# Doubt Resolving

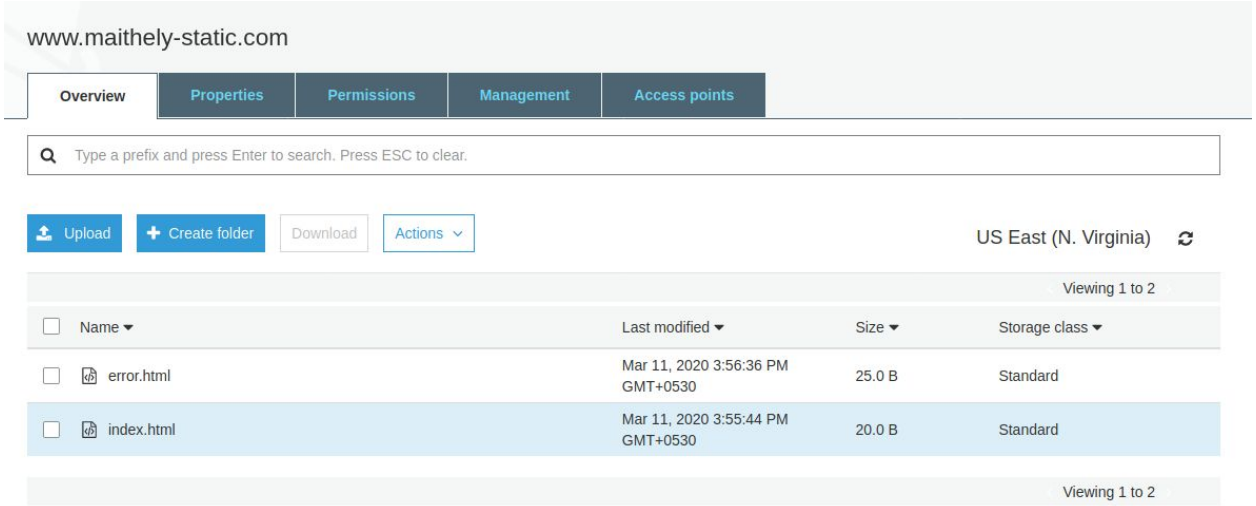
# ASSIGNMENT


Name – Maithely Sharma
College – University of Petroleum and Energy Studies
EmployeeID – 4057

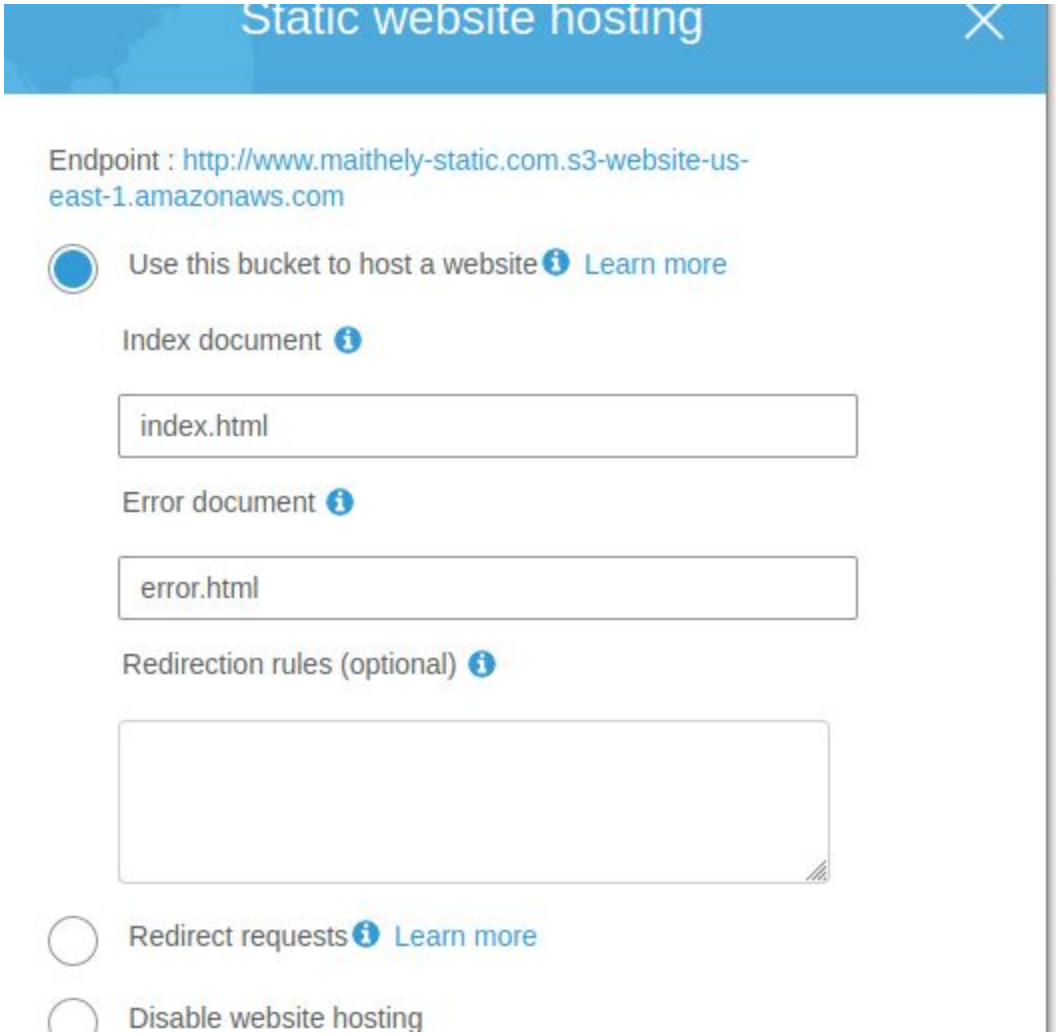
1. Static website hosting using s3(what is index and error page).
Firsty created two files : i) index.html , ii)error.html

```
maithely@maithely:~$ mkdir static
maithely@maithely:~$ cd static/
maithely@maithely:~/static$ sudo vim index.html
[sudo] password for maithely:
maithely@maithely:~/static$ cat index.html
index file created
maithely@maithely:~/static$ sudo vim error.html
maithely@maithely:~/static$ cat error.html
ERRROORRRRRRRRRRRRRRRRR
maithely@maithely:~/static$
```

Now create a bucket and upload two files created above

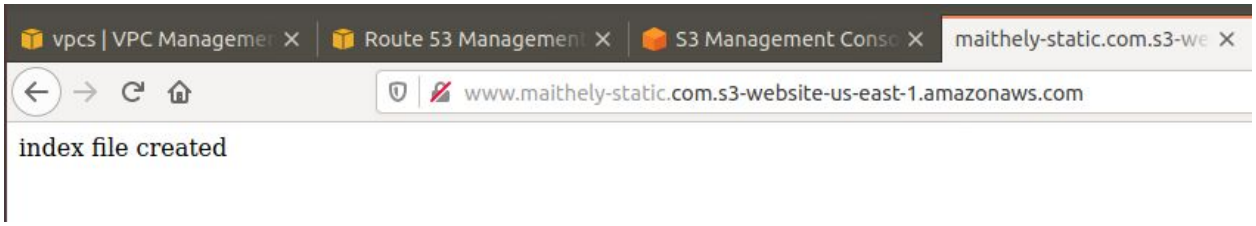


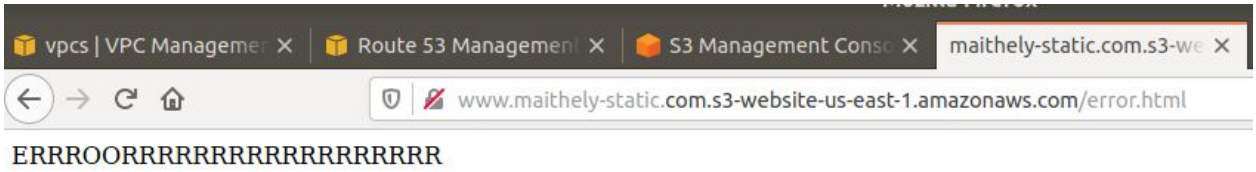

In properties select static hosting and enter the file names




Then save and again open dynamic hosting and you select the link

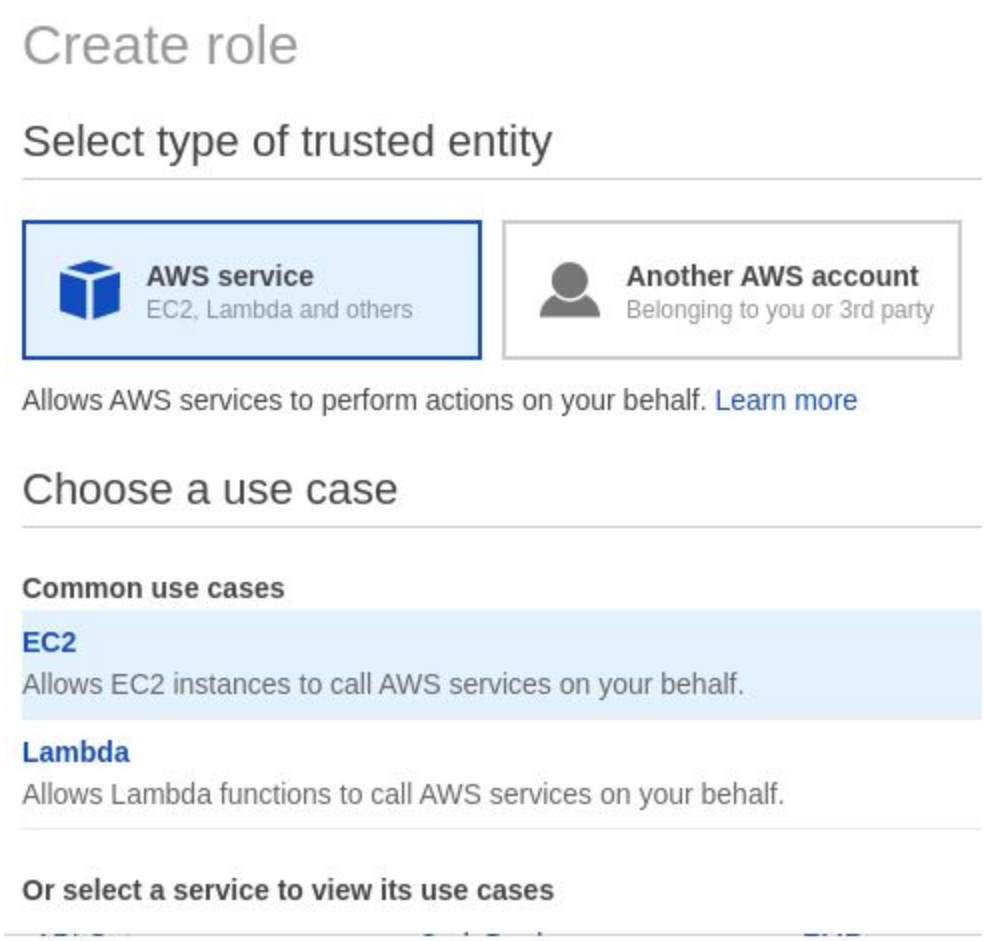

You can see the outputs as




2. Create an assume role to access s3 using ec2.

Create a Role with full access to S3



## Review

Provide the required information below and review this role before you create it.

**Role name*** Maithelyrole

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

**Role description** Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

**Trusted entities** AWS service: ec2.amazonaws.com

**Policies** Policies not attached

**Permissions boundary** Permissions boundary is not set

*** Required** Cancel Previous Create role

## Add permissions to Maithelyrole

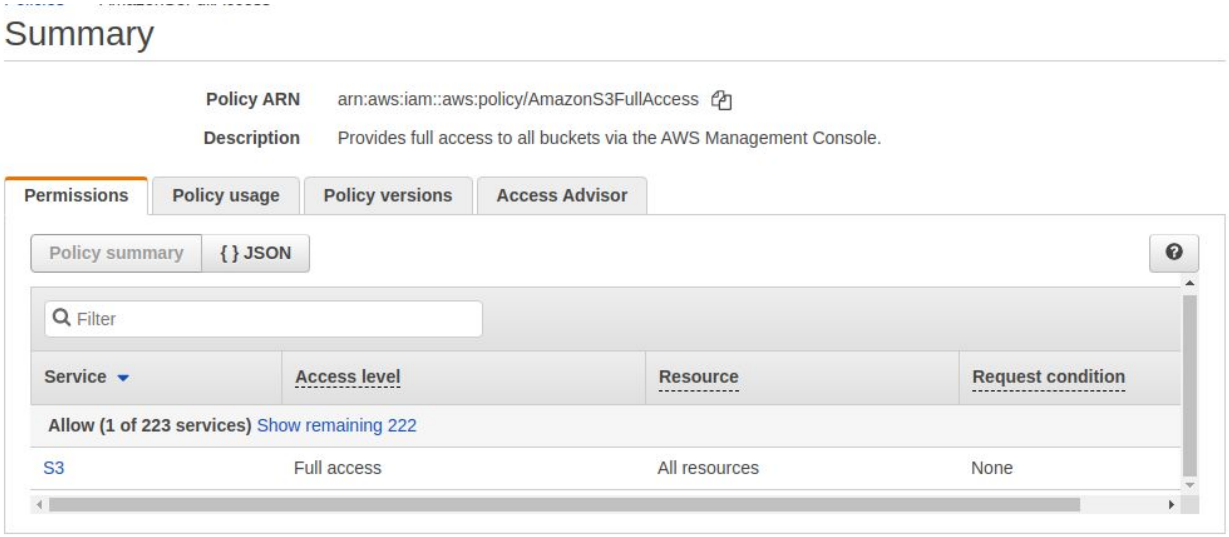
### Attach Permissions

Create policy

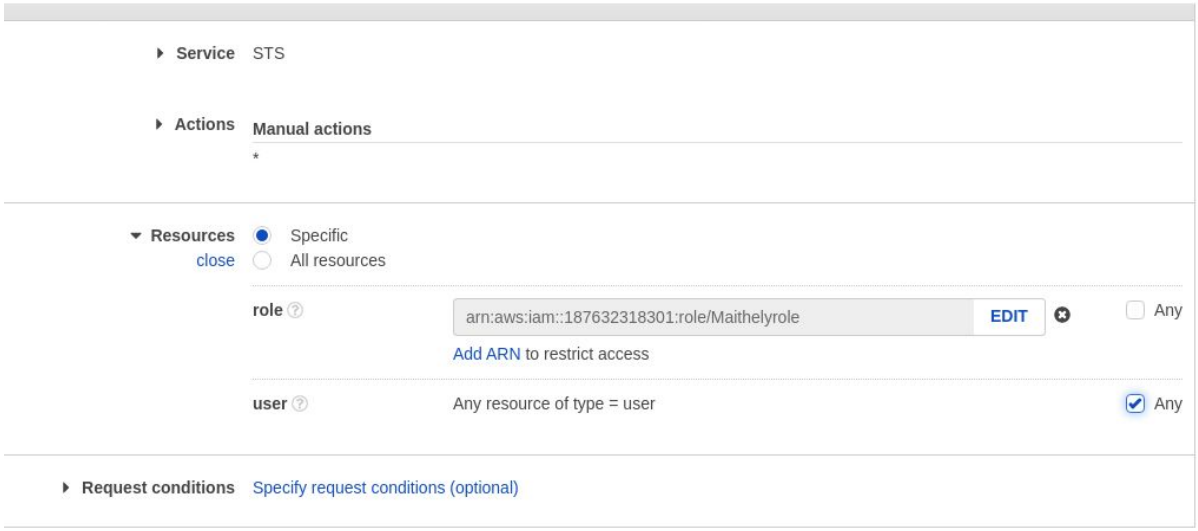
Filter policies s3 Showing 10 results

| Policy name | Type | Used as |
|---|---|---|
| AmazonDMSRedshiftS3Role | AWS managed | None |
| AmazonS3FullAccess | AWS managed | Permissions policy (29) |
| AmazonS3ReadOnlyAccess | AWS managed | Permissions policy (1) |
| AWSLambdaS3ExecutionRole-1a3ddce4-a989-4828-88a4-7f5e701af3ef | Customer managed | Permissions policy (1) |
| AWSLambdaS3ExecutionRole-34f3178e-e3ee-4238-8c64-0e27432978a2 | Customer managed | Permissions policy (1) |

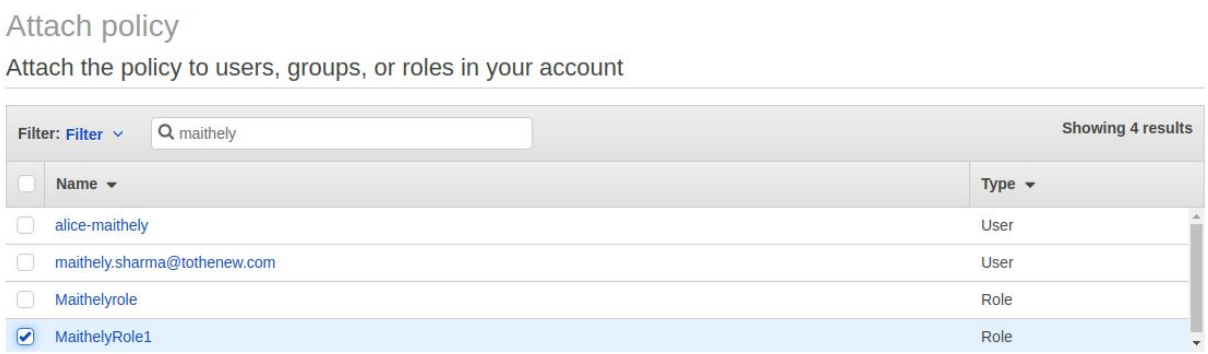
Cancel Attach policy



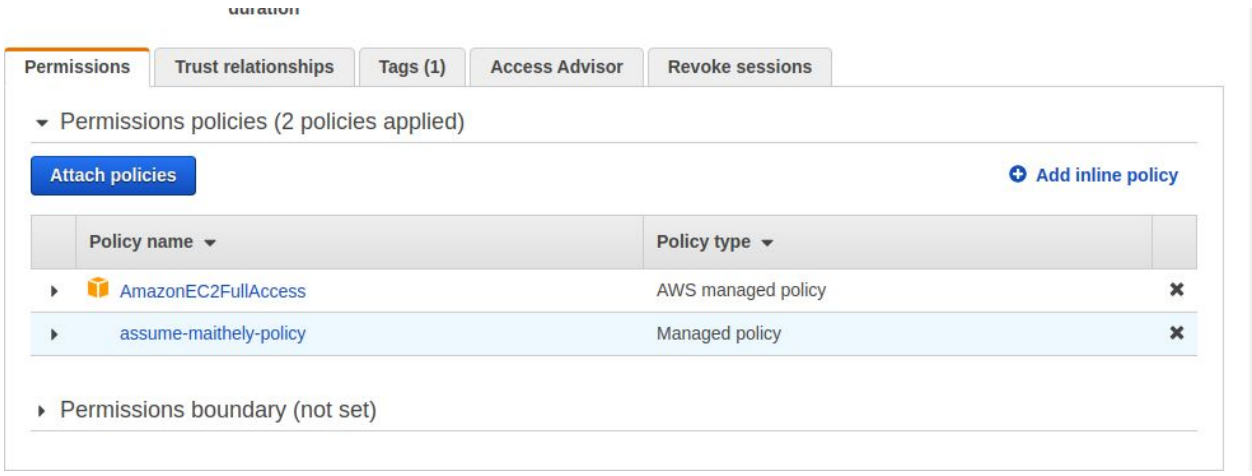

Create another role as Maithelyrole1 and attach policy of sts service to it



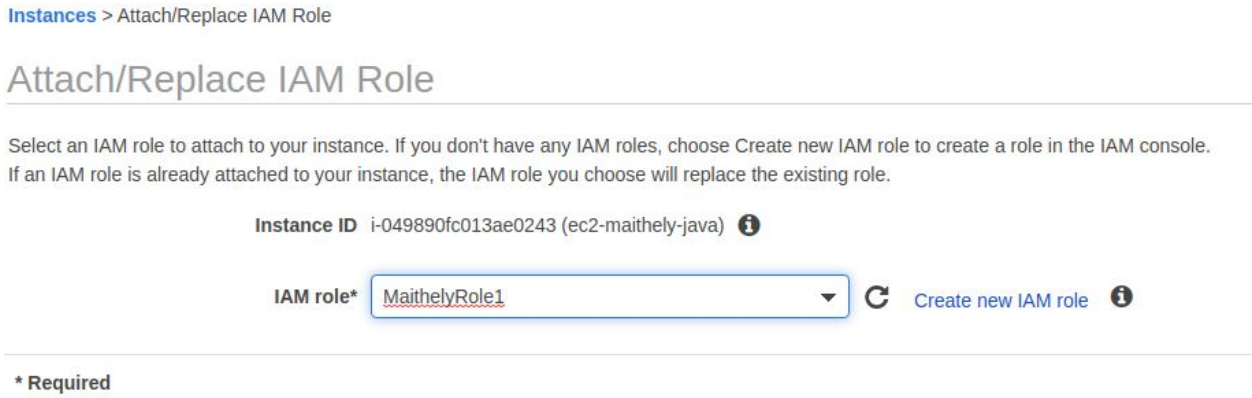

Now attach this policy to the new role created




Now you can see that assume role attached




Now create an ec2 instance and attach to the newrole created i.e Maithelyrole1



Now add the arn of new role i.e maithelyrole1 to old role i.e maithelyrole in trust relationship

## Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

**Policy Document**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS":"arn:aws:iam::187632318301:role/MaithelyRole1" ,
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Now you have to ssh to the instance created and update it. Also install aws cli

```
maithely@maithely:~$ ssh -i "maithelykeypair.pem" ubuntu@ec2-52-207-215-48.compute-1.amazonaws.com
Warning: Identity file maithelykeypair.pem not accessible: No such file or directory.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Feb 28 10:51:26 UTC 2020

  System load:  0.08              Processes:           86
  Usage of /:   16.5% of 7.69GB   Users logged in:     0
  Memory usage: 17%               IP address for eth0: 172.31.104.188
  Swap usage:   0%


54 packages can be updated.
32 updates are security updates.


Last login: Fri Feb 28 06:46:24 2020 from 182.71.160.186
ubuntu@ip-172-31-104-188:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [871 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1054 kB]
Fetched 2177 kB in 1s (2789 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-104-188:~$ sudo apt install awscli
Reading package lists... Done
Building dependency tree
```

3. Block s3 access on the basis of

i. IP

Now write this in bucket policy

```
182.71.160.186maithely@maithely:~$ curl ifconfig.me
61.12.91.218maithely@maithely:~$
```

www.maithely-static.com

Overview | Properties | **Permissions** | Management | Access points

Block public access | Access Control List | Bucket Policy | CORS configuration

## Bucket policy editor ARN: arn:aws:s3:::www.maithely-static.com

Type to add a new policy or edit an existing policy in the text area below.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::www.maithely-static.com/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "61.12.91.218/32"
                }
            }
        }
    ]
}
```

ii. Domain

## Bucket policy editor ARN: arn:aws:s3:::maithelybucket

Type to add a new policy or edit an existing policy in the text area below.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::maithelybucket/*",
            "Condition": {
                "StringLike": {
                    "aws:referer": [
                        "https://www.maithely.com/*",
                        "https://maithely.com/*"
                    ]
                }
            }
        }
    ]
}
```

iii. Pre-signed URL(Time based)

```
maithely@maithely:~$ aws s3 presign s3://www.maithely-static.com/error
.html --expires-in 120
https://s3.amazonaws.com/www.maithely-static.com/error.html?AWSAccessK
eyId=AKIASXL6B65OXQFOJAKS&Signature=yVuY2kVkpsEhSHM0nMewy3hJvm0%3D&Exp
ires=1583999311
maithely@maithely:~$
```

s3.amazonaws.com/www.maithely-static.com/error.html?AWSAccessKeyId=AKIASXL6B65OXQFOJAKS8

ERRROORRRRRRRRRRRRRRRRR

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

- **Parameter group:** You manage your DB engine configuration by associating your **DB instances with parameter groups**. Amazon RDS defines parameter groups with default settings that apply to newly created DB instances . You can define your own parameter groups with customized settings. Then you can modify your DB instances to use your own parameter groups.

  A DB parameter group acts as a container for engine configuration values that are applied to one or more DB instances.

  If you create a DB instance without specifying a DB parameter group, the DB instance uses a default DB parameter group. Each default DB parameter group contains database engine defaults and Amazon RDS system defaults based on the engine, compute class, and allocated storage of the instance. You can't modify the parameter settings of a default parameter group. Instead, you create your own parameter group where you choose your own parameter settings. Not all DB engine parameters can be changed in a parameter group that you create.

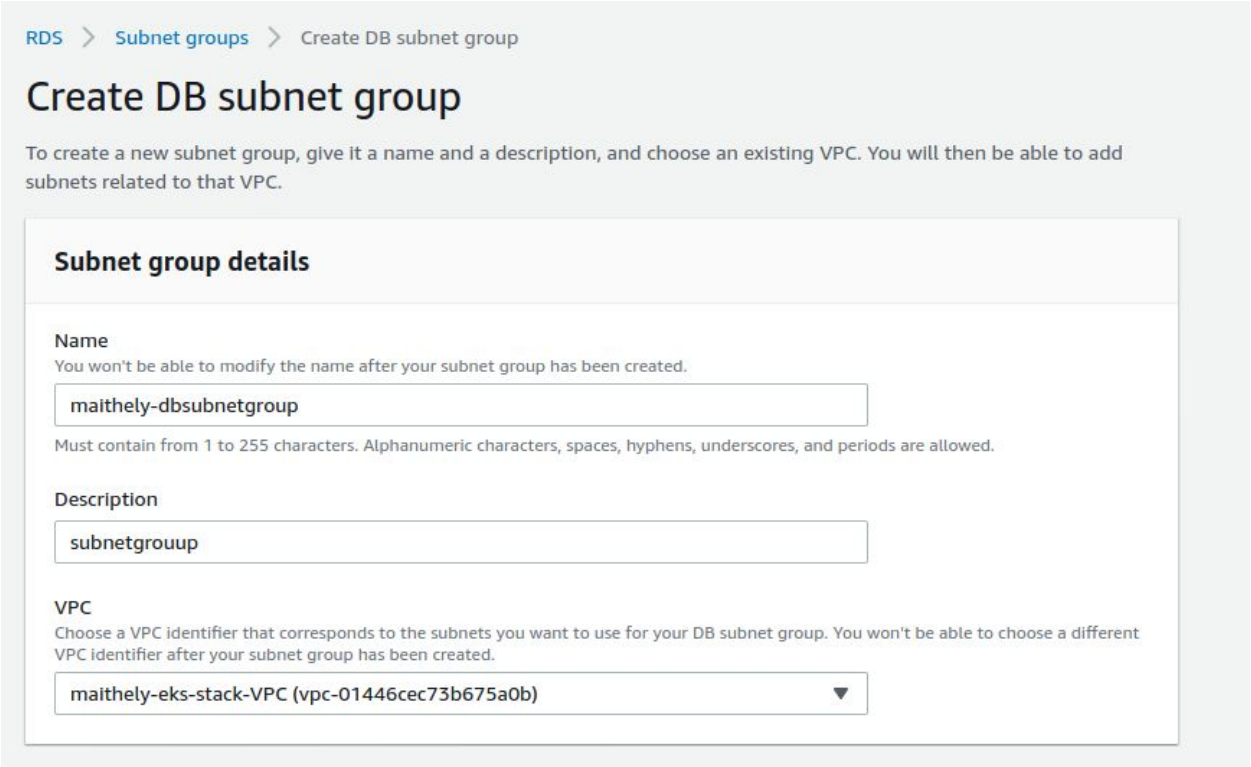
  If you want to use your own parameter group, you create a new parameter group and modify the parameters that you want to. You then modify your DB instance to use the new parameter group. If you update parameters within a DB parameter group, the changes apply to all DB instances that are associated with that parameter group.

- **Option group:** Some DB engines offer additional features that make it easier to manage data and databases, and to provide additional security for your database. 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. Options can have settings that specify how the option works. When you associate a DB instance with an option group, the specified options and option settings are enabled for that DB instance.

  Amazon RDS supports options for the following database engines:

| Database Engine | Relevant Documentation |
| --- | --- |
| MariaDB | Options for MariaDB Database Engine |
| Microsoft SQL Server | Options for the Microsoft SQL Server Database Engine |
| MySQL | Options for MySQL DB Instances |
| Oracle | Options for Oracle DB Instances |

Firstly create a subnet group under RDS service



## Add subnets

Add subnet(s) to this subnet group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Add all the subnets related to this VPC

Availability zone

us-east-1c

Subnet

subnet-05128b98c1ea54979 (192.168.192.0/18)

Add subnet

### Subnets in this subnet group (3)

| Availability zone | Subnet ID | CIDR block | Action |
|---|---|---|---|
| us-east-1a | subnet-05ccb7f834d214a5a | 192.168.64.0/18 | Remove |
| us-east-1b | subnet-02618d516e069dda9 | 192.168.128.0/18 | Remove |
| us-east-1c | subnet-05128b98c1ea54979 | 192.168.192.0/18 | Remove |

Now create a database (My-sql)

# Create database

## Choose a database creation method Info

- (●) **Standard Create**
  You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- ( ) **Easy Create**
  Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

## Engine options

Engine type Info

- ( ) Amazon Aurora
- (●) MySQL
- ( ) MariaDB

## Settings

**DB instance identifier** Info

Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

database-2-maithely

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

### ▼ Credentials Settings

**Master username** Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

**Master password** Info

••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

**Confirm password** Info

••••••••

## DB instance size

DB instance class Info

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

- Standard classes (includes m classes)
- Memory Optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized

Include previous generation classes

## Storage

Storage type Info

General Purpose (SSD)

Allocated storage

20 GiB

(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage may improve IOPS performance.

## Storage autoscaling Info

Provides dynamic scaling support for your database's storage based on your application's needs.

☑ **Enable storage autoscaling**

Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

**Maximum storage threshold** Info

Charges will apply when your database autoscales to the specified threshold

1000 GiB

Minimum: 21 GiB, Maximum: 16384 GiB

## Availability & durability

**Multi-AZ deployment** Info

- Do not create a standby instance
- Create a standby instance (recommended for production usage)

  Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

## Connectivity

Virtual Private Cloud (VPC) Info

VPC that defines the virtual networking environment for this DB instance.

maithely-eks-stack-VPC (vpc-01446cec73b675a0b)

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

### Additional connectivity configuration

Subnet group Info

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

maithely-dbsubnetgroup

Publicly accessible Info

- (●) Yes
  Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.
- ( ) No
  RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

- (●) Choose existing
  Choose existing VPC security groups
- ( ) Create new
  Create new VPC security group

Existing VPC security groups

*Choose VPC security groups*

maithely-eks-stack-ControlPlaneSecurityGroup-12OKBW6DIGY67 ×

default ×

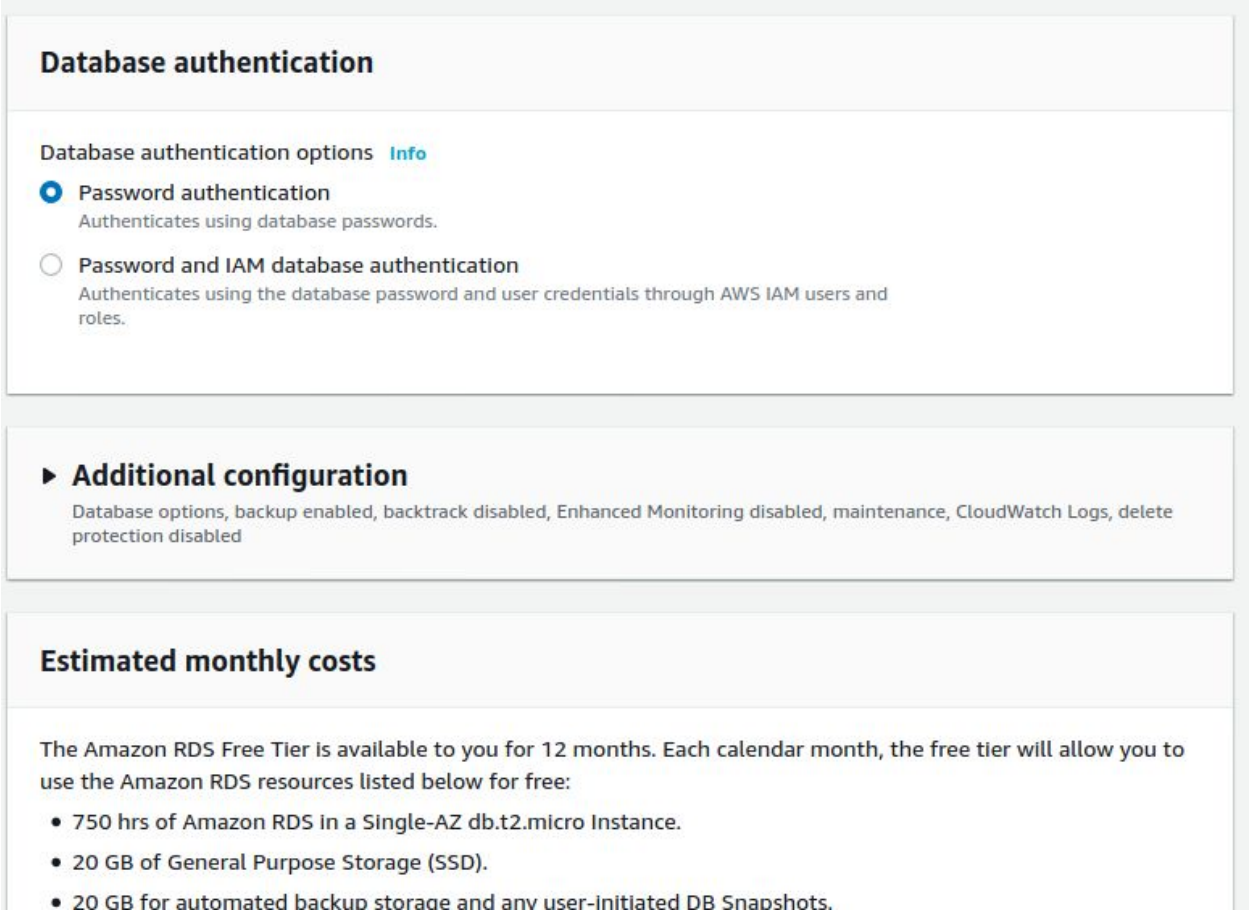
Availability zone Info

No preference

Database port Info

TCP/IP port the database will use for application connections.

3306




5. ACL, Bucket policy, IAM Policy.

**Amazon S3 access control lists (ACLs)** enable you to manage access to buckets and objects. Each bucket and object has an ACL attached to it as a subresource. It defines which AWS accounts or groups are granted access and the type of access. When a request is received against a resource, Amazon S3 checks the corresponding ACL to verify that the requester has the necessary access permissions.

When you create a bucket or an object, Amazon S3 creates a default ACL that grants the resource owner full control over the resource.

A grantee can be an AWS account or one of the predefined Amazon S3 groups. You grant permission to an AWS account using the email address or the canonical user ID.

**IAM policies** specify what actions are allowed or denied on what AWS resources (e.g. allow ec2:TerminateInstance on the EC2 instance with instance_id=i-8b3620ec). You attach IAM policies to IAM users, groups, or

roles, which are then subject to the permissions you've defined. In other words, IAM policies define what a principal can do in your AWS environment.

**S3 bucket policies**, on the other hand, are attached only to S3 buckets. S3 bucket policies specify what actions are allowed or denied for which principals on the bucket that the bucket policy is attached to (e.g. allow user Alice to PUT but not DELETE objects in the bucket). S3 bucket policies are a type of access control list, or ACL (here I mean "ACL" in the generic sense, not to be confused with S3 ACLs, which is a separate S3 feature discussed later in this post).

6. Mount S3 to an EC2 instance.

```
ubuntu@ip-172-31-61-252:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InR
elease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-upd
ates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-bac
kports InRelease [74.6 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/uni
verse amd64 Packages [8570 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelea
se [88.7 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/uni
verse Translation-en [4941 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mul
tiverse amd64 Packages [151 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/mul
tiverse Translation-en [108 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-upd
ates/main amd64 Packages [872 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-up
dates/main Translation-en [303 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-up
dates/restricted amd64 Packages [32.9 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-up
dates/restricted Translation-en [8468 B]
```

```
ubuntu@ip-172-31-61-252:~$ sudo apt-get install automake autoto
ols-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev
 libxml2-dev make pkg-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
fuse is already the newest version (2.9.7-1ubuntu1).
fuse set to manually installed.
git is already the newest version (1:2.17.1-1ubuntu0.5).
git set to manually installed.
The following additional packages will be installed:
  autoconf binutils binutils-common binutils-x86-64-linux-gnu
  build-essential cpp cpp-7 dpkg-dev fakeroot g++-7 gcc gcc-7
  gcc-7-base gcc-8-base gir1.2-harfbuzz-0.0 icu-devtools
  libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan4 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libdpkg-perl
  libfakeroot libfile-fcntllock-perl libgcc-7-dev libgcc1
  libglib2.0-bin libglib2.0-dev libglib2.0-dev-bin libgomp1
```

```
ubuntu@ip-172-31-61-252:~$ git clone https://github.com/s3fs-fu
se/s3fs-fuse.git
Cloning into 's3fs-fuse'...
remote: Enumerating objects: 5879, done.
remote: Total 5879 (delta 0), reused 0 (delta 0), pack-reused 5
879
Receiving objects: 100% (5879/5879), 3.46 MiB | 24.13 MiB/s, do
ne.
Resolving deltas: 100% (4079/4079), done.
```

```
ubuntu@ip-172-31-61-252:~$ cd s3fs-fuse
ubuntu@ip-172-31-61-252:~/s3fs-fuse$ ./autogen.sh
--- Make commit hash file -------
--- Finished commit hash file ---
--- Start autotools -------------
--- Finished autotools ----------
ubuntu@ip-172-31-61-252:~/s3fs-fuse$ ./configure --prefix=/usr
--with-openssl
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for g++... g++
```

```
ubuntu@ip-172-31-61-252:~/s3fs-fuse$ make
make  all-recursive
make[1]: Entering directory '/home/ubuntu/s3fs-fuse'
Making all in src
make[2]: Entering directory '/home/ubuntu/s3fs-fuse/src'
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/inc
lude/fuse -I/usr/include/x86_64-linux-gnu -I/usr/include/libxml
2    -g -O2 -Wall -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -M
T s3fs.o -MD -MP -MF .deps/s3fs.Tpo -c -o s3fs.o s3fs.cpp
mv -f .deps/s3fs.Tpo .deps/s3fs.Po
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/inc
lude/fuse -I/usr/include/x86_64-linux-gnu -I/usr/include/libxml
2    -g -O2 -Wall -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -M
T curl.o -MD -MP -MF .deps/curl.Tpo -c -o curl.o curl.cpp
mv -f .deps/curl.Tpo .deps/curl.Po
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/inc
lude/fuse -I/usr/include/x86_64-linux-gnu -I/usr/include/libxml
```

```
ubuntu@ip-172-31-61-252:~/s3fs-fuse$ sudo make install
Making install in src
make[1]: Entering directory '/home/ubuntu/s3fs-fuse/src'
make[2]: Entering directory '/home/ubuntu/s3fs-fuse/src'
 /bin/mkdir -p '/usr/bin'
  /usr/bin/install -c s3fs '/usr/bin'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ubuntu/s3fs-fuse/src'
make[1]: Leaving directory '/home/ubuntu/s3fs-fuse/src'
Making install in test
make[1]: Entering directory '/home/ubuntu/s3fs-fuse/test'
make[2]: Entering directory '/home/ubuntu/s3fs-fuse/test'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ubuntu/s3fs-fuse/test'
make[1]: Leaving directory '/home/ubuntu/s3fs-fuse/test'
Making install in doc
```

Now for credential purposes install and configure awscli

```
ubuntu@ip-172-31-61-252:~$ sudo apt install awscli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docutils-common libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2
  libpaper-utils libpaper1 libtiff5 libwebp6 libwebpdemux2
  libwebpmux3 python3-botocore python3-dateutil
  python3-docutils python3-jmespath python3-olefile
  python3-pil python3-pygments python3-roman python3-rsa
  python3-s3transfer sgml-base xml-core
Suggested packages:
  liblcms2-utils docutils-doc fonts-linuxlibertine
  | ttf-linux-libertine texlive-lang-french
  texlive-latex-base texlive-latex-recommended python-pil-doc
```

```
ubuntu@ip-172-31-61-252:~$ aws configure
AWS Access Key ID [None]:  AKIASXL6B65OXQFOJAKS
AWS Secret Access Key [None]: wDZf2P6wG5u735RLIcIbpWPioCz4OKhsr
Bisy8XW
Default region name [None]:
Default output format [None]:
```

Create a new file in /etc with the name passwd-s3fs and Paste the access key and secret key in the below format .

```
ubuntu@ip-172-31-61-252:~$ sudo  touch /etc/passwd-s3fs
ubuntu@ip-172-31-61-252:~$ sudo vim /etc/passwd-s3fs
ubuntu@ip-172-31-61-252:~$ sudo chmod 640 /etc/passwd-s3fs
```

```
Syntax
Your_accesskey:Your_secretkeycar
```

```
ubuntu@ip-172-31-61-252:/mybucket$ sudo cat /etc/passwd-s3fs
AKIASXL6B65OXQFOJAKS:wDZf2P6wG5u735RLIcIbpWPioCz4OKhsrBisy8XW
ubuntu@ip-172-31-61-252:/mybucket$
```

Now create a directory or provide the path of an existing directory and mount S3bucket in it.

```
ubuntu@ip-172-31-61-252:~$ mkdir /mybucket
mkdir: cannot create directory '/mybucket': Permission denied
ubuntu@ip-172-31-61-252:~$ sudo !!
sudo mkdir /mybucket
```

```
ubuntu@ip-172-31-61-252:~$ s3fs maithelybucket -o use_cache=/tm
p -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5
/mybucket
s3fs: MOUNTPOINT: /mybucket permission denied.
ubuntu@ip-172-31-61-252:~$ sudo !!
sudo s3fs maithelybucket -o use_cache=/tmp -o allow_other -o ui
d=1001 -o mp_umask=002 -o multireq_max=5 /mybucket
ubuntu@ip-172-31-61-252:~$ which s3fs
/usr/bin/s3fs
```

```
ubuntu@ip-172-31-61-252:~$ which s3fs
/usr/bin/s3fs
ubuntu@ip-172-31-61-252:~$ sudo vim /etc/rc.local
```

```
ubuntu@ip-172-31-61-252:/mybucket$ cat /etc/rc.local
/usr/local/bin/s3fs maithelybucket -o use_cache=/tmp -o allow_o
ther -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mybucket

ubuntu@ip-172-31-61-252:/mybucket$
```

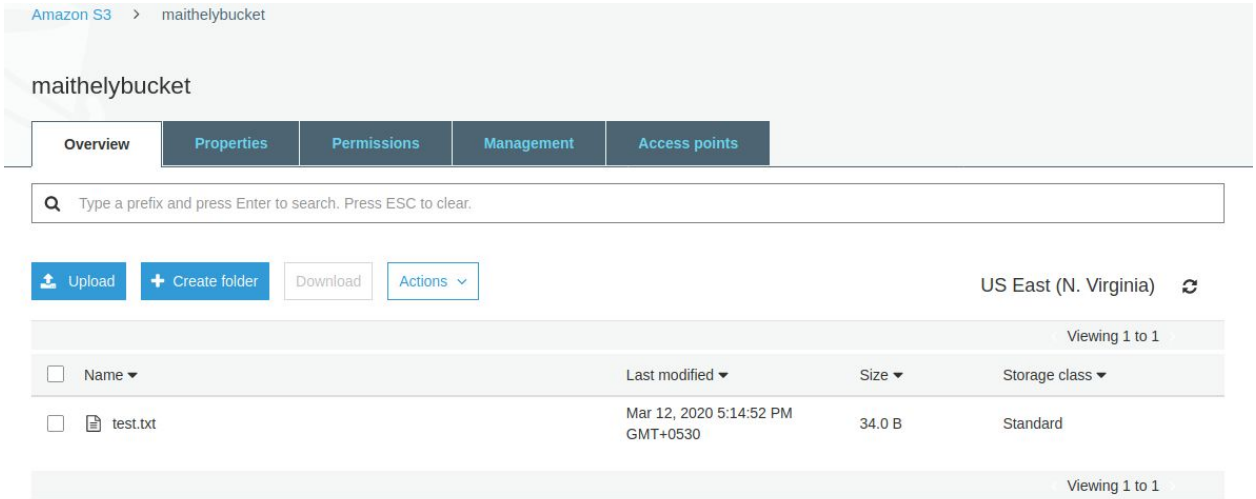
Check mounted s3 bucket

```
ubuntu@ip-172-31-61-252:~$ df -Th /mybucket
Filesystem     Type       Size  Used Avail Use% Mounted on
s3fs           fuse.s3fs  256T     0  256T   0% /mybucket
```

If it shows the mounted file system, you have successfully mounted the S3 bucket on your EC2 Instance. You can also test it further by creating a test file.

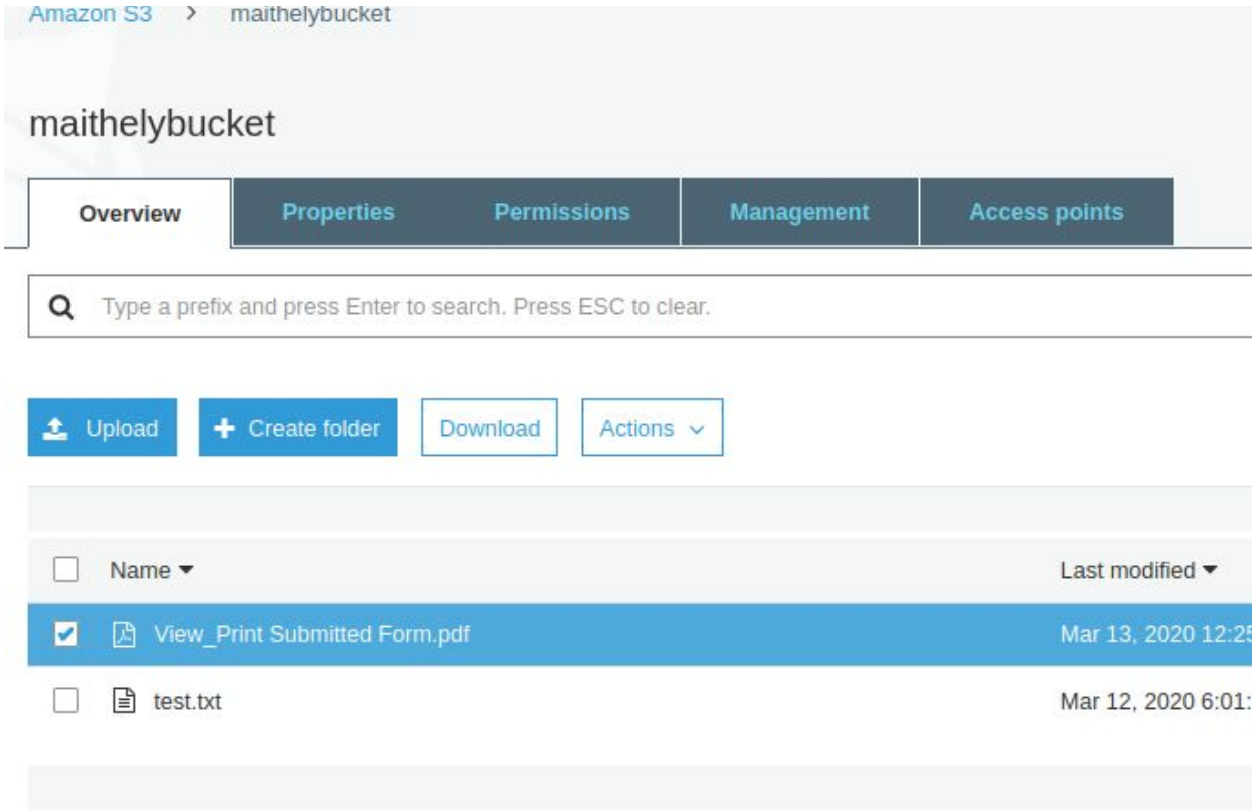
```
ubuntu@ip-172-31-61-252:/mybucket$ sudo vim test.txt
ubuntu@ip-172-31-61-252:/mybucket$ cat test.txt
this is a test file to check s3fs
```

You can see test.txt has been created in the bucket

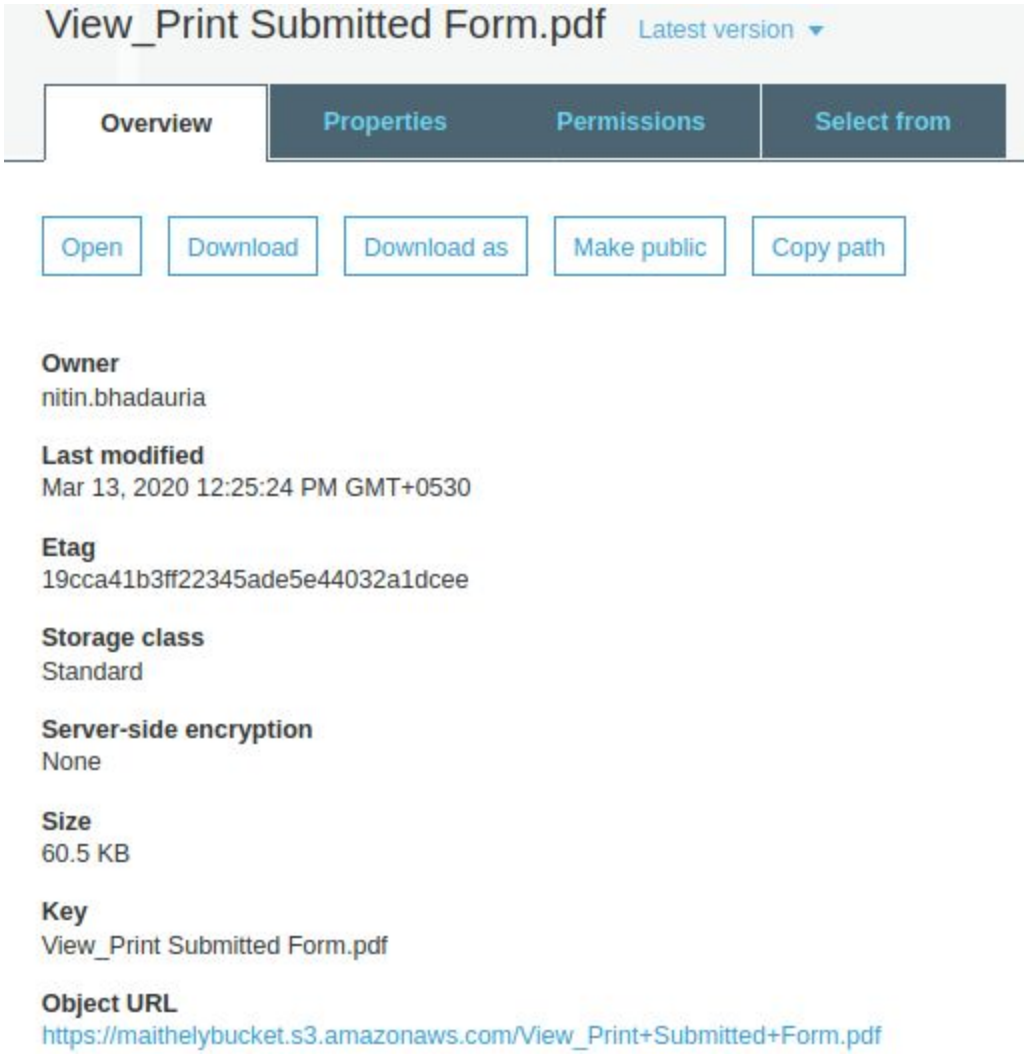


7. Change content type using s3

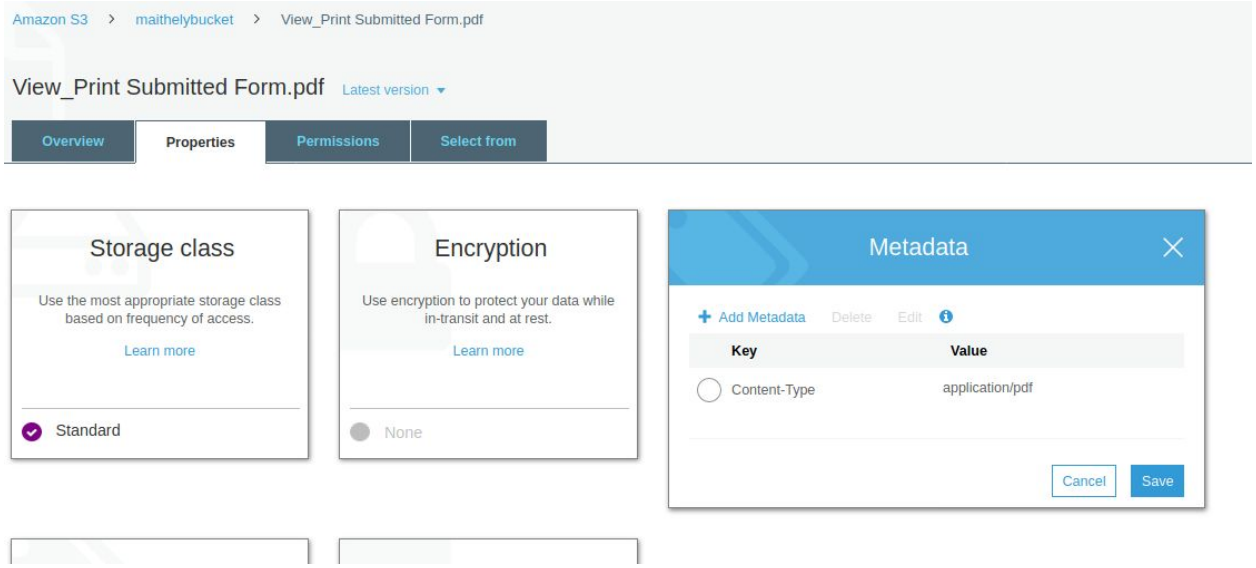
Add an object to the bucket




Now click on the object and access the object url

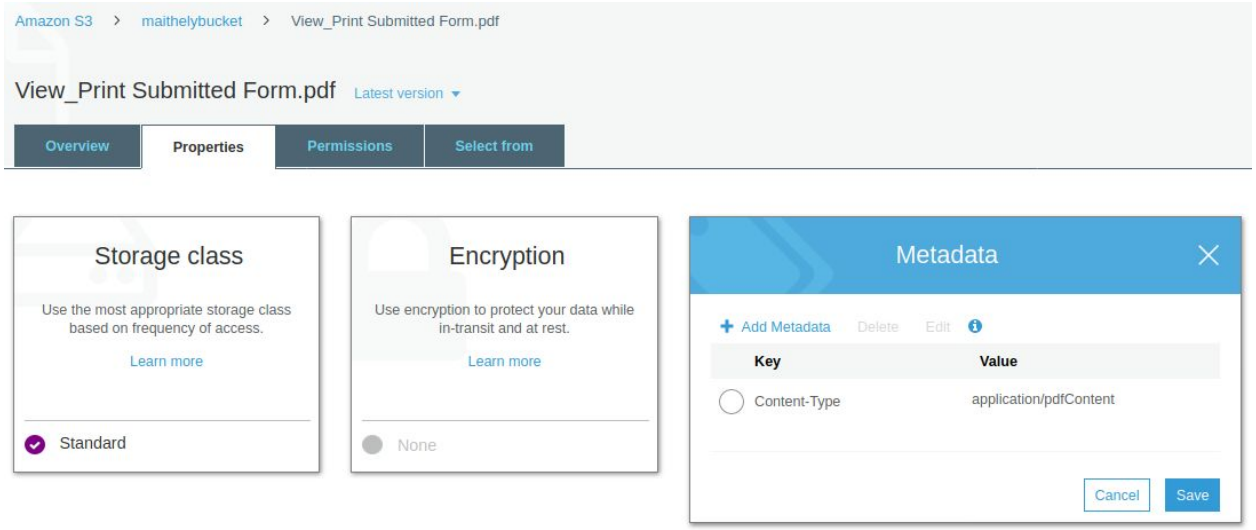



For rendering on the browser:
Go in properties and choose metadata where you have content-type as application/pdf

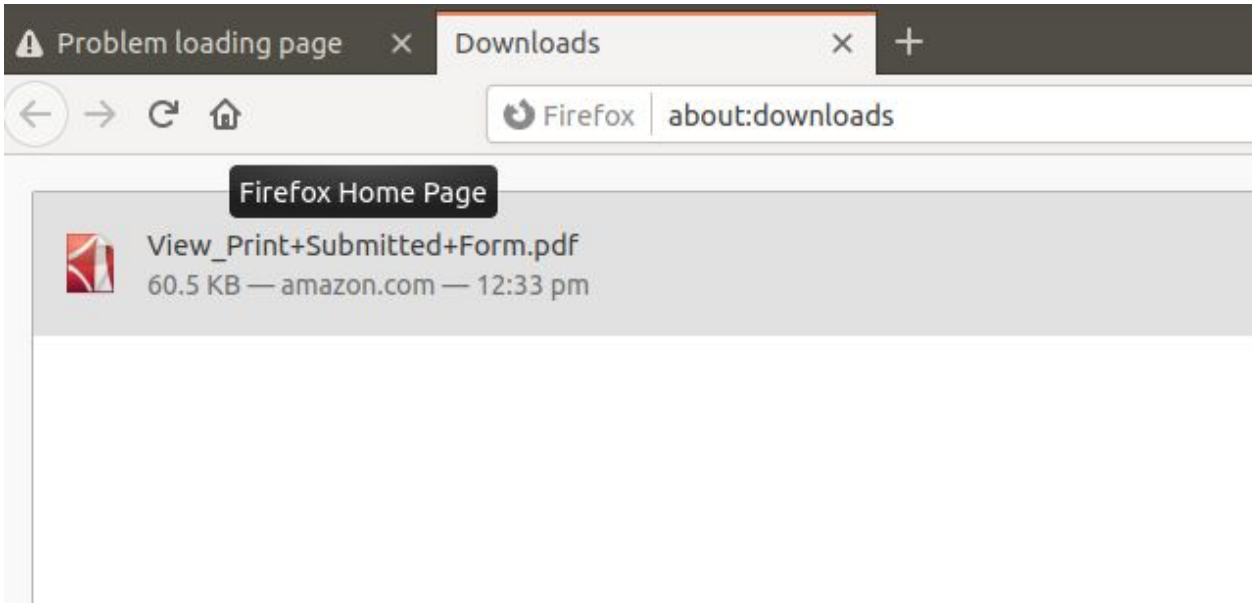



Here you can see the file

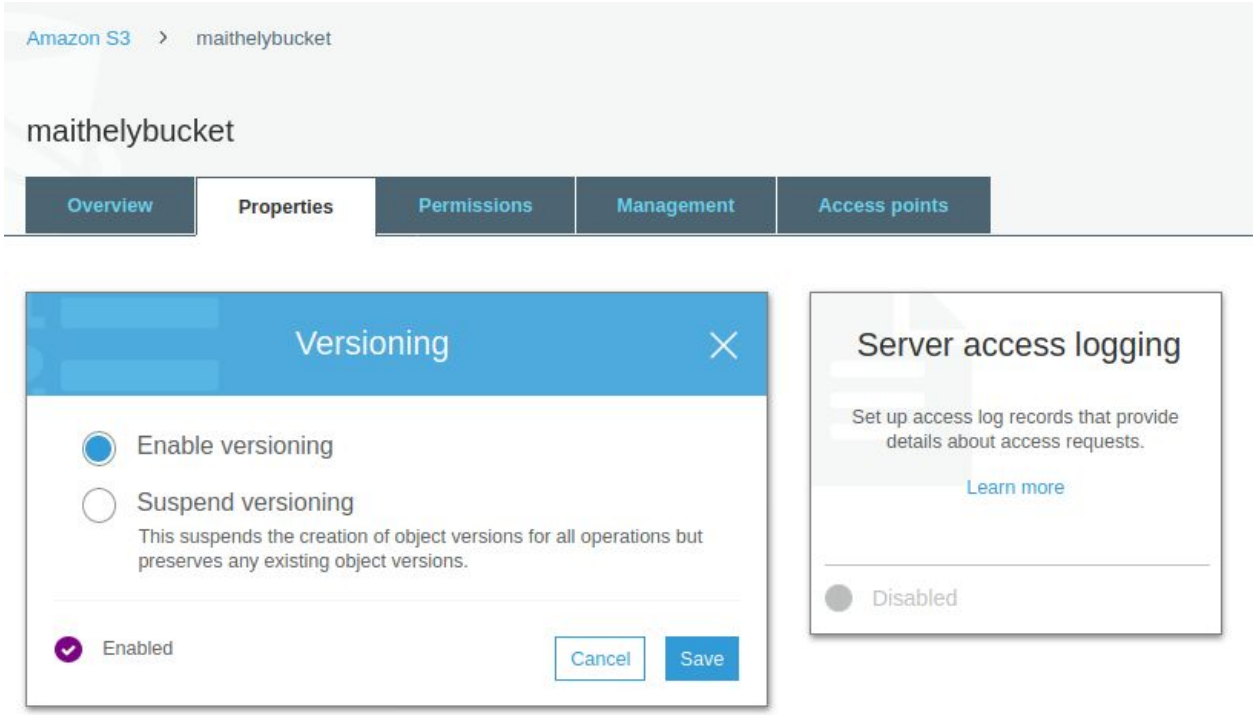

For downloading :
Go in properties and choose metadata where you have content-type as application/pdfContent



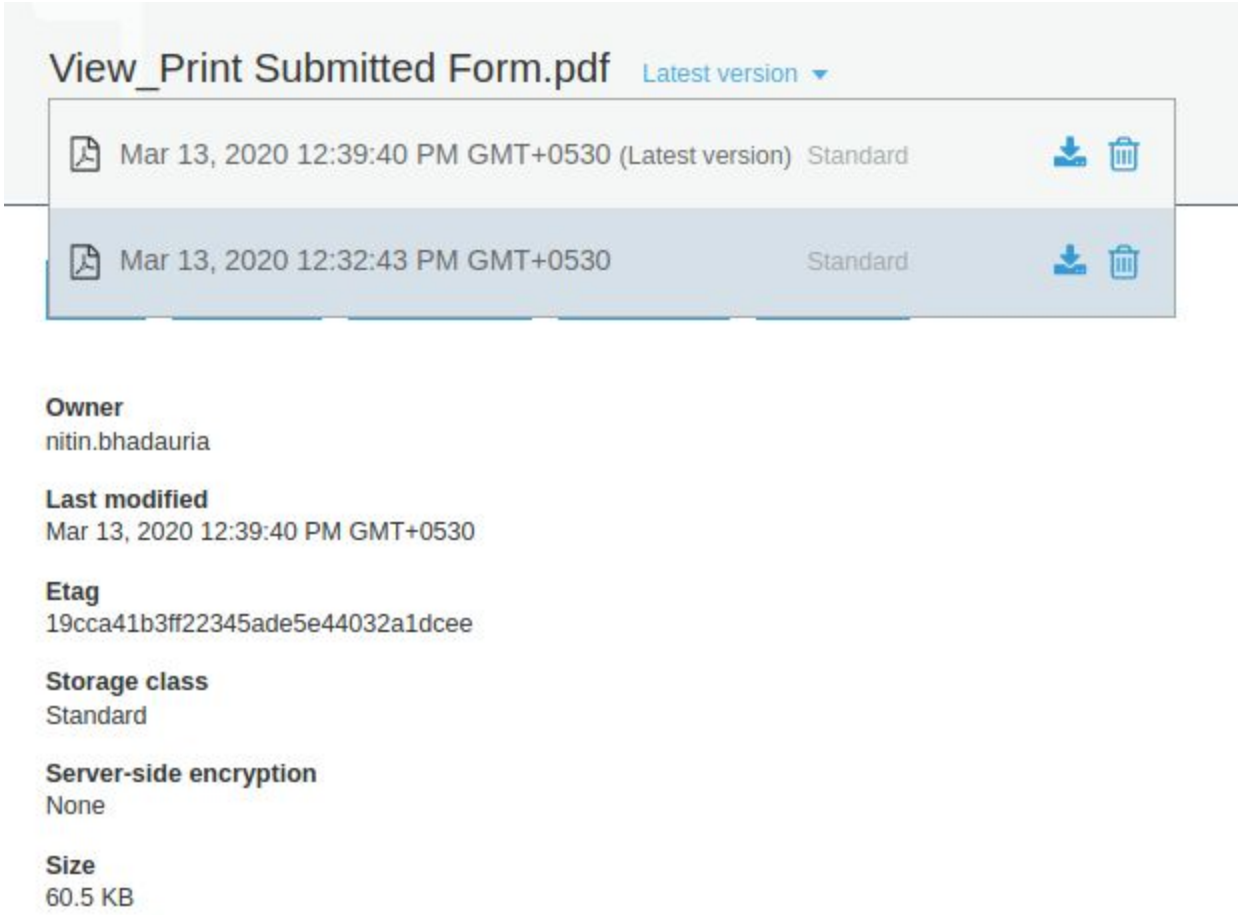

Here the file gets downloaded




8. Retrieve previous version of S3(enable versioning).




Upload the same file and you could see the versions




9. S3 VPC endpoint.

Create a VPC with subnets.

From the navigation pane, choose endpoints, then create endpoints.

aws Services Resource Groups maithely.sharma@tothenew.co...

Cloud

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections

Create Endpoint | Actions

Filter by tags and attributes or search by keyword

| Name | Endpoint ID | VPC ID | Service name |
|---|---|---|---|
| | vpce-0326435ae0... | vpc-0faa7c85885... | com.amazonaws.us-east-1 |
| | vpce-084d36ce5df... | vpc-d38d68b7 \| d... | com.amazonaws.us-east-1 |
| | vpce-0992a50134... | vpc-d38d68b7 \| d... | com.amazonaws.us-east-1 |

# Create Endpoint

A VPC endpoint allows you to securely connect your VPC to another service.
An interface endpoint is powered by PrivateLink, and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.
A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

**Service category**
- (●) AWS services
- ( ) Find service by name
- ( ) Your AWS Marketplace services

**Service Name** com.amazonaws.us-east-1.s3

search : s3 | Add filter

| Service Name | Owner | Type |
|---|---|---|
| com.amazonaws.us-east-1.s3 | amazon | Gateway |

**VPC*** vpc-0847695ca84e79af3

**Configure route tables** A rule with destination **pl-63a5400a (com.amazonaws.us-east-1.s3)** and a target with this endpoints' ID (e.g. vpce-12345678) will be added to the route tables you select below.

Subnets associated with selected route tables will be able to access this endpoint.

rtb-0ec3f8525a87f4860 rtb-02f1a06d9ef24a280

| | Route Table ID | Main | Associated With |
|---|---|---|---|
| ☑ | rtb-0ec3f8525a87f4860 | No | 3 subnets |
| ☑ | rtb-02f1a06d9ef24a280 | Yes | subnet-0e85ed72871e50725 \| kaushu_tag |

**Warning**
When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.

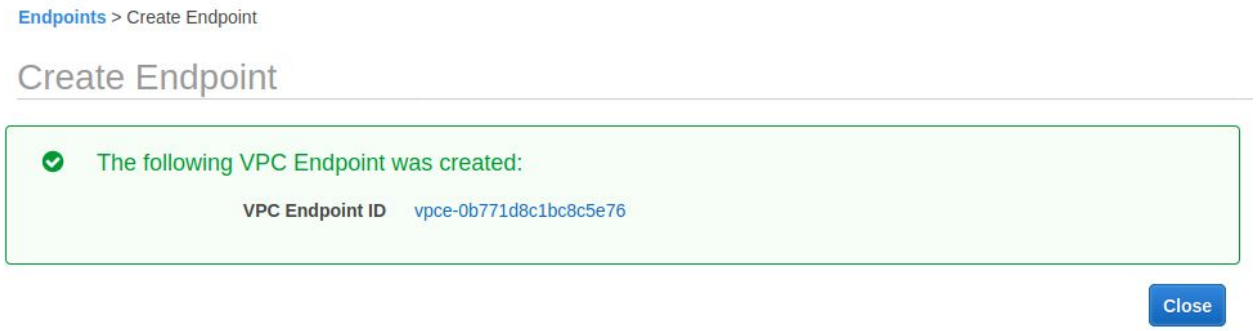
**Policy***

- (●) Full Access - Allow access by any user or service within the VPC using credentials from any AWS accounts to any resources in this AWS service. All policies — IAM user policies, VPC endpoint policies, and AWS service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must grant the necessary permissions for access to succeed.
- ( ) Custom

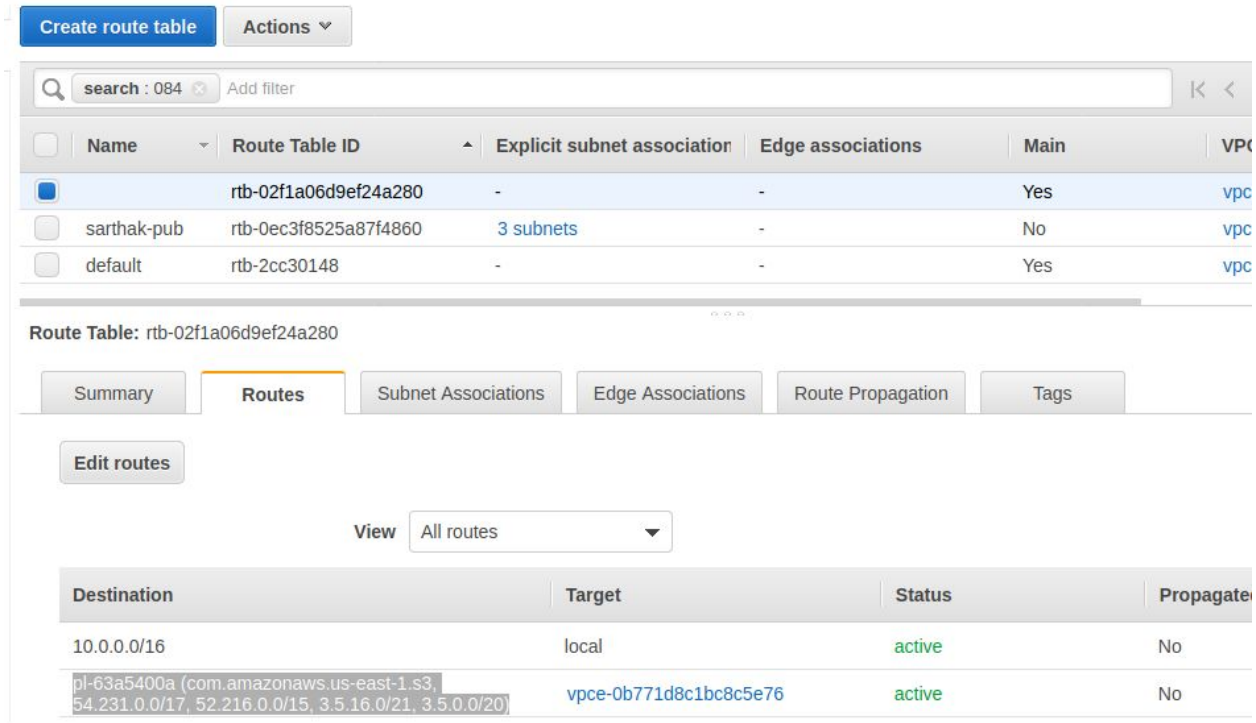
Use the policy creation tool to generate a policy, then paste the generated policy below.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

| **Key** (128 characters maximum) | **Value** (256 characters maximum) |
|---|---|
| owner | maithely |




You can see in route table of that subnet that endpoint has been added to it




Then Generate a policy and add it to the bucket policy.

- Add Bucket ARN
- Add conditions: StringEquals
- Add VPCE ID

## AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.

### Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

**Select Type of Policy** VPC Endpoint Policy

### Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

**Effect** Allow Deny

**Principal** *

Use a comma to separate multiple values.

**AWS Service** Amazon S3 All Services ('*')

Use multiple statements to add permissions for more than one service.

**Actions** -- Select Actions -- All Actions ('*')

Use multiple statements to add permissions for more than one service.

**Actions** -- Select Actions -- All Actions ('*')

**Amazon Resource Name (ARN)** arn:aws:s3:::maithelybucket

ARN should follow the following format: arn:aws:s3:::<bucket_name>/<key_name>.
Use a comma to separate multiple values.

Add Conditions (Optional) Hide

Conditions are any restrictions or details about the statement.(More Details).

**Condition** StringEquals

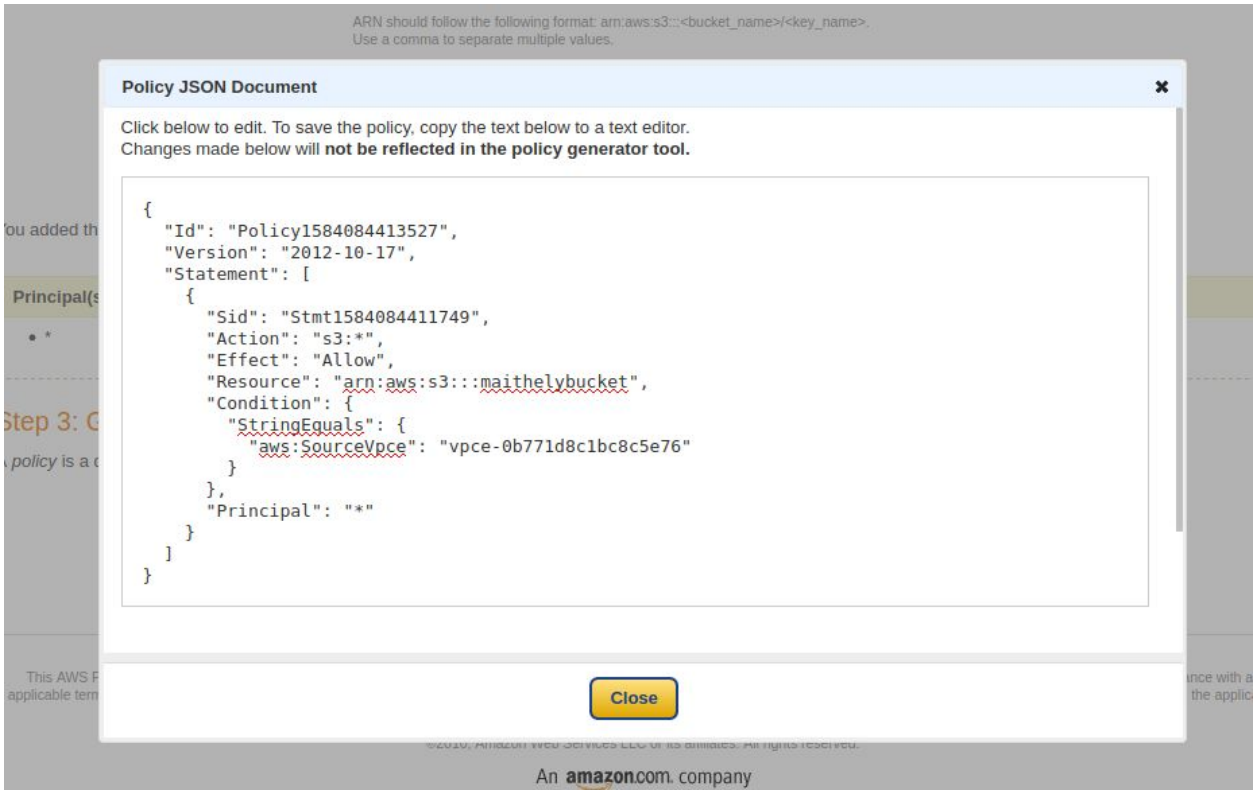
**Key** aws:SourceVpce

**Value** vpce-0b771d8c1bc8c5e76

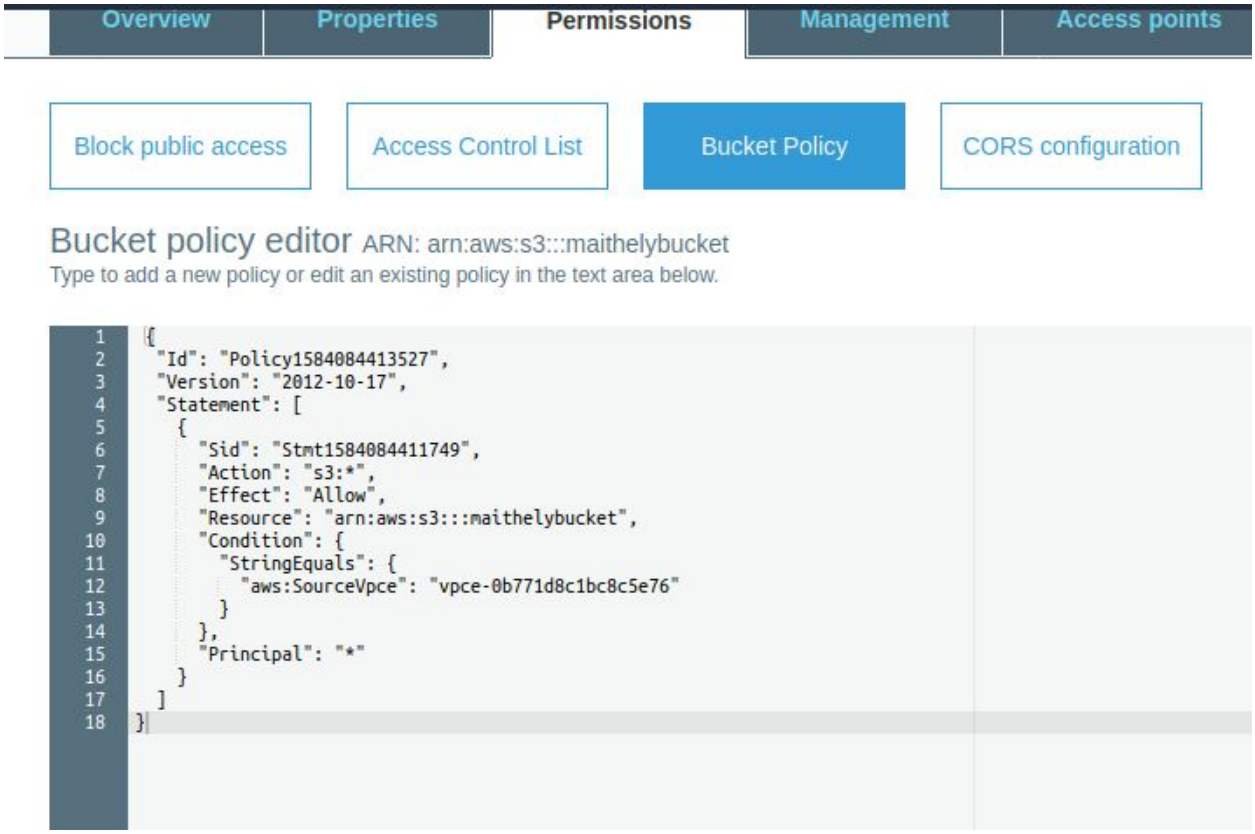
Add Condition

Add Statement No Action selected. You must select at least one Action

Copy the policy




Paste to bucket policy




10. CORS, Enable CORS for 2 specific website.

Amazon S3 > maithelybucket

# maithelybucket

Overview | Properties | **Permissions** | Management | Access points

Block public access | Access Control List | Bucket Policy | CORS configuration

## CORS configuration editor ARN: arn:aws:s3:::maithelybucket

Add a new cors configuration or edit an existing one in the text area below.

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>https://www.facebook.com/*</AllowedOrigin>
    <AllowedOrigin>https://s3.console.aws.amazon.com/*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>PUT</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <AllowedHeader>Authorization</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```

Amazon S3 > maithelybucket

# maithelybucket

**Overview** | Properties | Permissions | Management | Access points

Type a prefix and press Enter to search. Press ESC to clear.

Upload | Create folder | Download | Actions | Versions Hide Show

| Name | Last modified | Size |
|---|---|---|
| Untitled document.pdf | Mar 13, 2020 4:58:58 PM GMT+0530 | 9.7 KB |
| test.txt | Mar 13, 2020 1:24:37 PM GMT+0530 | 34.0 B |

We are able to view this document because we have added “amazon console link”