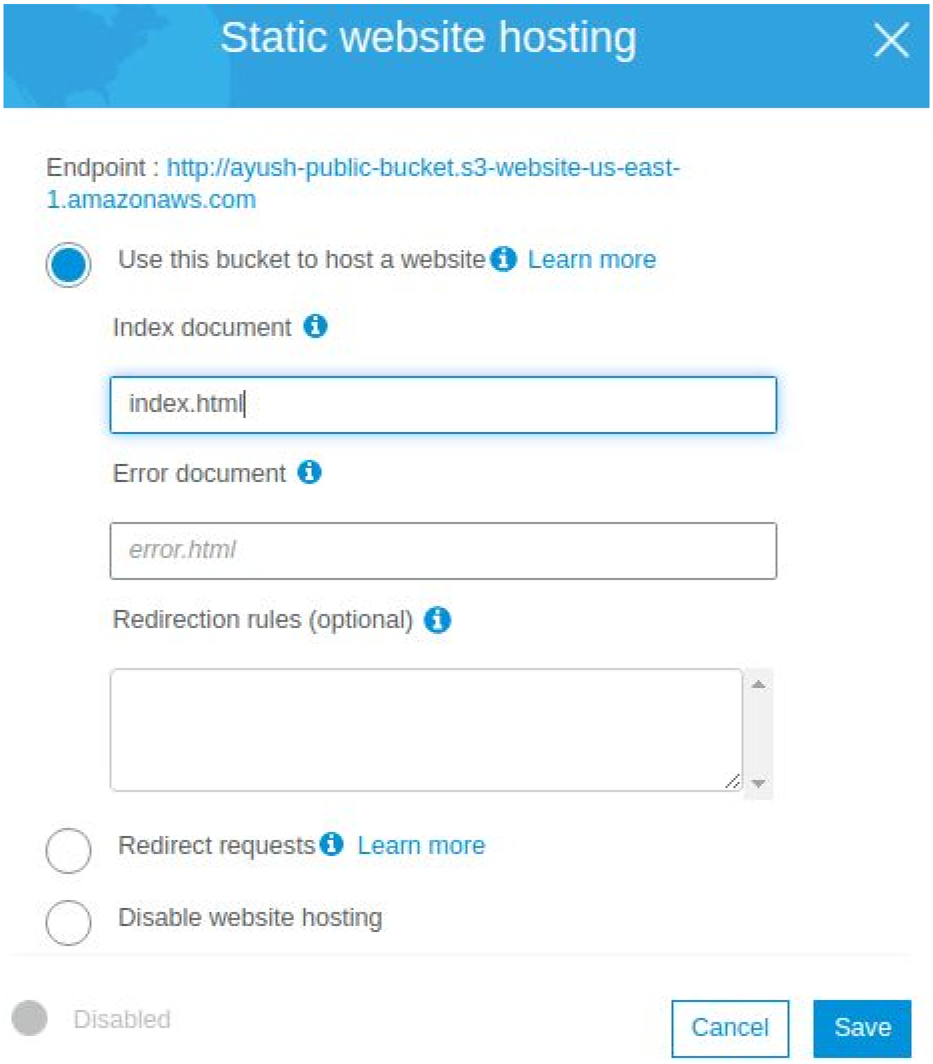
ANS:h1. Static website hosting using s3(what is index and error page).



Static website hosting

Endpoint : http://ayush-public-bucket.s3-website-us-east-1.amazonaws.com

● Use this bucket to host a website

Index document: index.html

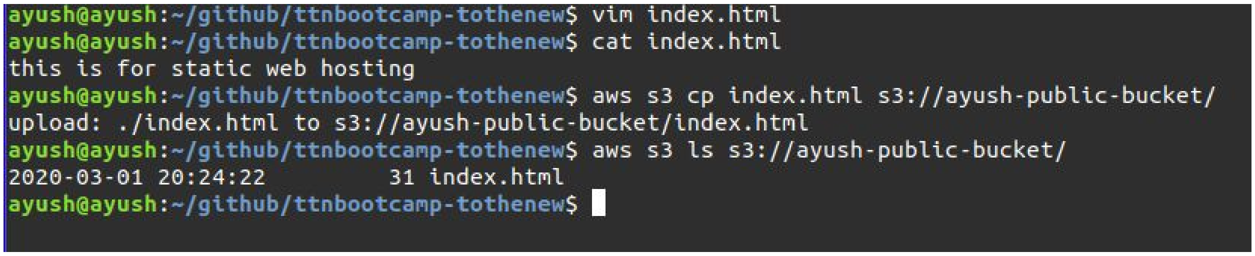
Error document: error.html

Redirection rules (optional)

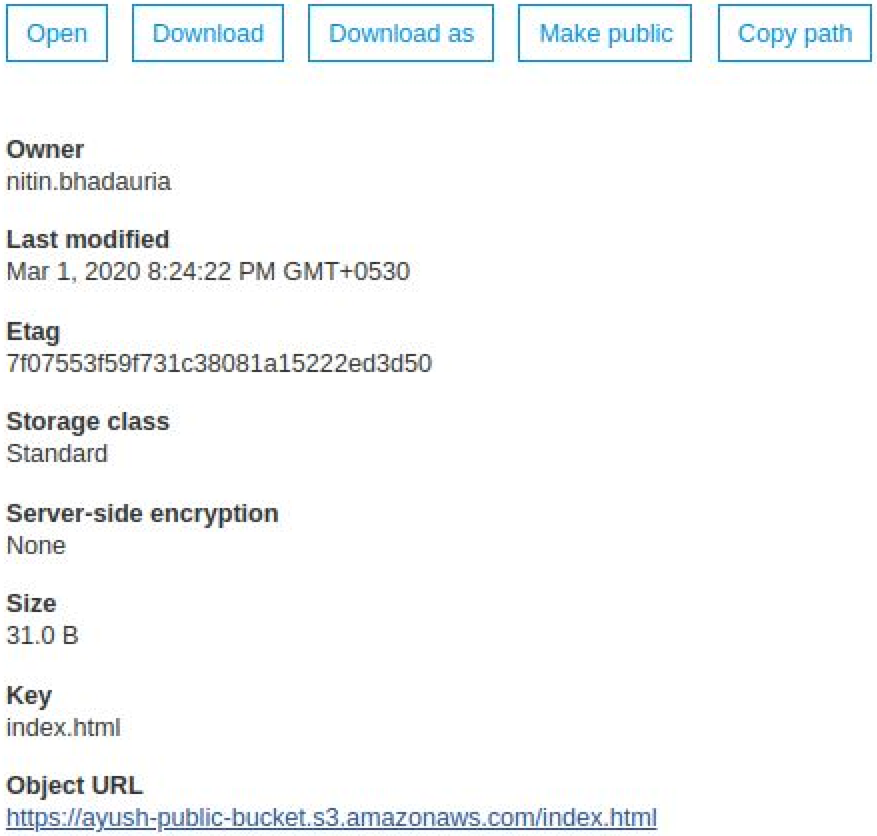
○ Redirect requests

○ Disable website hosting

h



```
ayush@ayush:~/github/ttnbootcamp-tothenew$ vim index.html
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat index.html
this is for static web hosting
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws s3 cp index.html s3://ayush-public-bucket/
upload: ./index.html to s3://ayush-public-bucket/index.html
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws s3 ls s3://ayush-public-bucket/
2020-03-01 20:24:22         31 index.html
ayush@ayush:~/github/ttnbootcamp-tothenew$ 
```



Owner
nitin.bhadauria

Last modified
Mar 1, 2020 8:24:22 PM GMT+0530

Etag
7f07553f59f731c38081a15222ed3d50

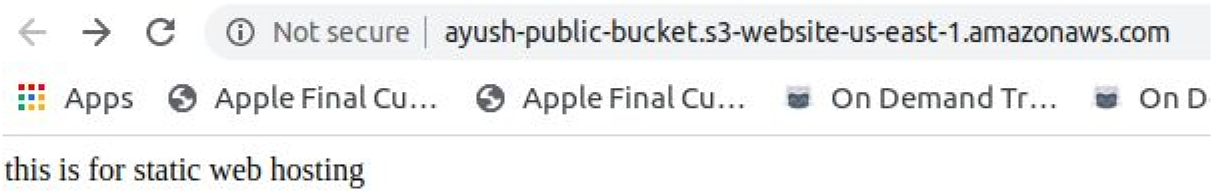
Storage class
Standard

Server-side encryption
None

Size
31.0 B

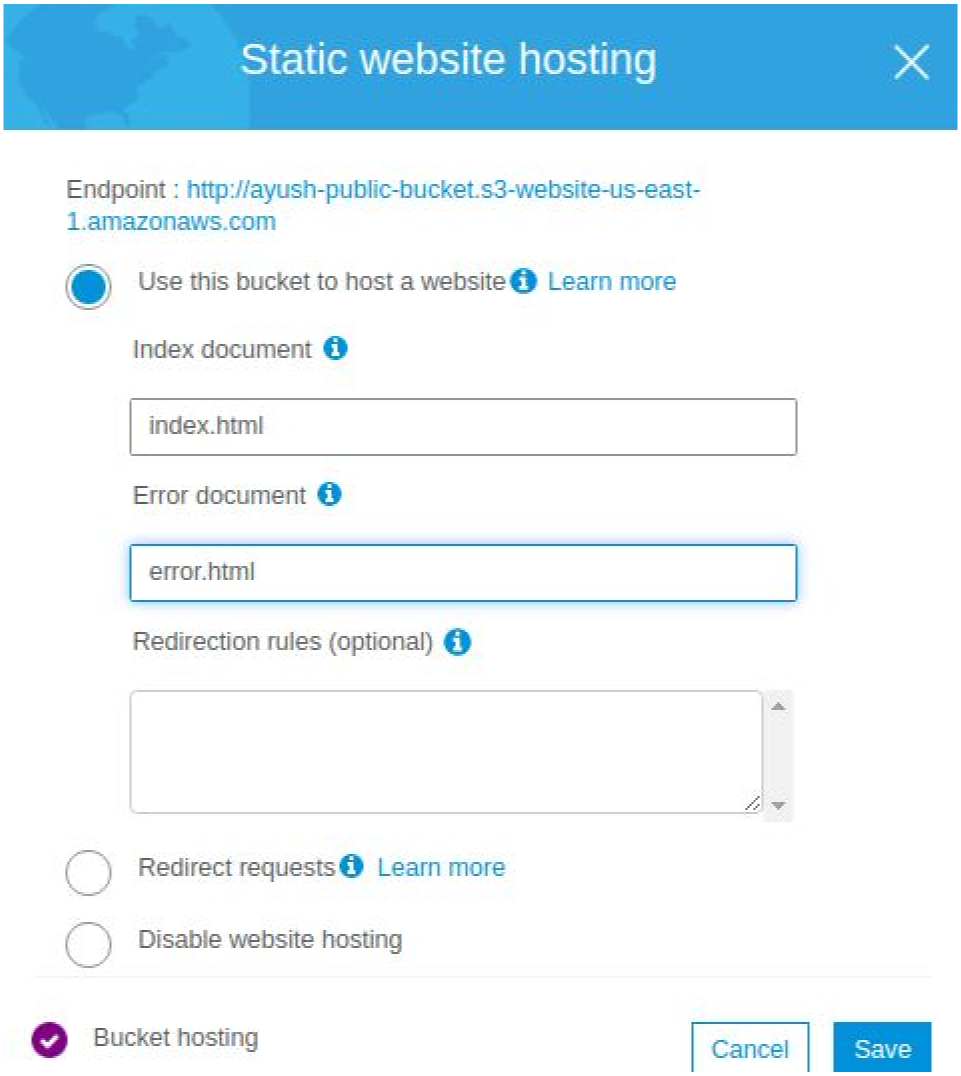
Key
index.html

Object URL
https://ayush-public-bucket.s3.amazonaws.com/index.html



ayush-public-bucket.s3-website-us-east-1.amazonaws.com

this is for static web hosting



Static website hosting

Endpoint : http://ayush-public-bucket.s3-website-us-east-1.amazonaws.com

● Use this bucket to host a website Learn more

Index document: index.html

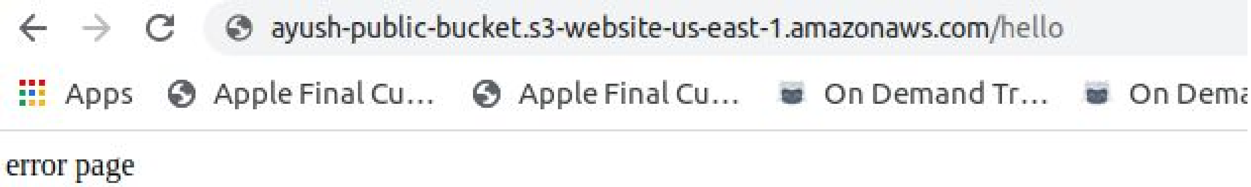
Error document: error.html

Redirection rules (optional):

○ Redirect requests Learn more

○ Disable website hosting

✓ Bucket hosting

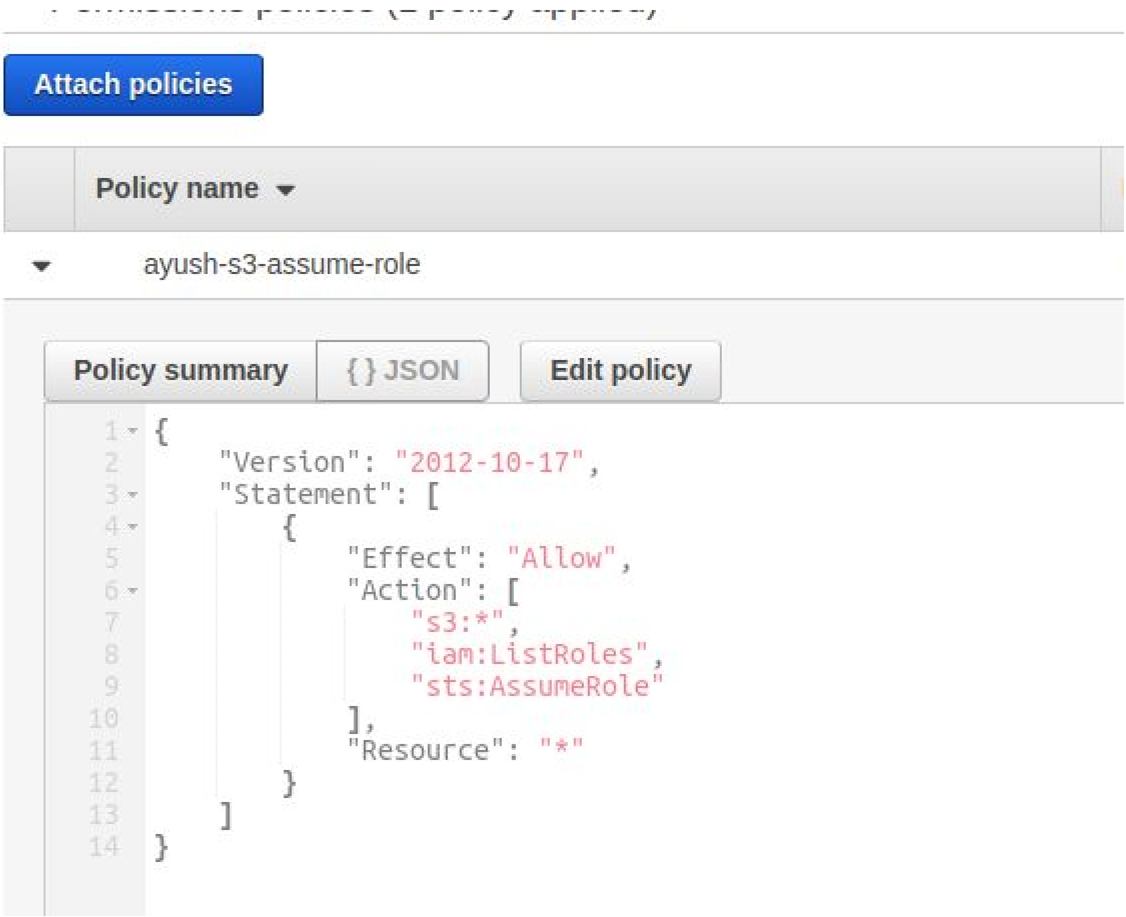


ayush-public-bucket.s3-website-us-east-1.amazonaws.com/hello

error page

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

Create a role that supports sts assume role

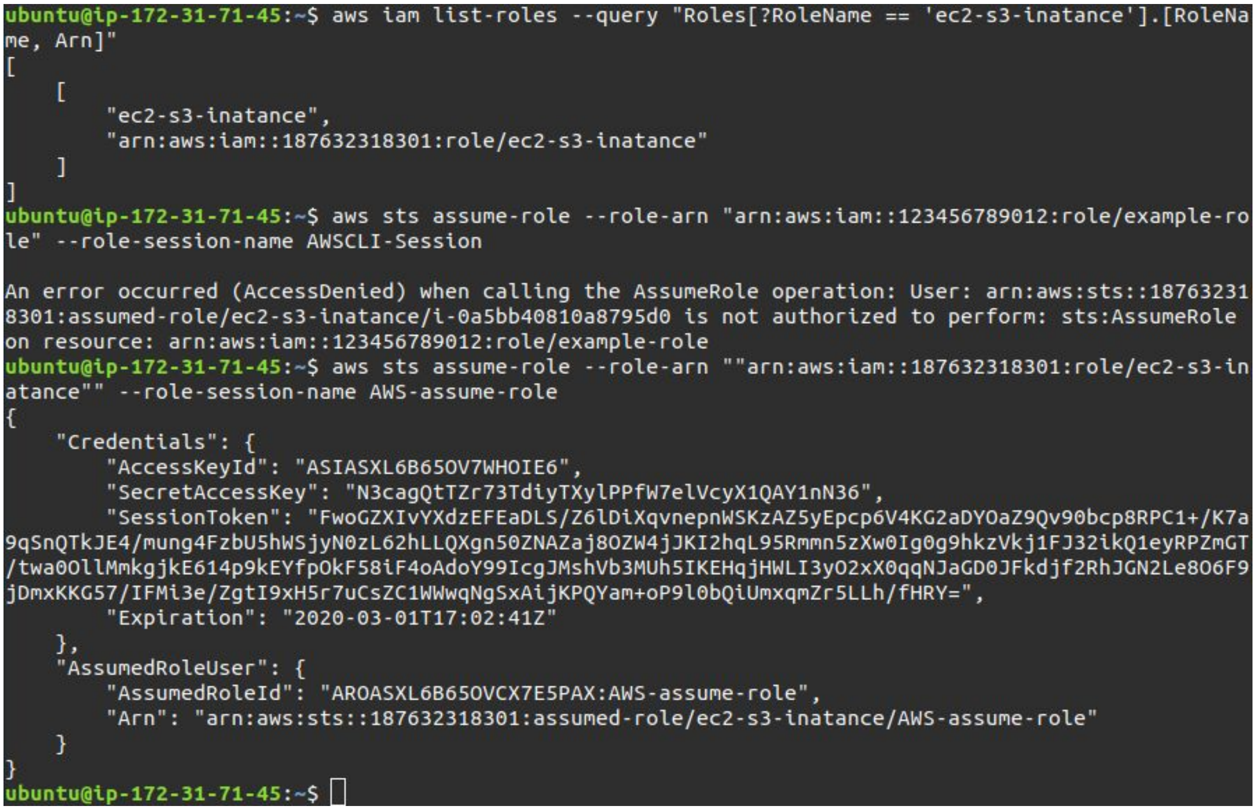


ayush-s3-assume-role

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "iam:ListRoles",
                "sts:AssumeRole"
            ],
            "Resource": "*"
        }
    ]
}
```

Attach this role to first ec2 instance

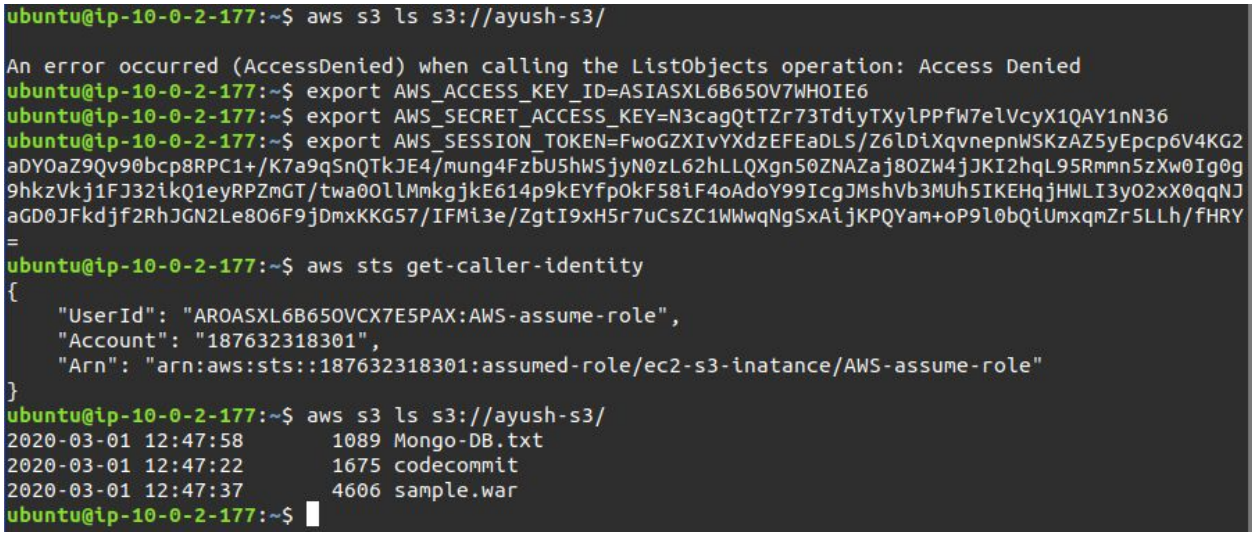
Generate sts credentials for second instance



```
ubuntu@ip-172-31-71-45:~$ aws iam list-roles --query "Roles[?RoleName == 'ec2-s3-inatance'].[RoleNa
me, Arn]"
[
    [
        "ec2-s3-inatance",
        "arn:aws:iam::187632318301:role/ec2-s3-inatance"
    ]
]
ubuntu@ip-172-31-71-45:~$ aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/example-ro
le" --role-session-name AWSCLI-Session

An error occurred (AccessDenied) when calling the AssumeRole operation: User: arn:aws:sts::18763231
8301:assumed-role/ec2-s3-inatance/i-0a5bb40810a8795d0 is not authorized to perform: sts:AssumeRole
on resource: arn:aws:iam::123456789012:role/example-role
ubuntu@ip-172-31-71-45:~$ aws sts assume-role --role-arn ""arn:aws:iam::187632318301:role/ec2-s3-in
atance"" --role-session-name AWS-assume-role
{
    "Credentials": {
        "AccessKeyId": "ASIASXL6B65OV7WHOIE6",
        "SecretAccessKey": "N3cagQtTZr73TdiyTXylPPfW7elVcyX1QAY1nN36",
        "SessionToken": "FwoGZXIvYXdzEFEaDLS/Z6lDiXqvnepnWSKzAZ5yEpcp6V4KG2aDYOaZ9Qv90bcp8RPC1+/K7a
9qSnQTkJE4/mung4FzbU5hWSjyN0zL62hLLQXgn50ZNAZaj8OZW4jJKI2hqL95Rmmn5zXw0Ig0g9hkzVkj1FJ32ikQ1eyRPZmGT
/twa0OllMmkgjkE614p9kEYfpOkF58iF4oAdoY99IcgJMshVb3MUh5IKEHqjHWLI3yO2xX0qqNJaGD0JFkdjf2RhJGN2Le8O6F9
jDmxKKG57/IFMi3e/ZgtI9xH5r7uCsZC1WWwqNgSxAijKPQYam+oP9l0bQiUmxqmZr5LLh/fHRY=",
        "Expiration": "2020-03-01T17:02:41Z"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROASXL6B65OVCX7E5PAX:AWS-assume-role",
        "Arn": "arn:aws:sts::187632318301:assumed-role/ec2-s3-inatance/AWS-assume-role"
    }
}
ubuntu@ip-172-31-71-45:~$ 
```

Add those credentials to this instance and get access to s3

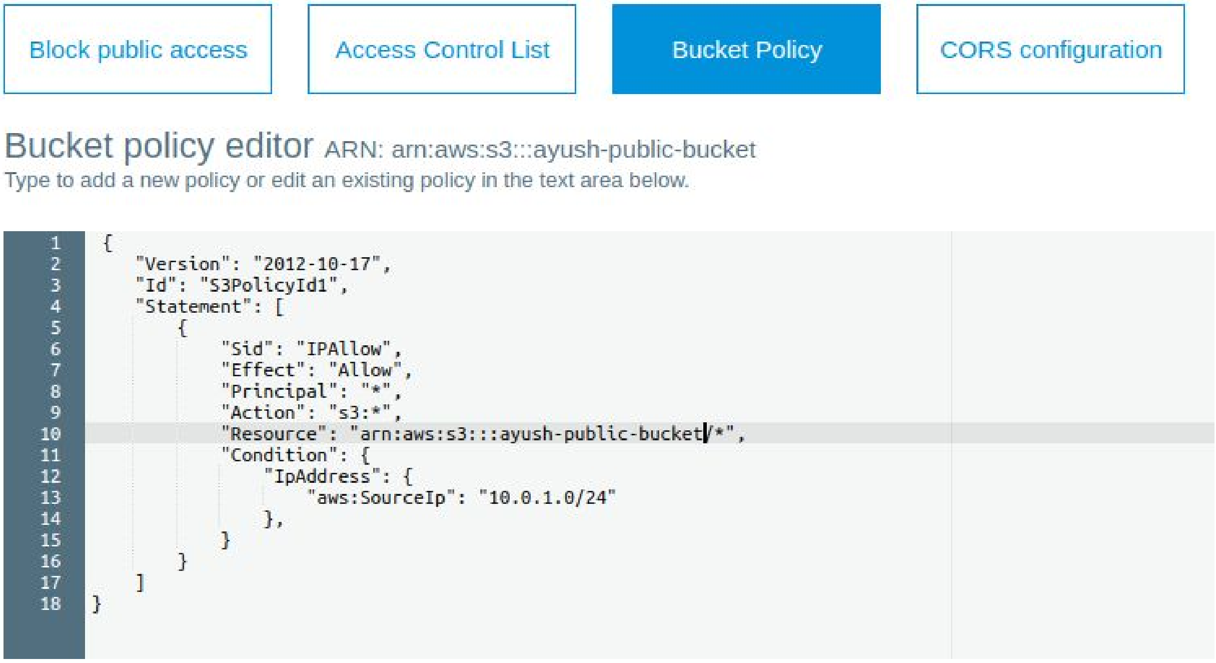


```
ubuntu@ip-10-0-2-177:~$ aws s3 ls s3://ayush-s3/

An error occurred (AccessDenied) when calling the ListObjects operation: Access Denied
ubuntu@ip-10-0-2-177:~$ export AWS_ACCESS_KEY_ID=ASIASXL6B65OV7WHOIE6
ubuntu@ip-10-0-2-177:~$ export AWS_SECRET_ACCESS_KEY=N3cagQtTZr73TdiyTXylPPfW7elVcyX1QAY1nN36
ubuntu@ip-10-0-2-177:~$ export AWS_SESSION_TOKEN=FwoGZXIvYXdzEFEaDLS/Z6lDiXqvnepnWSKzAZ5yEpcp6V4KG2
aDYOaZ9Qv90bcp8RPC1+/K7a9qSnQTkJE4/mung4FzbU5hWSjyN0zL62hLLQXgn50ZNAZaj8OZW4jJKI2hqL95Rmmn5zXw0Ig0g
9hkzVkj1FJ32ikQ1eyRPZmGT/twa0OllMmkgjkE614p9kEYfpOkF58iF4oAdoY99IcgJMshVb3MUh5IKEHqjHWLI3yO2xX0qqNJ
aGD0JFkdjf2RhJGN2Le8O6F9jDmxKKG57/IFMi3e/ZgtI9xH5r7uCsZC1WWwqNgSxAijKPQYam+oP9l0bQiUmxqmZr5LLh/fHRY
=
ubuntu@ip-10-0-2-177:~$ aws sts get-caller-identity
{
    "UserId": "AROASXL6B65OVCX7E5PAX:AWS-assume-role",
    "Account": "187632318301",
    "Arn": "arn:aws:sts::187632318301:assumed-role/ec2-s3-inatance/AWS-assume-role"
}
ubuntu@ip-10-0-2-177:~$ aws s3 ls s3://ayush-s3/
2020-03-01 12:47:58       1089 Mongo-DB.txt
2020-03-01 12:47:22       1675 codecommit
2020-03-01 12:47:37       4606 sample.war
ubuntu@ip-10-0-2-177:~$ 
```

3. Block s3 access on the basis of

i. IP

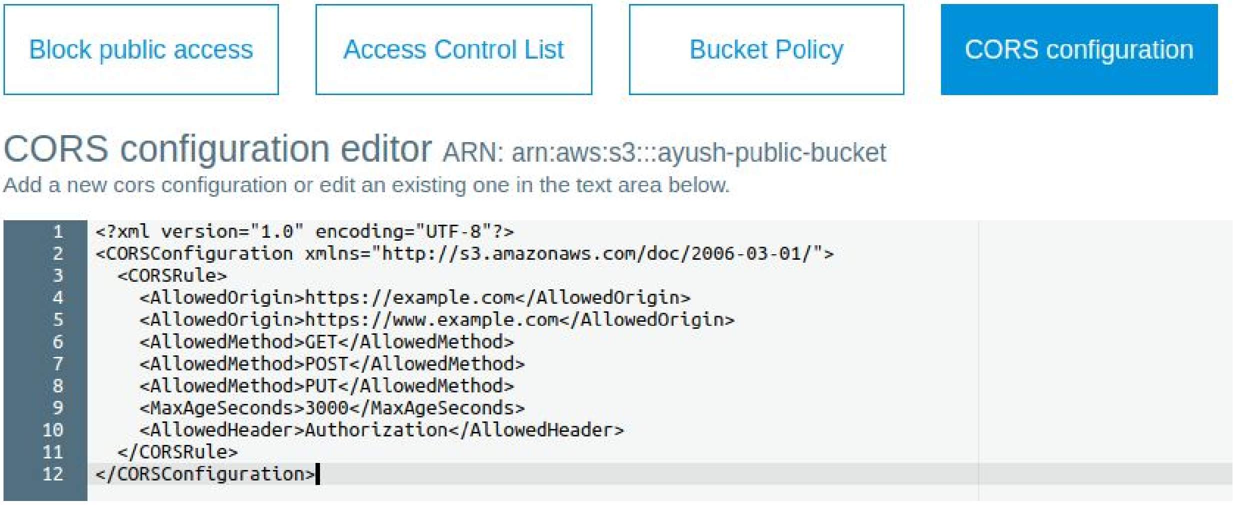


Bucket policy editor ARN: arn:aws:s3:::ayush-public-bucket

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": "Allow",
           "Principal": "*",
           "Action": "s3:*",
           "Resource": "arn:aws:s3:::ayush-public-bucket/*",
           "Condition": {
               "IpAddress": {
                   "aws:SourceIp": "10.0.1.0/24"
               },
           }
       }
   ]
}
```

ii. Domain



CORS configuration editor ARN: arn:aws:s3:::ayush-public-bucket

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

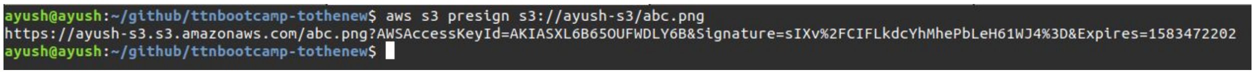
```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <CORSRule>
    <AllowedOrigin>https://example.com</AllowedOrigin>
    <AllowedOrigin>https://www.example.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>POST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

iii. Pre-signed URL(Time based)

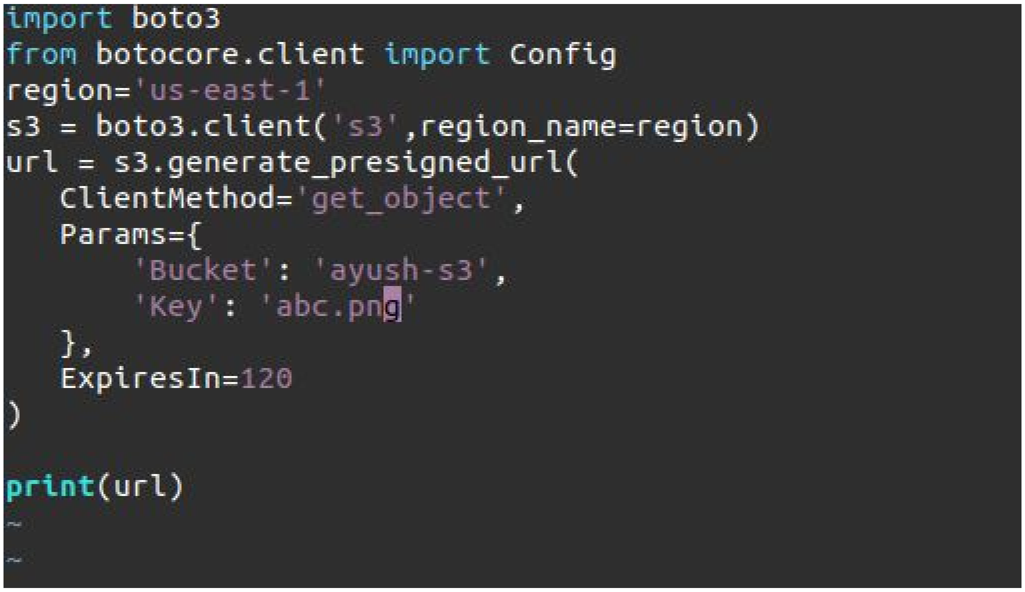
Query string authentication and URL-based access are hidden gems in Amazon S3. These methods allow you to grant permissions based on a specific URL. There are two common patterns for using this type of authentication

- Allowing someone or something to upload a key to your bucket
- Providing temporary access to a specific key

This is a great method of securing providing one-time access to your Amazon S3 buckets.

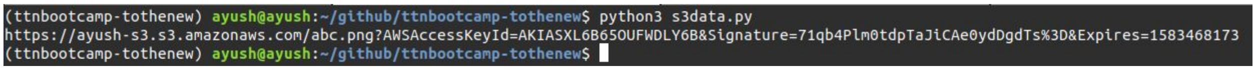


```
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws s3 presign s3://ayush-s3/abc.png
https://ayush-s3.s3.amazonaws.com/abc.png?AWSAccessKeyId=AKIASXL6B65OUFWDLY6B&Signature=sIXv%2FCIFLkdcYhMhePbLeH61WJ4%3D&Expires=1583472202
ayush@ayush:~/github/ttnbootcamp-tothenew$ 
```

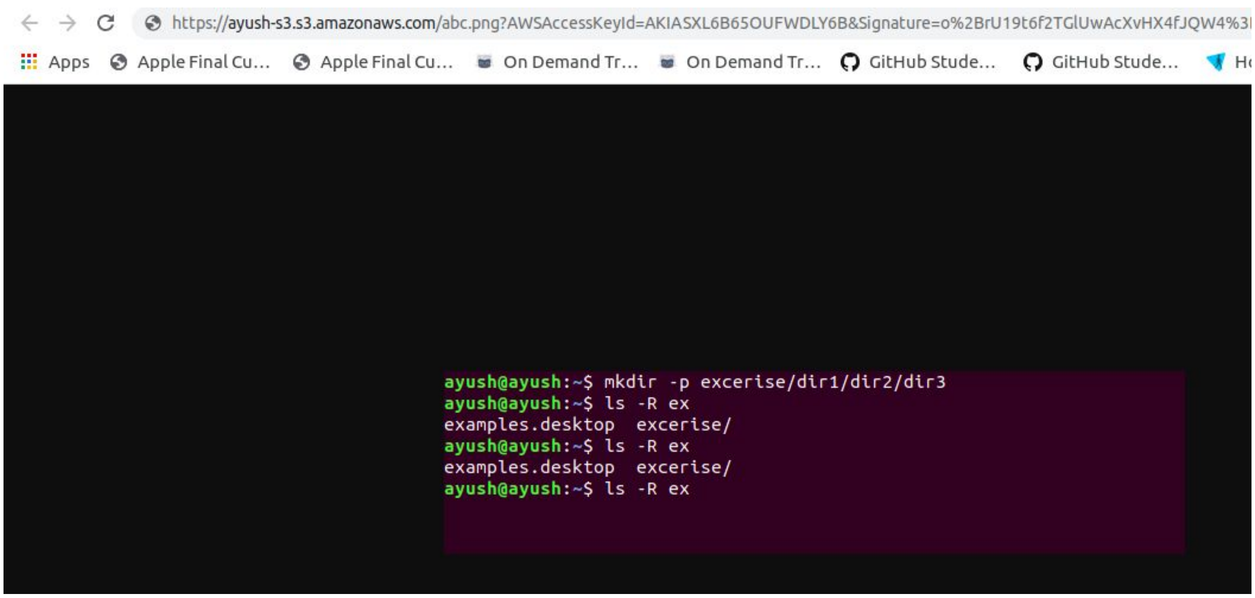


```
import boto3
from botocore.client import Config
region='us-east-1'
s3 = boto3.client('s3',region_name=region)
url = s3.generate_presigned_url(
   ClientMethod='get_object',
   Params={
       'Bucket': 'ayush-s3',
       'Key': 'abc.png'
   },
   ExpiresIn=120
)

print(url)
~
~
```



```
(ttnbootcamp-tothenew) ayush@ayush:~/github/ttnbootcamp-tothenew$ python3 s3data.py
https://ayush-s3.s3.amazonaws.com/abc.png?AWSAccessKeyId=AKIASXL6B65OUFWDLY6B&Signature=71qb4Plm0tdpTaJiCAe0ydDgdTs%3D&Expires=1583468173
(ttnbootcamp-tothenew) ayush@ayush:~/github/ttnbootcamp-tothenew$ 
```

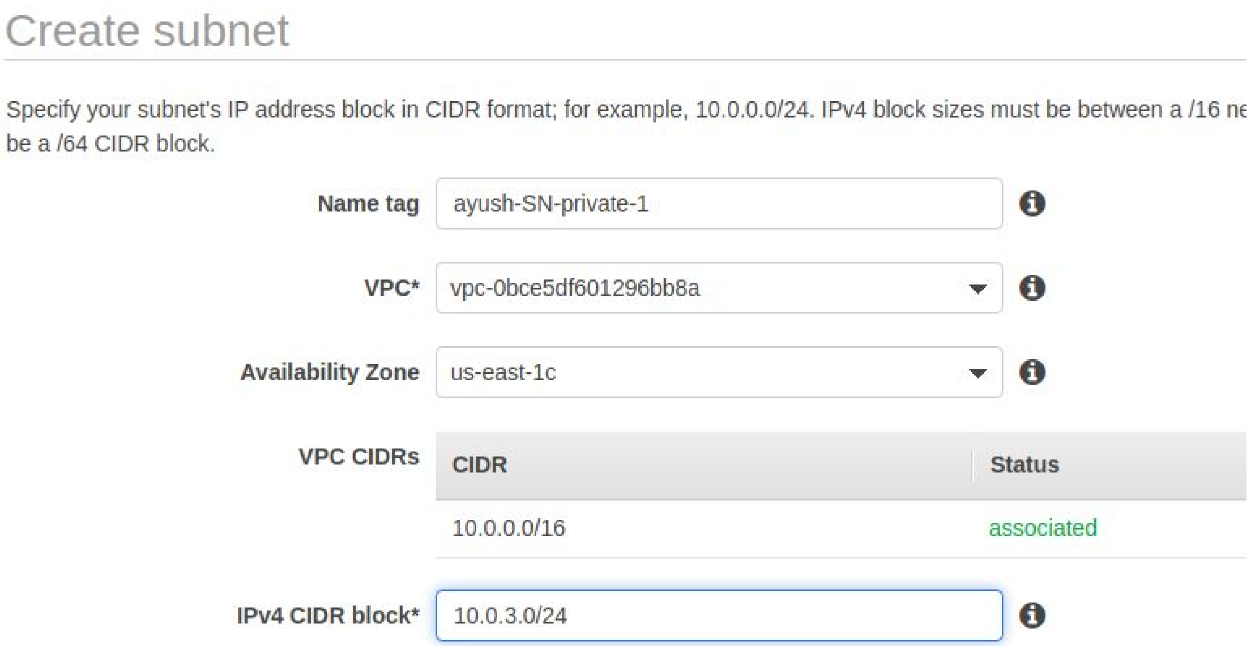


https://ayush-s3.s3.amazonaws.com/abc.png?AWSAccessKeyId=AKIASXL6B65OUFWDLY6B&Signature=o%2BrU19t6f2TGlUwAcXvHX4fJQW4%3

```
ayush@ayush:~$ mkdir -p excerise/dir1/dir2/dir3
ayush@ayush:~$ ls -R ex
examples.desktop  excerise/
ayush@ayush:~$ ls -R ex
examples.desktop  excerise/
ayush@ayush:~$ ls -R ex
```

4. Create RDS subnet and launch RDS instance.

Add subnet through subnet group



Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 ne be a /64 CIDR block.

Name tag: ayush-SN-private-1

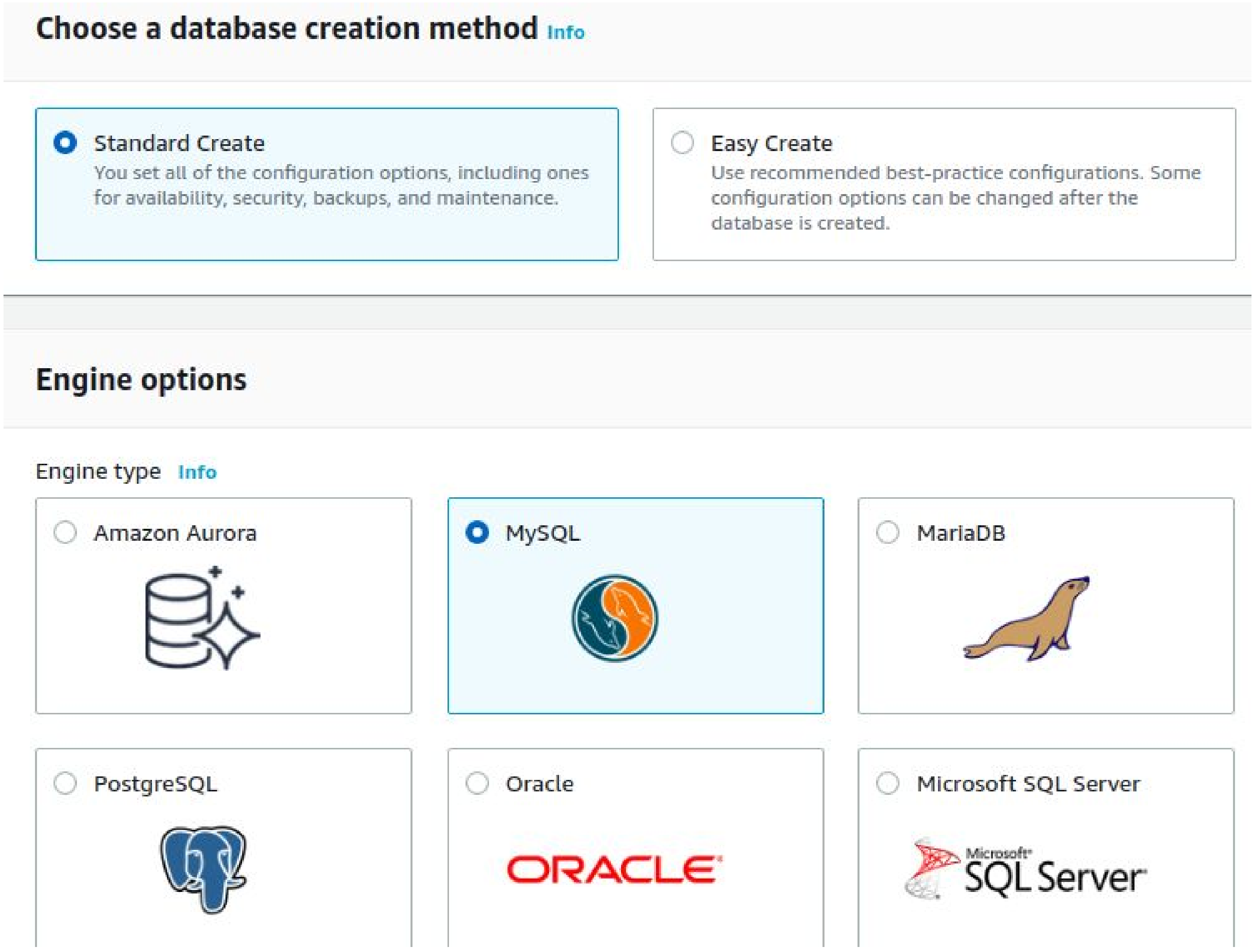
VPC*: vpc-0bce5df601296bb8a

Availability Zone: us-east-1c

VPC CIDRs

CIDR	Status
10.0.0.0/16	associated

IPv4 CIDR block*: 10.0.3.0/24



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

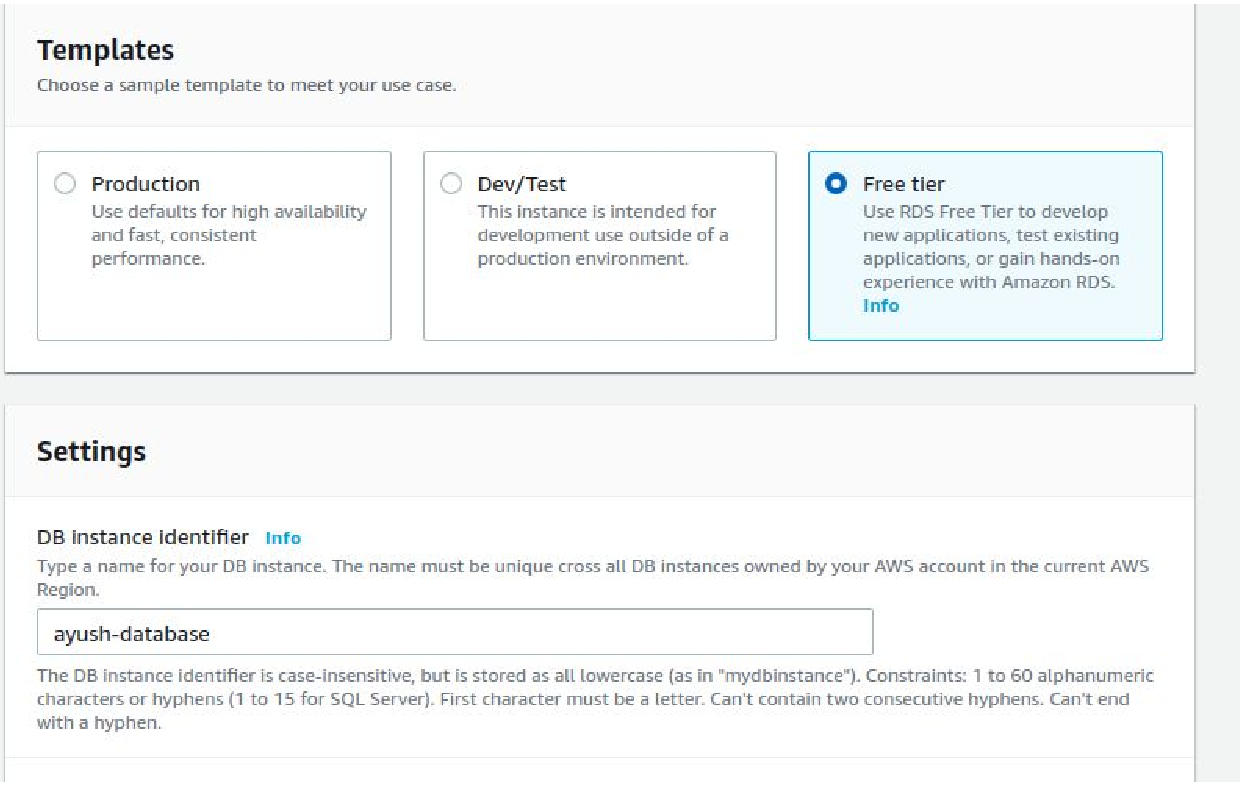
● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server



Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

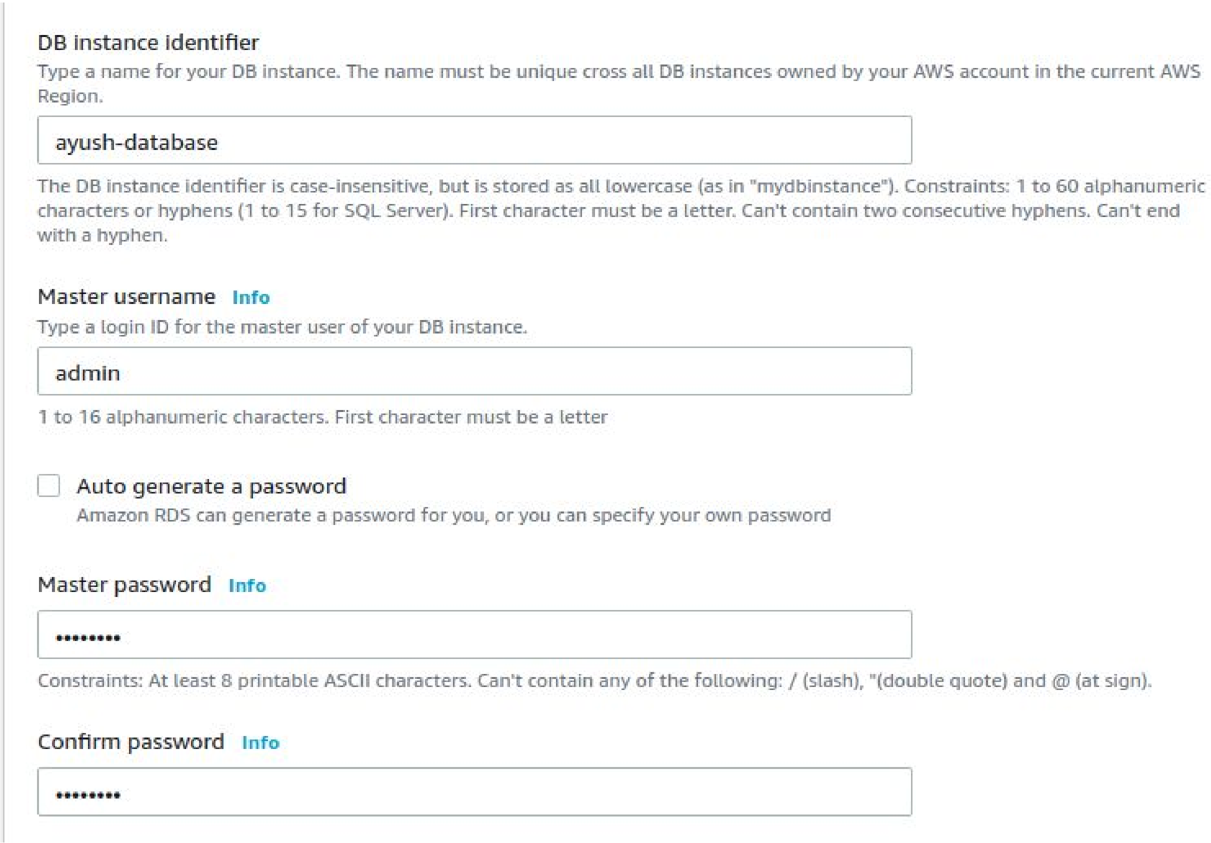
Settings

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.

ayush-database

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.



DB instance identifier

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.

ayush-database

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.

Master username

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

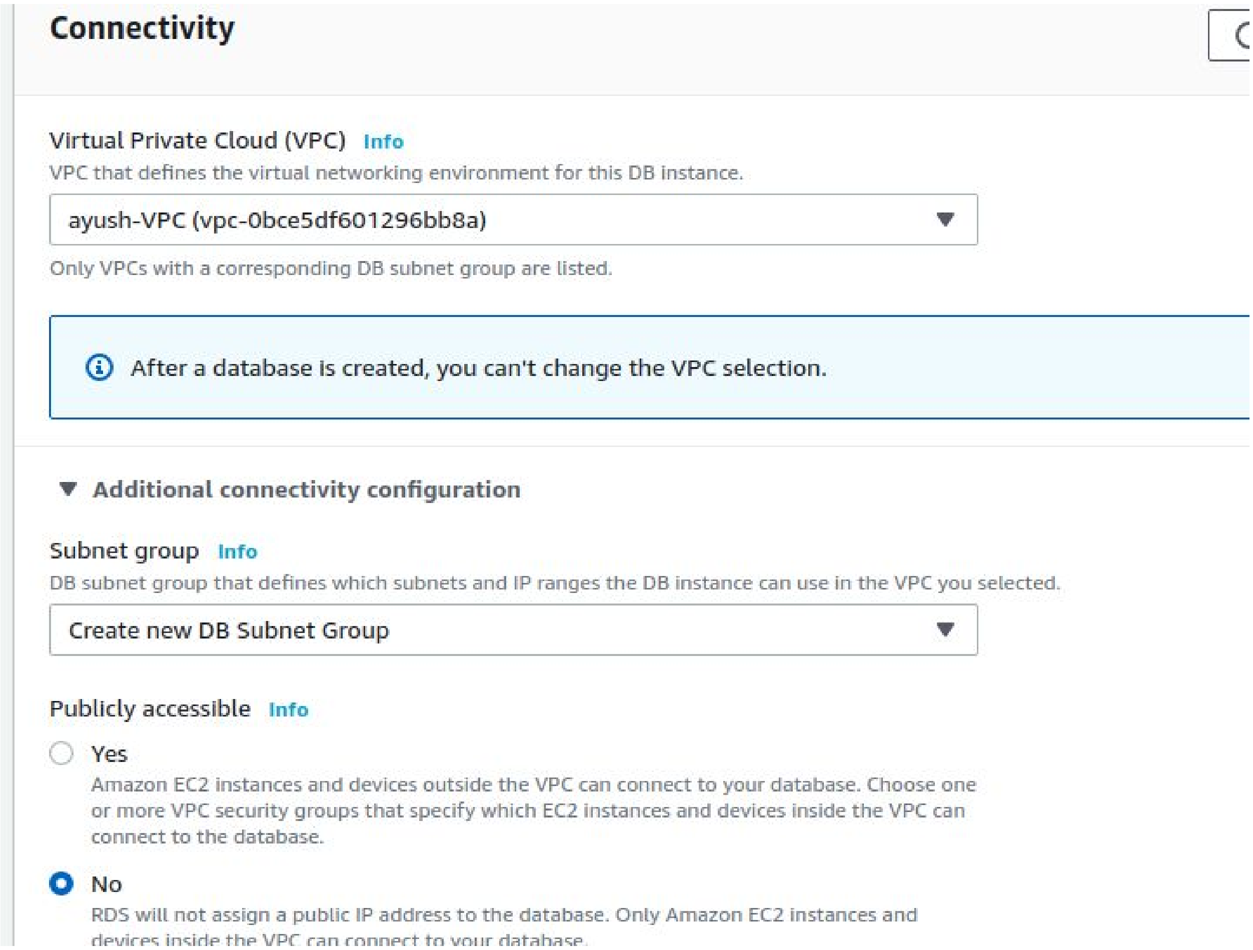
Master password

•••••••••

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

Confirm password

•••••••••



Connectivity

Virtual Private Cloud (VPC) Info

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

ayush-VPC (vpc-0bce5df601296bb8a) ▼

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.

Create new DB Subnet Group ▼

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.

5. what is parameter group and option group,ACL, Bucket policy, IAM Policy?

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.

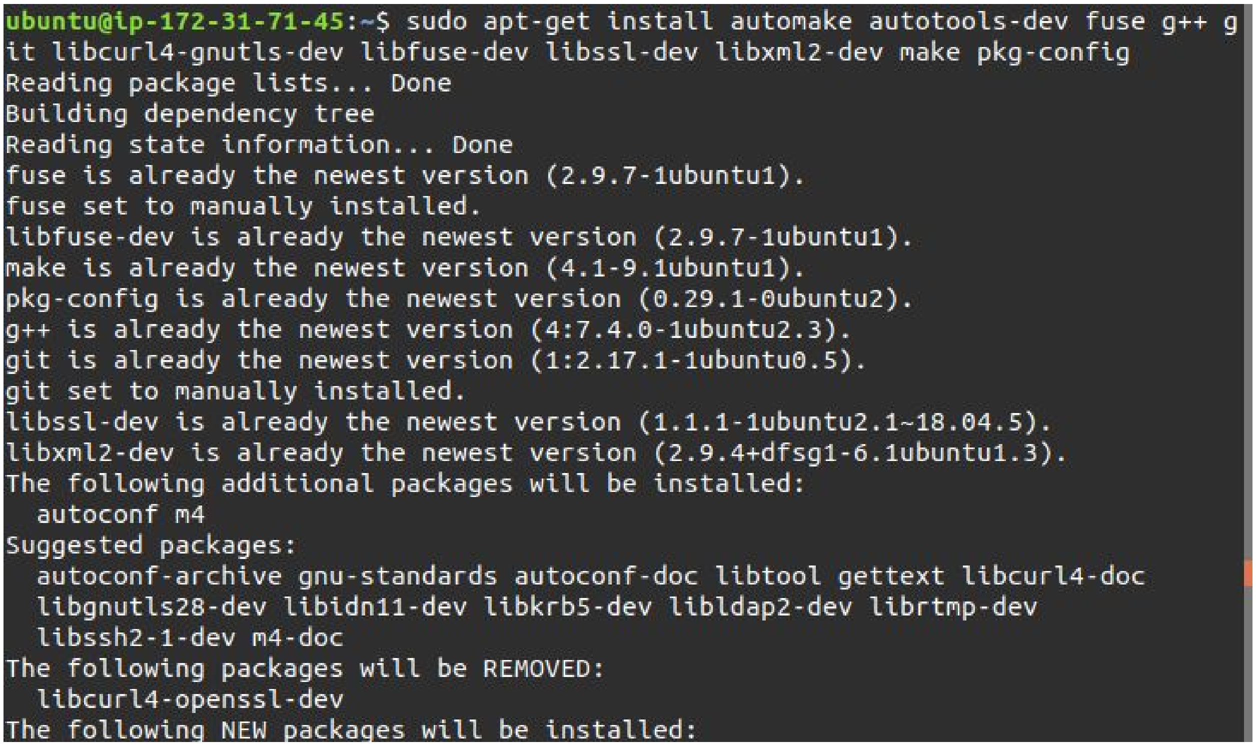
Option Group: Some DB engines offer additional features that make it easier to manage data and databases, and to provide additional security for your database. Amazon RDS uses option groups to enable and configure these features. An *option group* can specify features, called options, that are available 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.

ACL: 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.

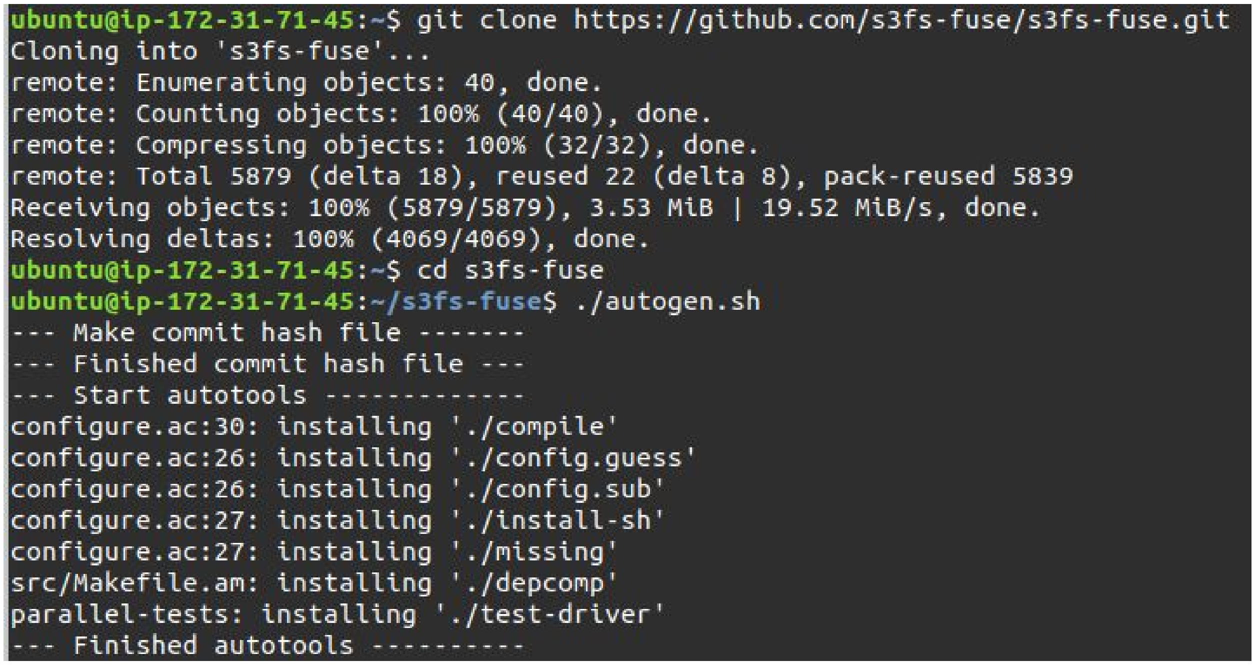
Bucket policy: A bucket policy is a resource-based AWS Identity and Access Management (IAM) policy. You add a bucket policy to a bucket to grant other AWS accounts or IAM users access permissions for the bucket and the objects in it. Object permissions apply only to the objects that the bucket owner creates.

IAM Policy: A policy is an entity that, when attached to an identity or resource, defines their permissions. You can use the AWS Management Console, AWS CLI, or AWS API to create customer managed policies in IAM. Customer managed policies are standalone policies that you administer in your own AWS account.

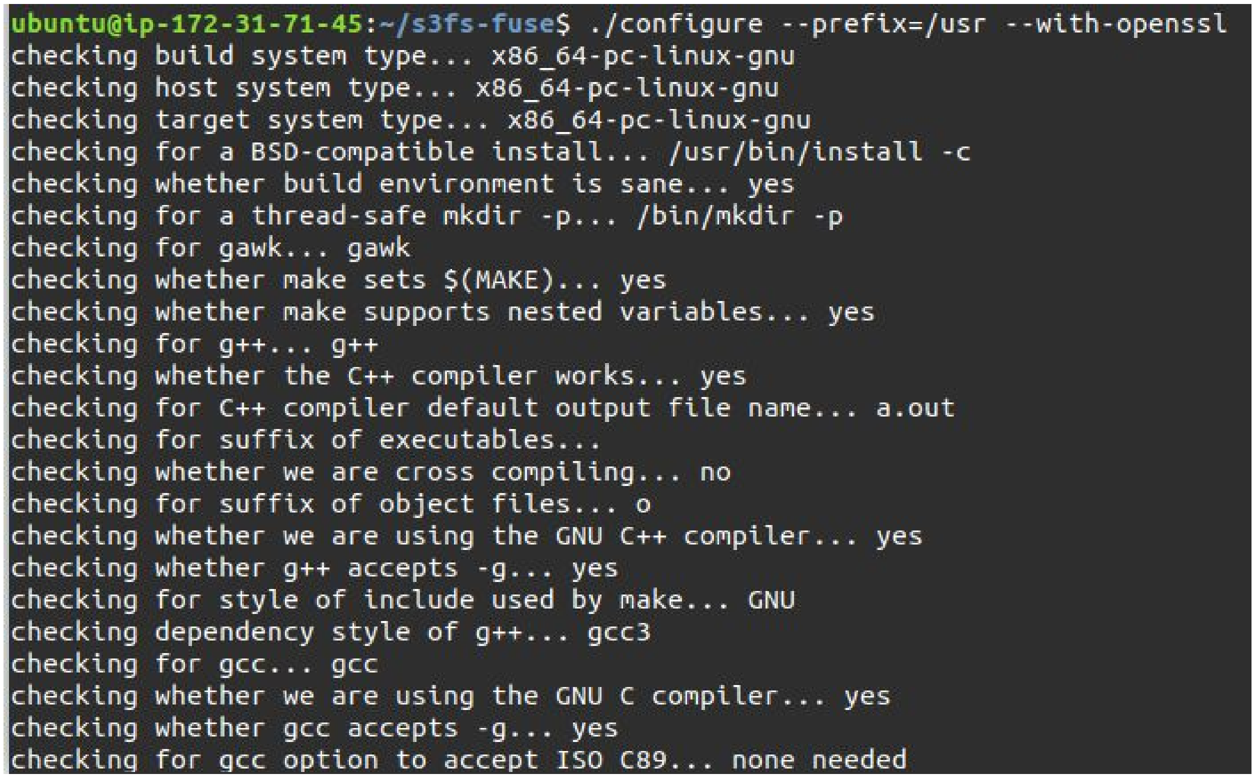
6. Mount S3 to an EC2 instance.



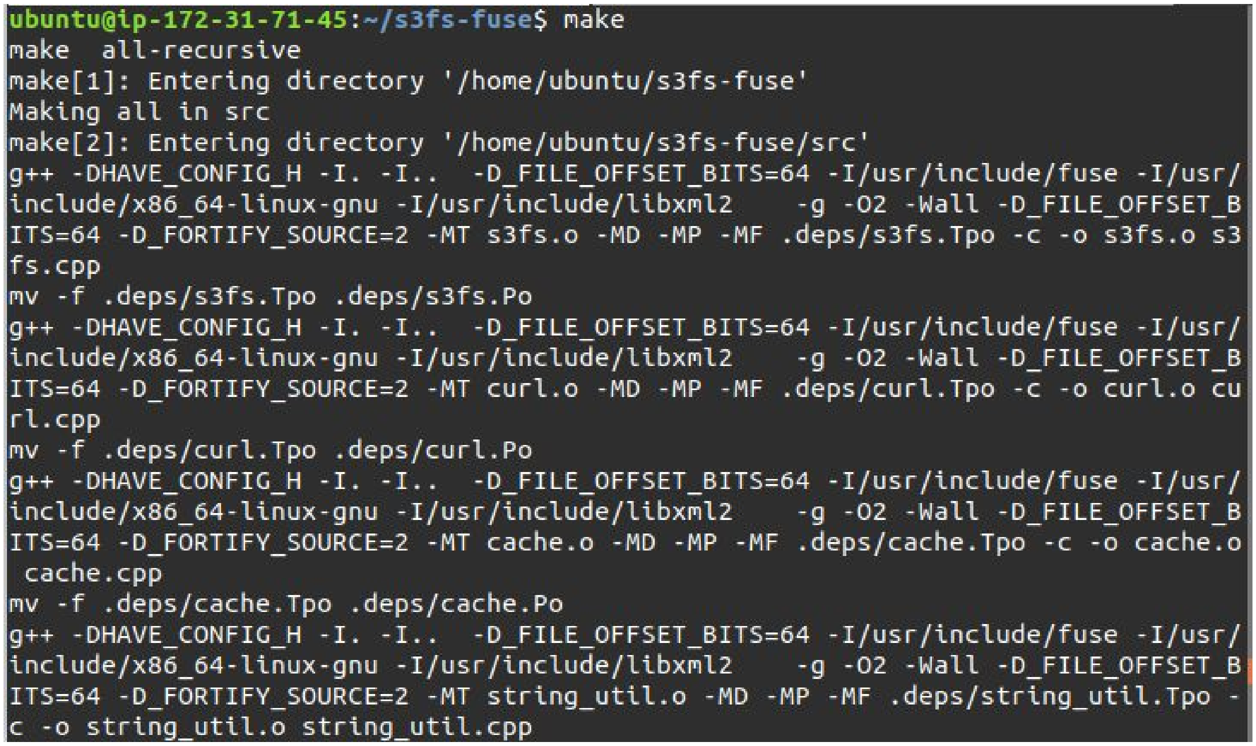
```
ubuntu@ip-172-31-71-45:~$ sudo apt-get install automake autotools-dev fuse g++ g
it libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
fuse is already the newest version (2.9.7-1ubuntu1).
fuse set to manually installed.
libfuse-dev is already the newest version (2.9.7-1ubuntu1).
make is already the newest version (4.1-9.1ubuntu1).
pkg-config is already the newest version (0.29.1-0ubuntu2).
g++ is already the newest version (4:7.4.0-1ubuntu2.3).
git is already the newest version (1:2.17.1-1ubuntu0.5).
git set to manually installed.
libssl-dev is already the newest version (1.1.1-1ubuntu2.1~18.04.5).
libxml2-dev is already the newest version (2.9.4+dfsg1-6.1ubuntu1.3).
The following additional packages will be installed:
  autoconf m4
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc libtool gettext libcurl4-doc
  libgnutls28-dev libidn11-dev libkrb5-dev libldap2-dev librtmp-dev
  libssh2-1-dev m4-doc
The following packages will be REMOVED:
  libcurl4-openssl-dev
The following NEW packages will be installed:
```



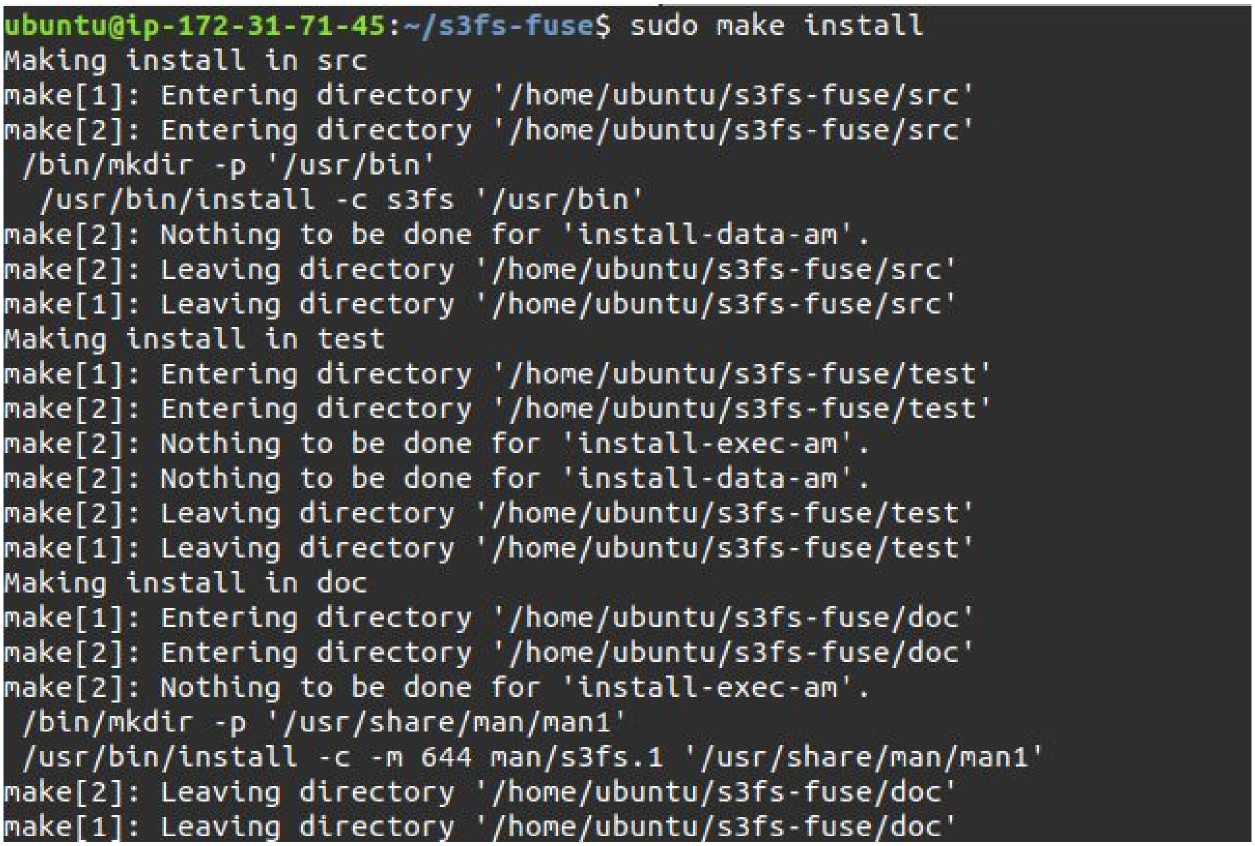
```
ubuntu@ip-172-31-71-45:~$ git clone https://github.com/s3fs-fuse/s3fs-fuse.git
Cloning into 's3fs-fuse'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 5879 (delta 18), reused 22 (delta 8), pack-reused 5839
Receiving objects: 100% (5879/5879), 3.53 MiB | 19.52 MiB/s, done.
Resolving deltas: 100% (4069/4069), done.
ubuntu@ip-172-31-71-45:~$ cd s3fs-fuse
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ ./autogen.sh
--- Make commit hash file -------
--- Finished commit hash file ---
--- Start autotools -------------
configure.ac:30: installing './compile'
configure.ac:26: installing './config.guess'
configure.ac:26: installing './config.sub'
configure.ac:27: installing './install-sh'
configure.ac:27: installing './missing'
src/Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
--- Finished autotools ----------
```



```
ubuntu@ip-172-31-71-45:~/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++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking for style of include used by make... GNU
checking dependency style of g++... gcc3
checking for gcc... gcc
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
```

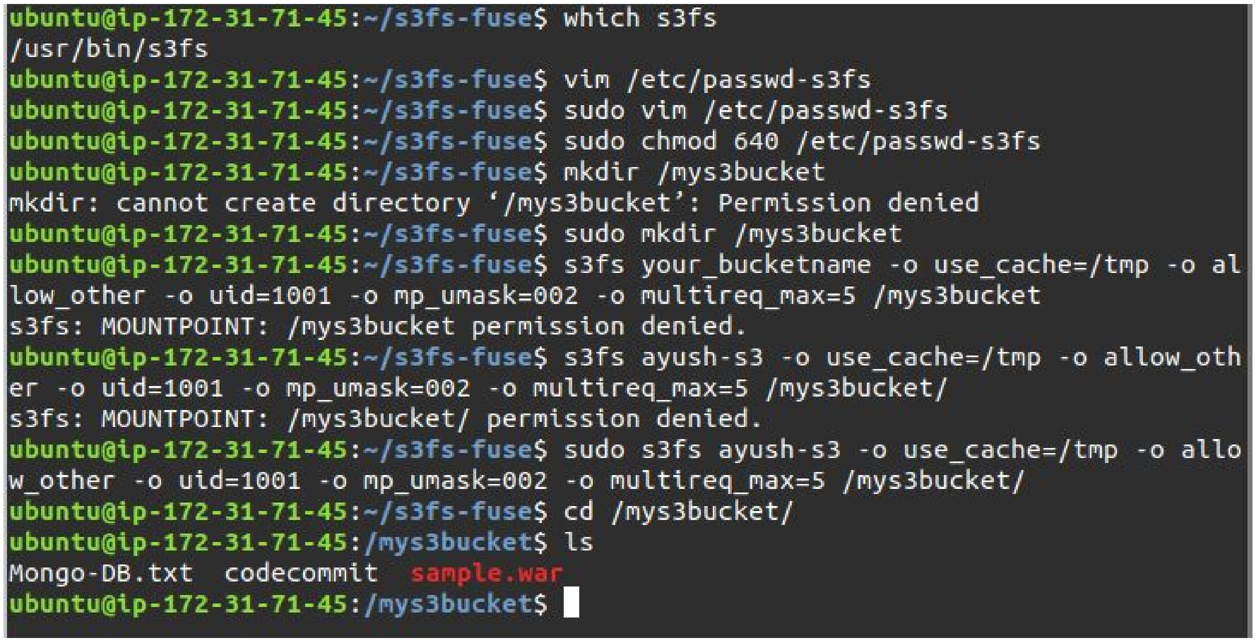


```
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ make
make  all-recursive
make[1]: Entering directory '/home/ubuntu/s3fs-fuse'
Making all in src
make[2]: Entering directory '/home/ubuntu/s3fs-fuse/src'
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/
include/x86_64-linux-gnu -I/usr/include/libxml2    -g -O2 -Wall -D_FILE_OFFSET_B
ITS=64 -D_FORTIFY_SOURCE=2 -MT s3fs.o -MD -MP -MF .deps/s3fs.Tpo -c -o s3fs.o s3
fs.cpp
mv -f .deps/s3fs.Tpo .deps/s3fs.Po
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/
include/x86_64-linux-gnu -I/usr/include/libxml2    -g -O2 -Wall -D_FILE_OFFSET_B
ITS=64 -D_FORTIFY_SOURCE=2 -MT curl.o -MD -MP -MF .deps/curl.Tpo -c -o curl.o cu
rl.cpp
mv -f .deps/curl.Tpo .deps/curl.Po
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/
include/x86_64-linux-gnu -I/usr/include/libxml2    -g -O2 -Wall -D_FILE_OFFSET_B
ITS=64 -D_FORTIFY_SOURCE=2 -MT cache.o -MD -MP -MF .deps/cache.Tpo -c -o cache.o
 cache.cpp
mv -f .deps/cache.Tpo .deps/cache.Po
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/
include/x86_64-linux-gnu -I/usr/include/libxml2    -g -O2 -Wall -D_FILE_OFFSET_B
ITS=64 -D_FORTIFY_SOURCE=2 -MT string_util.o -MD -MP -MF .deps/string_util.Tpo -
c -o string_util.o string_util.cpp
```



```
ubuntu@ip-172-31-71-45:~/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
make[1]: Entering directory '/home/ubuntu/s3fs-fuse/doc'
make[2]: Entering directory '/home/ubuntu/s3fs-fuse/doc'
make[2]: Nothing to be done for 'install-exec-am'.
 /bin/mkdir -p '/usr/share/man/man1'
 /usr/bin/install -c -m 644 man/s3fs.1 '/usr/share/man/man1'
make[2]: Leaving directory '/home/ubuntu/s3fs-fuse/doc'
make[1]: Leaving directory '/home/ubuntu/s3fs-fuse/doc'
```

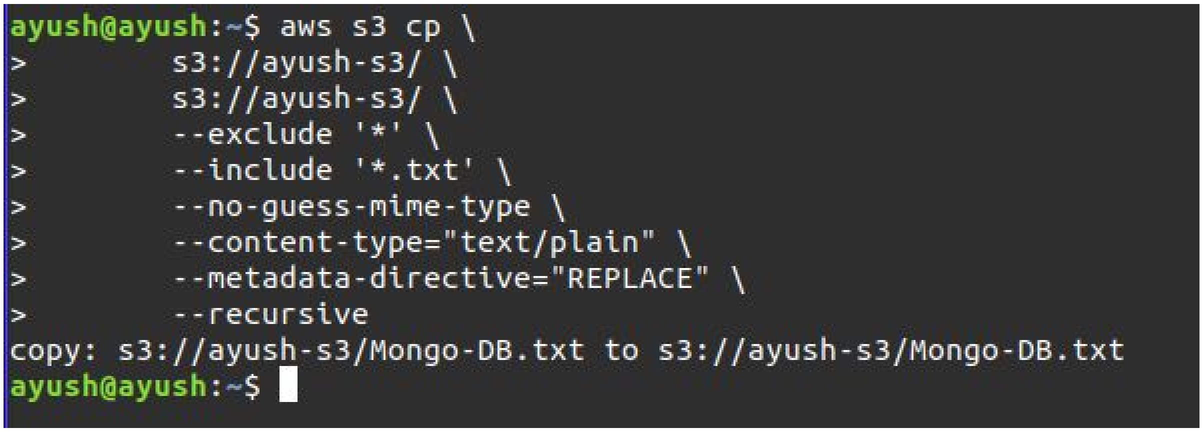
Add account credentials in /etc/passwd-s3fs in Your_access_key_id:Your_secret_key_id format



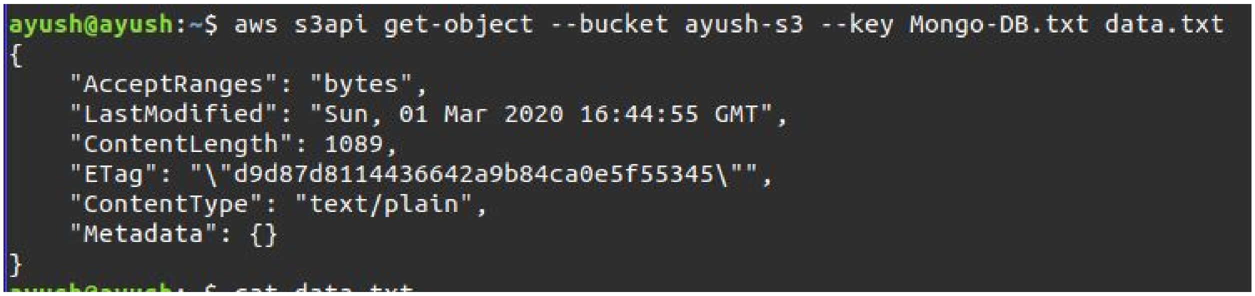
```
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ which s3fs
/usr/bin/s3fs
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ vim /etc/passwd-s3fs
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ sudo vim /etc/passwd-s3fs
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ sudo chmod 640 /etc/passwd-s3fs
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ mkdir /mys3bucket
mkdir: cannot create directory ‘/mys3bucket’: Permission denied
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ sudo mkdir /mys3bucket
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ s3fs your_bucketname -o use_cache=/tmp -o al
low_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket
s3fs: MOUNTPOINT: /mys3bucket permission denied.
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ s3fs ayush-s3 -o use_cache=/tmp -o allow_oth
er -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket/
s3fs: MOUNTPOINT: /mys3bucket/ permission denied.
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ sudo s3fs ayush-s3 -o use_cache=/tmp -o allo
w_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket/
ubuntu@ip-172-31-71-45:~/s3fs-fuse$ cd /mys3bucket/
ubuntu@ip-172-31-71-45:/mys3bucket$ ls
Mongo-DB.txt  codecommit  sample.war
ubuntu@ip-172-31-71-45:/mys3bucket$ 
```

Reference:
https://cloudkul.com/blog/mounting-s3-bucket-linux-ec2-instance/

7. Change content type using s3.

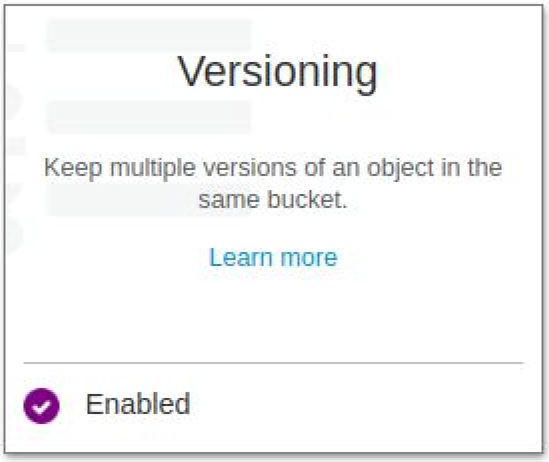


```
ayush@ayush:~$ aws s3 cp \
>         s3://ayush-s3/ \
>         s3://ayush-s3/ \
>         --exclude '*' \
>         --include '*.txt' \
>         --no-guess-mime-type \
>         --content-type="text/plain" \
>         --metadata-directive="REPLACE" \
>         --recursive
copy: s3://ayush-s3/Mongo-DB.txt to s3://ayush-s3/Mongo-DB.txt
ayush@ayush:~$ 
```



```
ayush@ayush:~$ aws s3api get-object --bucket ayush-s3 --key Mongo-DB.txt data.txt
{
    "AcceptRanges": "bytes",
    "LastModified": "Sun, 01 Mar 2020 16:44:55 GMT",
    "ContentLength": 1089,
    "ETag": "\"d9d87d8114436642a9b84ca0e5f55345\"",
    "ContentType": "text/plain",
    "Metadata": {}
}
ayush@ayush:~$ cat data.txt
```

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

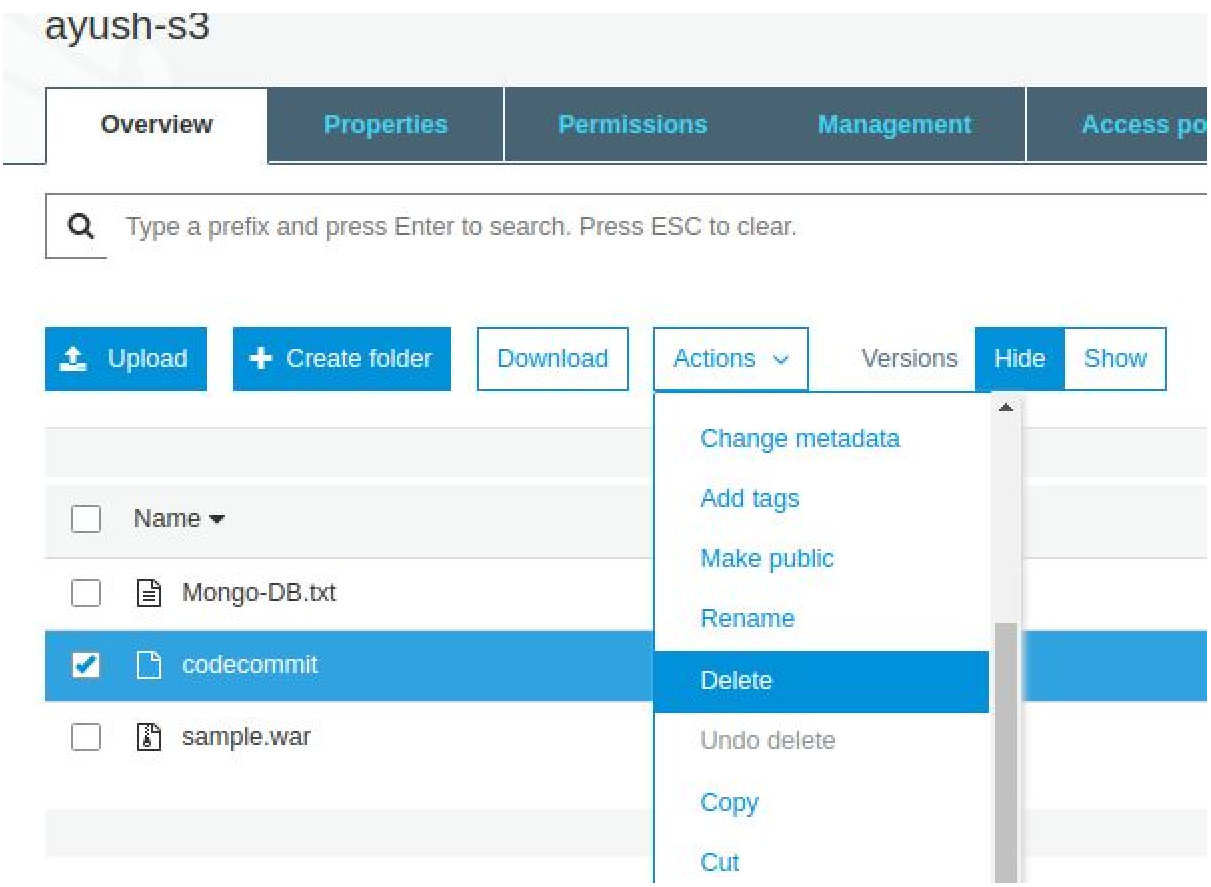


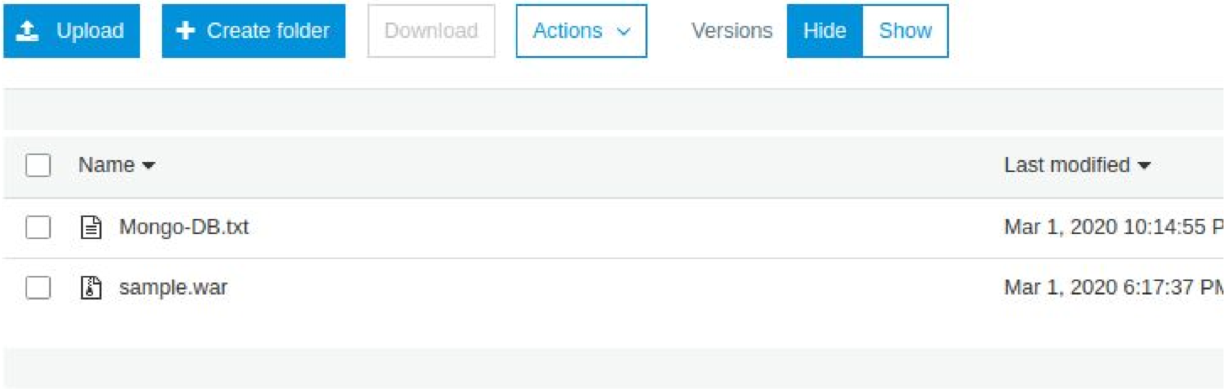
Versioning

Keep multiple versions of an object in the same bucket.

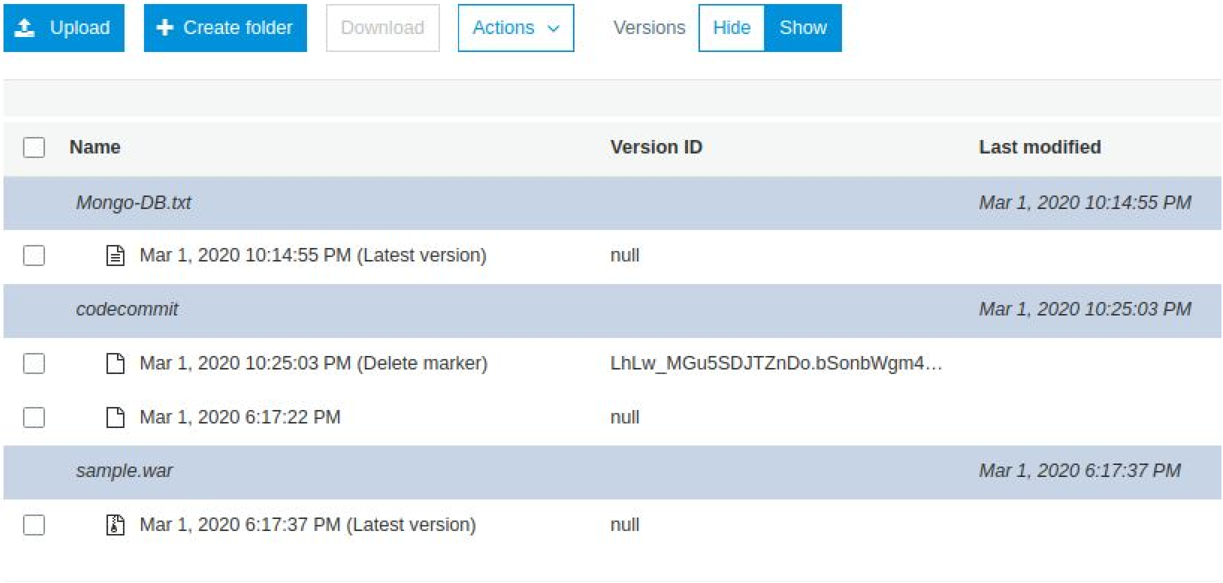
Learn more

● Enabled

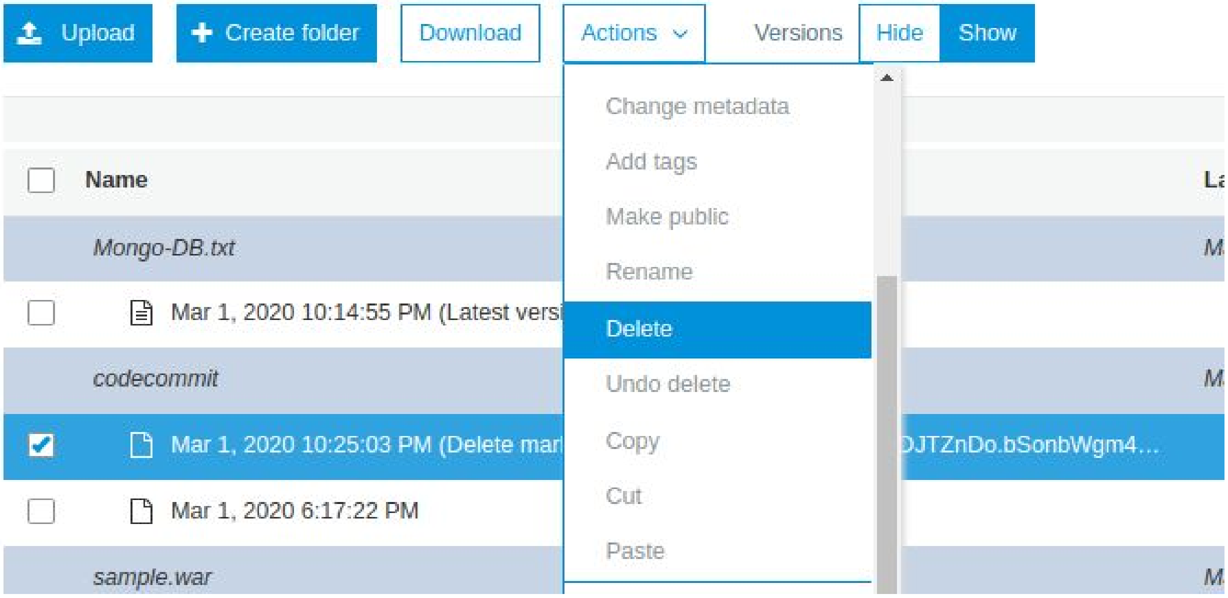




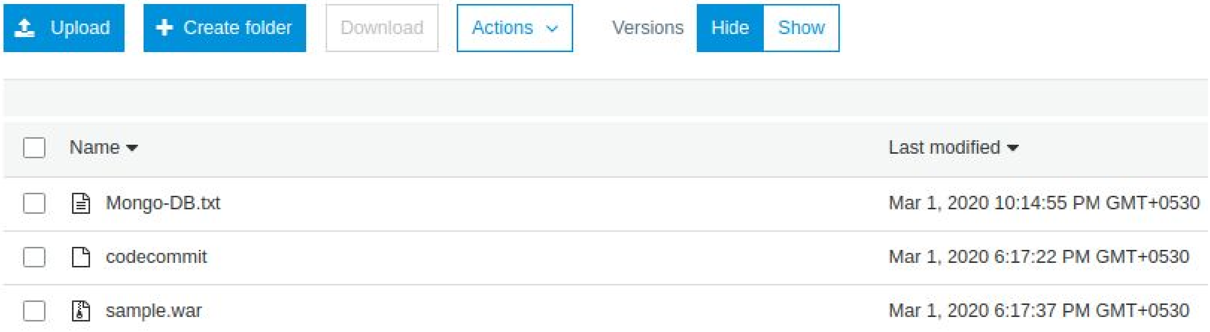
□	Name	Last modified
□	Mongo-DB.txt	Mar 1, 2020 10:14:55 P
□	sample.war	Mar 1, 2020 6:17:37 PM



□	Name	Version ID	Last modified
	Mongo-DB.txt		*Mar 1, 2020 10:14:55 PM*
□	Mar 1, 2020 10:14:55 PM (Latest version)	null	
	codecommit		*Mar 1, 2020 10:25:03 PM*
□	Mar 1, 2020 10:25:03 PM (Delete marker)	LhLw_MGu5SDJTZnDo.bSonbWgm4…	
□	Mar 1, 2020 6:17:22 PM	null	
	sample.war		*Mar 1, 2020 6:17:37 PM*
□	Mar 1, 2020 6:17:37 PM (Latest version)	null	

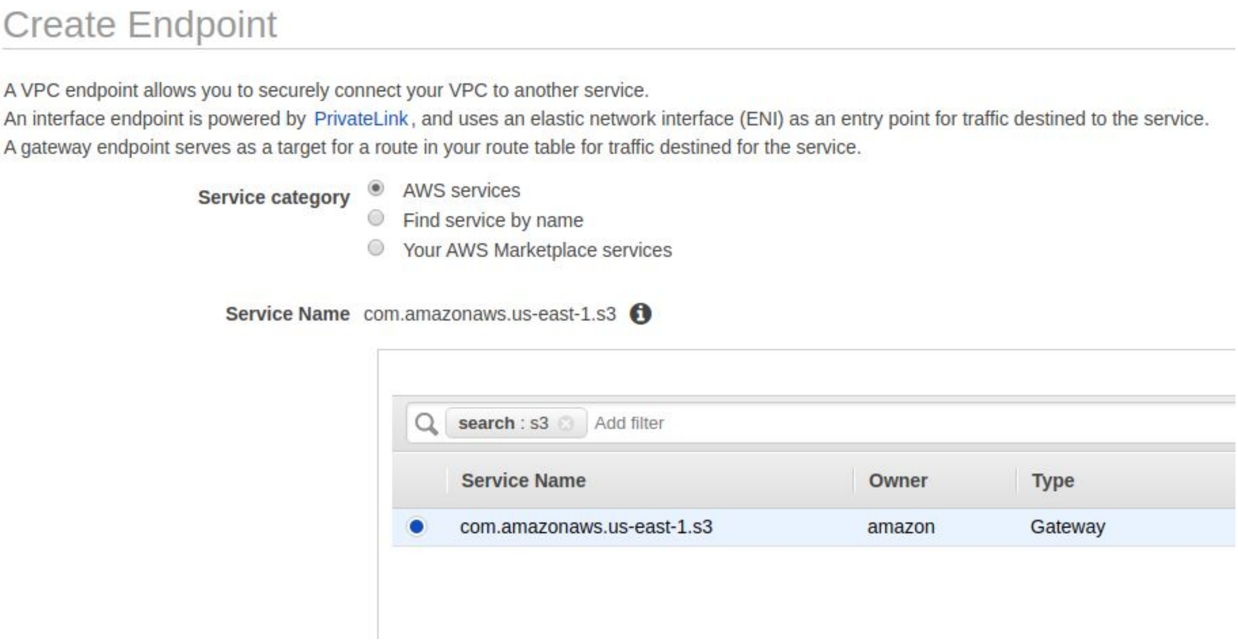


□	Name		La
	Mongo-DB.txt		*M*
□	Mar 1, 2020 10:14:55 PM (Latest versi		
	codecommit		*M*
☑	Mar 1, 2020 10:25:03 PM (Delete mar	DJTZnDo.bSonbWgm4…	
□	Mar 1, 2020 6:17:22 PM		
	sample.war		*M*



□	Name	Last modified
□	Mongo-DB.txt	Mar 1, 2020 10:14:55 PM GMT+0530
□	codecommit	Mar 1, 2020 6:17:22 PM GMT+0530
□	sample.war	Mar 1, 2020 6:17:37 PM GMT+0530

9. S3 VPC endpoint.



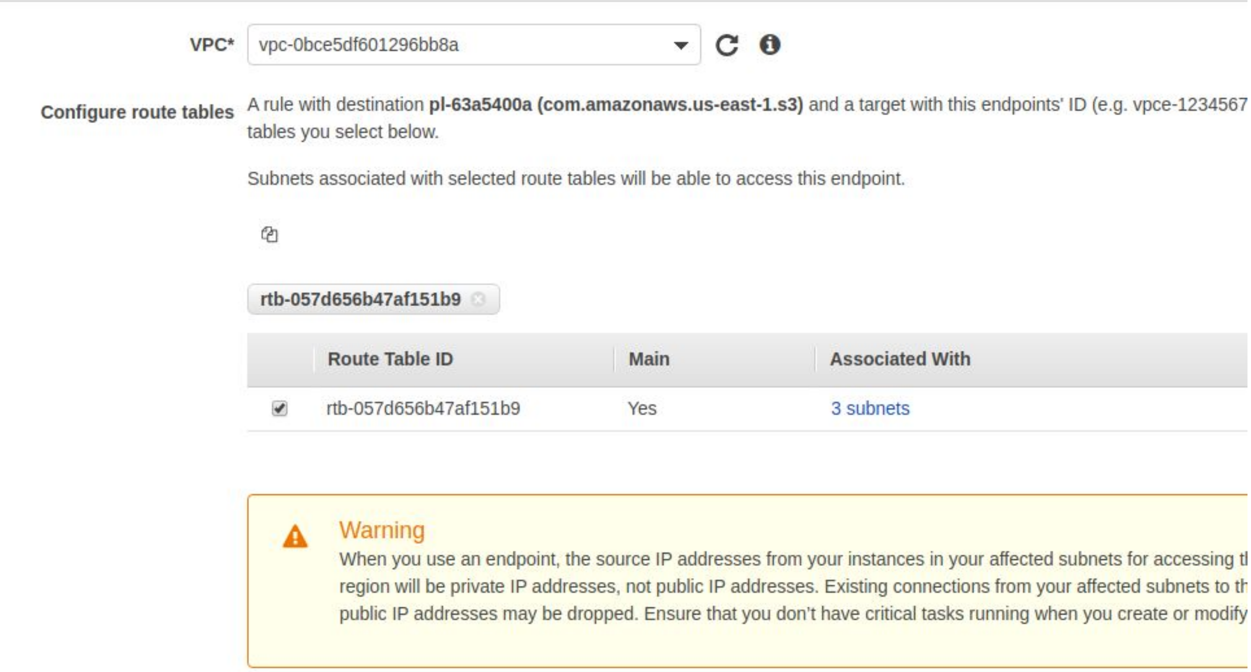
Create Endpoint

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

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



VPC* vpc-0bce5df601296bb8a

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-1234567
tables you select below.

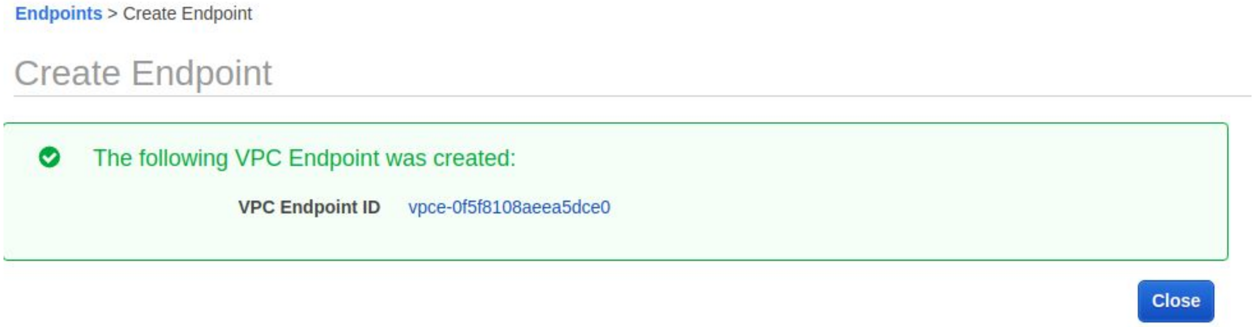
Subnets associated with selected route tables will be able to access this endpoint.

rtb-057d656b47af151b9

	Route Table ID	Main	Associated With
☑	rtb-057d656b47af151b9	Yes	3 subnets

Warning

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



Endpoints > Create Endpoint

Create Endpoint

The following VPC Endpoint was created:

VPC Endpoint ID vpce-0f5f8108aeea5dce0

10. CORS, Enable CORS for 2 specific website.

Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration
xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
 <CORSRule>
   <AllowedOrigin>https://website-1.com</AllowedOrigin>
   <AllowedOrigin>https://website-2.com</AllowedOrigin>
   <AllowedMethod>GET</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>PUT</AllowedMethod>
   <MaxAgeSeconds>3000</MaxAgeSeconds>
   <AllowedHeader>Authorization</AllowedHeader>
 </CORSRule>
</CORSConfiguration>
```