deployed on a different domain).

Created a bucket and uploaded a html page



Then in the cross configuration added a new configuration