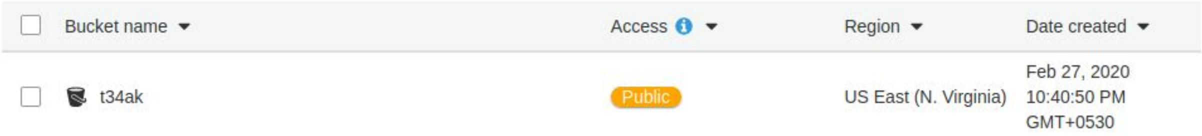
S3 , Route 53 & DNS

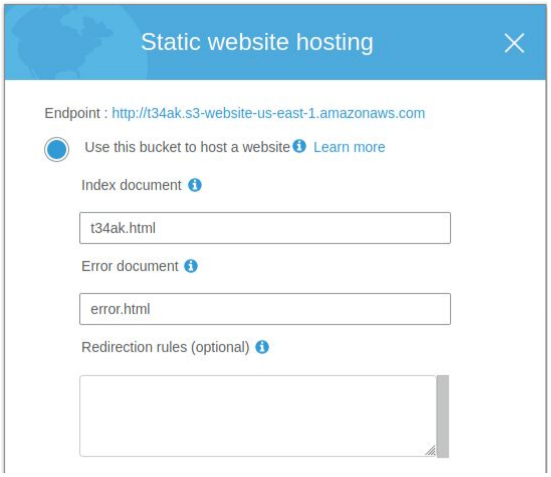
Ques 1:- Static website hosting using S3 (index page, error page)

Ans 1:- First we have to create the Bucket in the S3



□	Bucket name ▾	Access ℹ ▾	Region ▾	Date created ▾
□	t34ak	Public	US East (N. Virginia)	Feb 27, 2020 10:40:50 PM GMT+0530

Then we have to give the permissions to the bucket for making everything public



Static website hosting

Endpoint : http://t34ak.s3-website-us-east-1.amazonaws.com

● Use this bucket to host a website ℹ Learn more

Index document ℹ

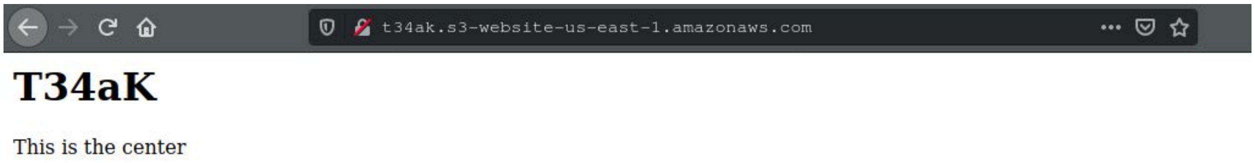
t34ak.html

Error document ℹ

error.html

Redirection rules (optional) ℹ

Just paste the s3 bucket url in browser it will loads up the file

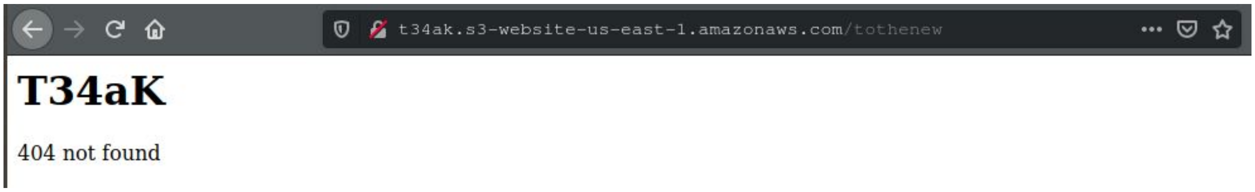


t34ak.s3-website-us-east-1.amazonaws.com

T34aK

This is the center

If we type any anonymous url then it will show the error page



T34aK

404 not found

Ques 2:- create an assumed role to access s3 using EC2.

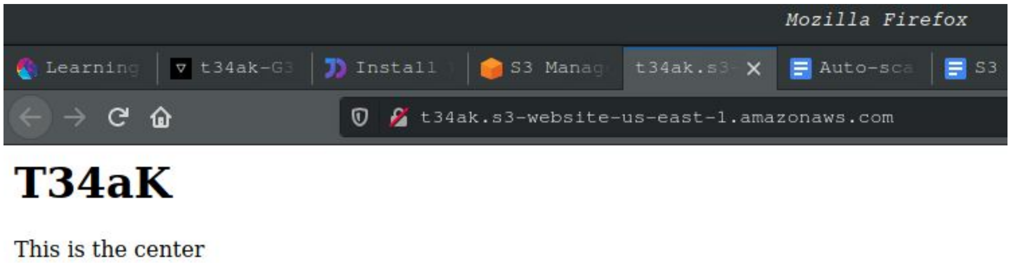
Ans 2:-

Ques 3:-Block s3 access on the basis of

On basis IP-Address

Ans 3:- first we have to add and edit the Bucket policy

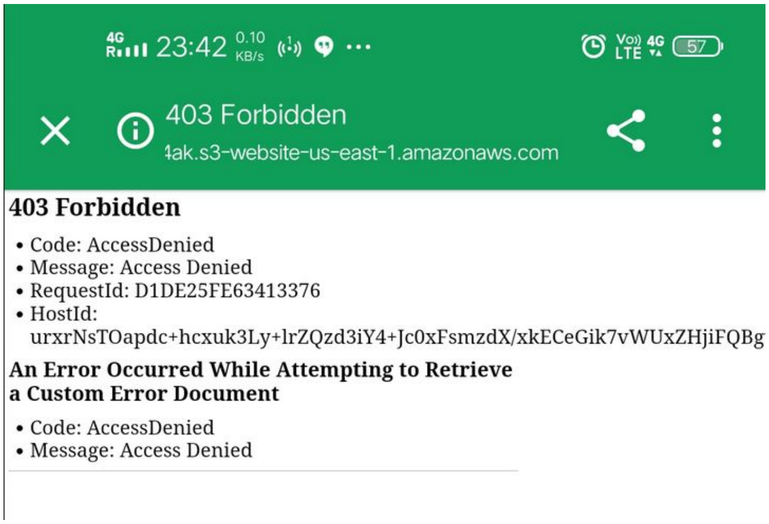
```
{
      "Version": "2012-10-17",
      "Id": "VPCe and SourceIP",
      "Statement": [
      {
            "Sid": "VPCe and SourceIP",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
            "arn:aws:s3:::t34ak",
            "arn:aws:s3:::t34ak/*"
            ],
            "IpAddress": {
                  "aws:SourceIp": "103.83.127.16"
            }
            }
      }
      ]
}
```



T34aK

This is the center

Picture of a mobile which is on another network



403 Forbidden

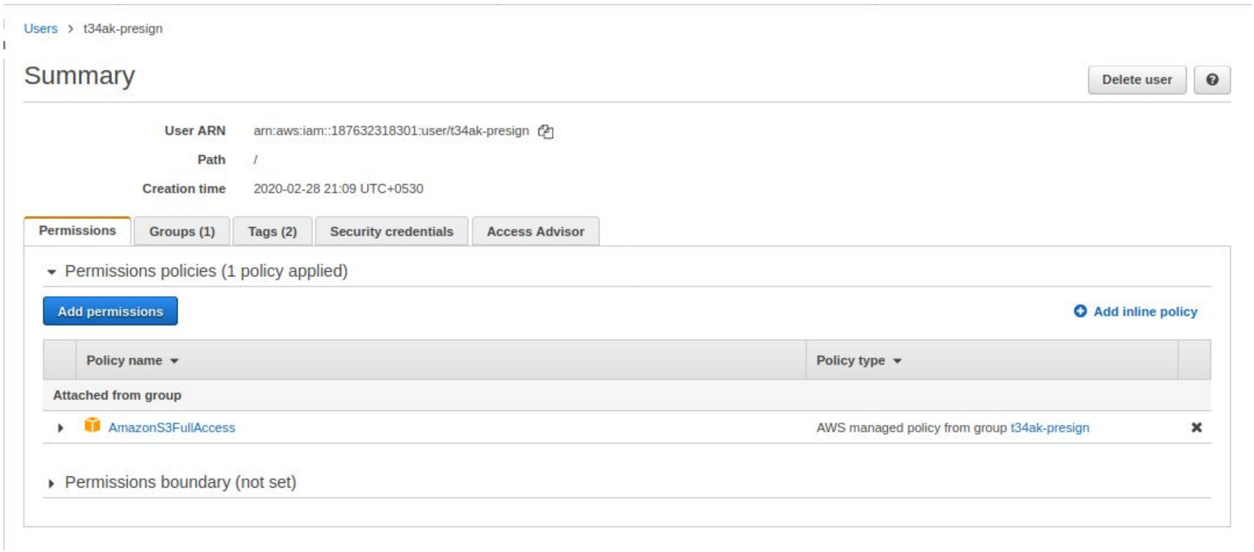
- Code: AccessDenied
- Message: Access Denied
- RequestId: D1DE25FE63413376
- HostId: urxrNsTOapdc+hcxuk3Ly+lrZQzd3iY4+Jc0xFsmzdX/xkECeGik7vWUxZHjiFQBg

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

On basis presign URL

To create the presign url first we have to generate and download the credentials of the account



Users > t34ak-presign

Summary

Delete user

User ARN arn:aws:iam::187632318301:user/t34ak-presign

Path /

Creation time 2020-02-28 21:09 UTC+0530

Permissions | Groups (1) | Tags (2) | Security credentials | Access Advisor

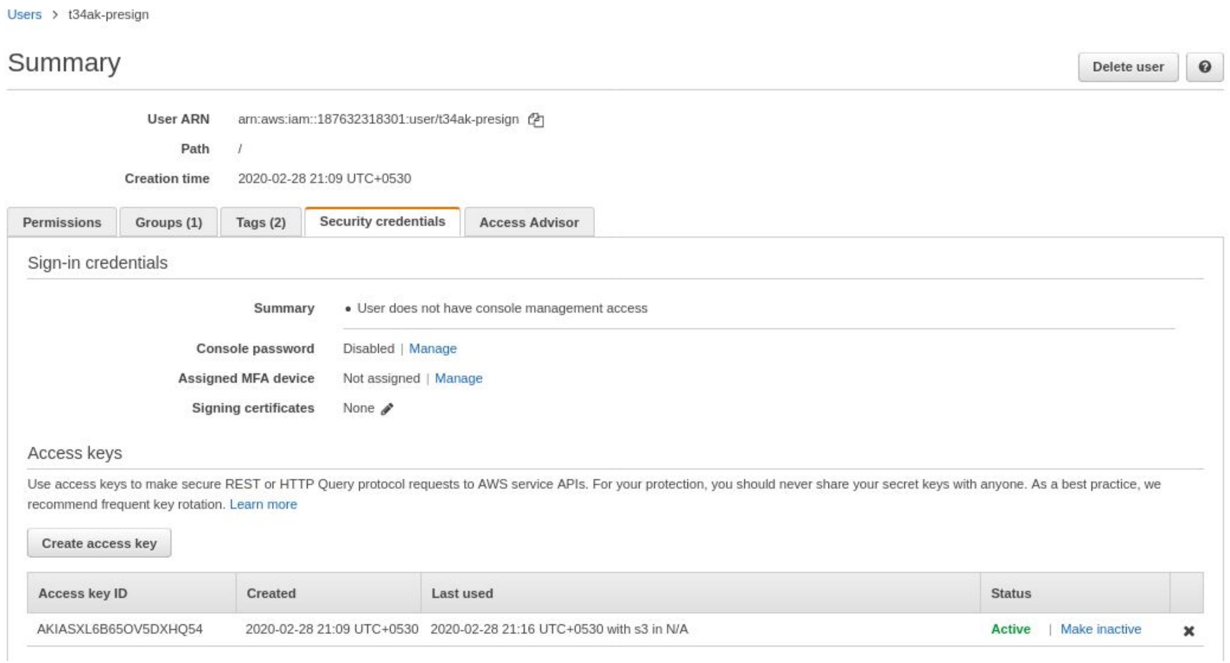
Permissions policies (1 policy applied)

Add permissions

Add inline policy

	Policy name	Policy type	
Attached from group			
▸	AmazonS3FullAccess	AWS managed policy from group t34ak-presign	✖

Permissions boundary (not set)



Users > t34ak-presign

Summary

Delete user

User ARN arn:aws:iam::187632318301:user/t34ak-presign

Path /

Creation time 2020-02-28 21:09 UTC+0530

Permissions | Groups (1) | Tags (2) | Security credentials | Access Advisor

Sign-in credentials

Summary • User does not have console management access

Console password Disabled | Manage

Assigned MFA device Not assigned | Manage

Signing certificates None

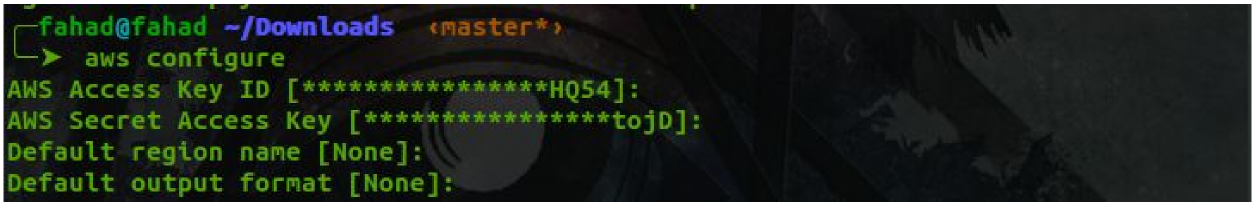
Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

Create access key

Access key ID	Created	Last used	Status	
AKIASXL6B65OV5DXHQ54	2020-02-28 21:09 UTC+0530	2020-02-28 21:16 UTC+0530 with s3 in N/A	Active | Make inactive	✖

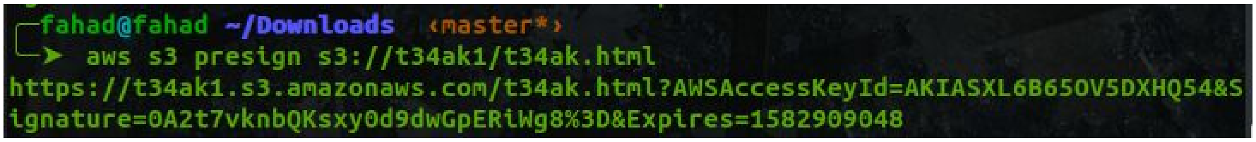
Now on console first we have to do aws configure to set our credentials



```
┌─fahad@fahad ~/Downloads  ‹master*›
└─➤ aws configure
AWS Access Key ID [****************HQ54]:
AWS Secret Access Key [****************tojD]:
Default region name [None]:
Default output format [None]:
```

Now we have to generate the url by

aws s3 presign s3://t34ak1/t34ak.html



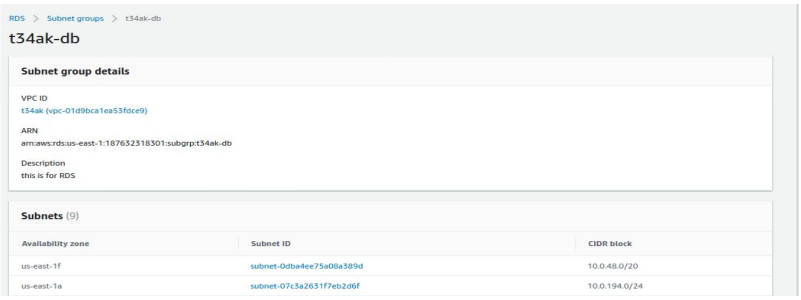
```
┌─fahad@fahad ~/Downloads  ‹master*›
└─➤ aws s3 presign s3://t34ak1/t34ak.html
https://t34ak1.s3.amazonaws.com/t34ak.html?AWSAccessKeyId=AKIASXL6B65OV5DXHQ54&S
ignature=0A2t7vknbQKsxy0d9dwGpERiWg8%3D&Expires=1582909048
```

On basis of domain

Ques 3:-

Ans 3:-

First we have to create the RDS subnet then we have to make the db



t34ak-db

Subnet group details

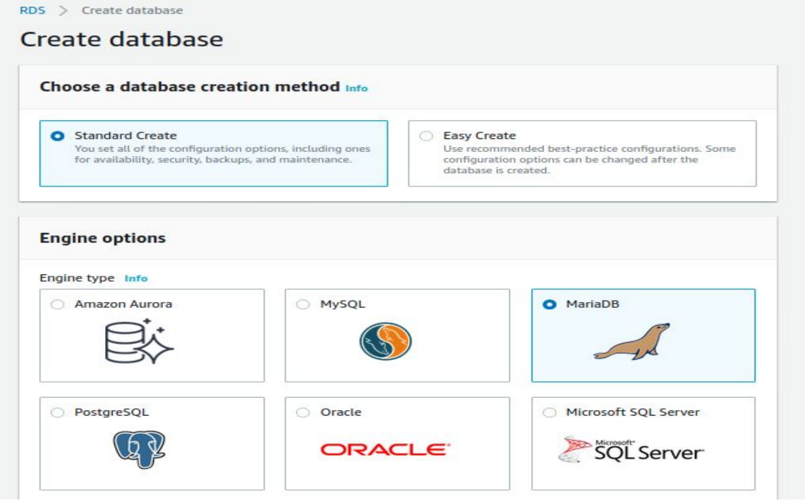
VPC ID
t34ak (vpc-01d9bca1ea53fdce9)

ARN
arn:aws:rds:us-east-1:187632318301:subgrp:t34ak-db

Description
this is for 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



RDS > Create database

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

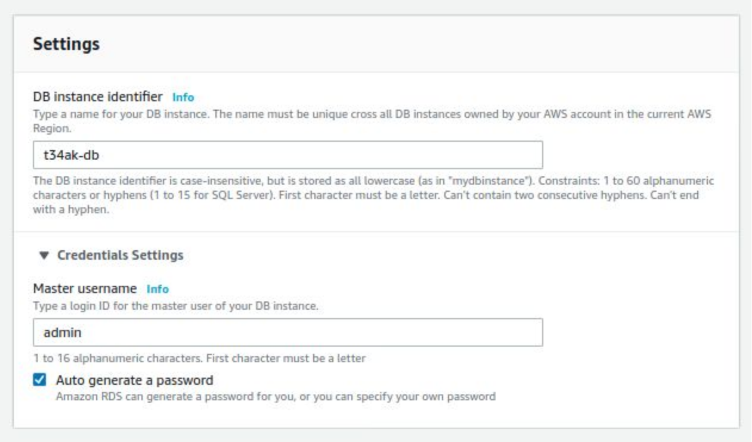
○ MySQL

● MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server



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.

t34ak-db

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

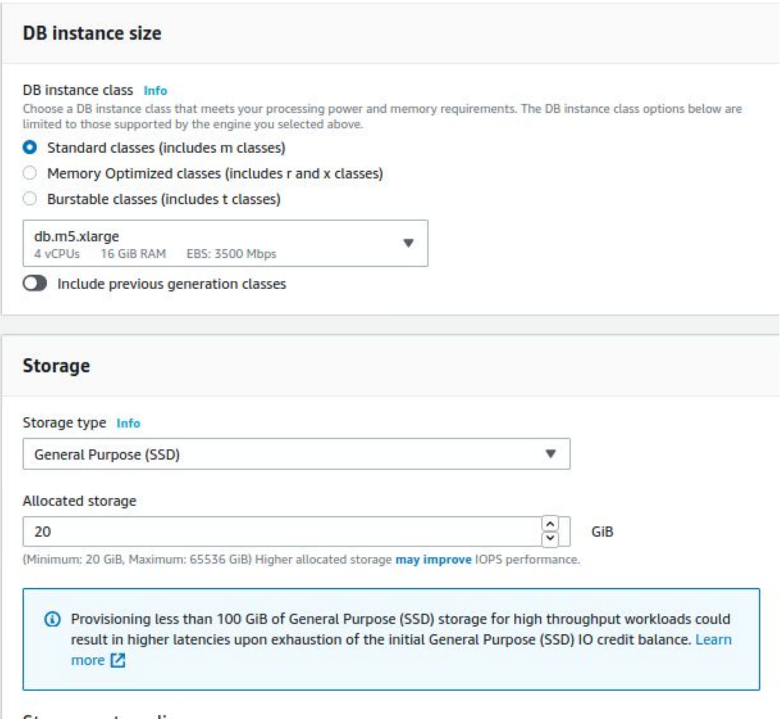
Master username Info
Type a login ID for the master user of your DB instance.

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

Now we have to specify the DB size
And storage specifications and choose the vpc and the subnets that we make



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.m5.xlarge
4 vCPUs 16 GiB RAM EBS: 3500 Mbps

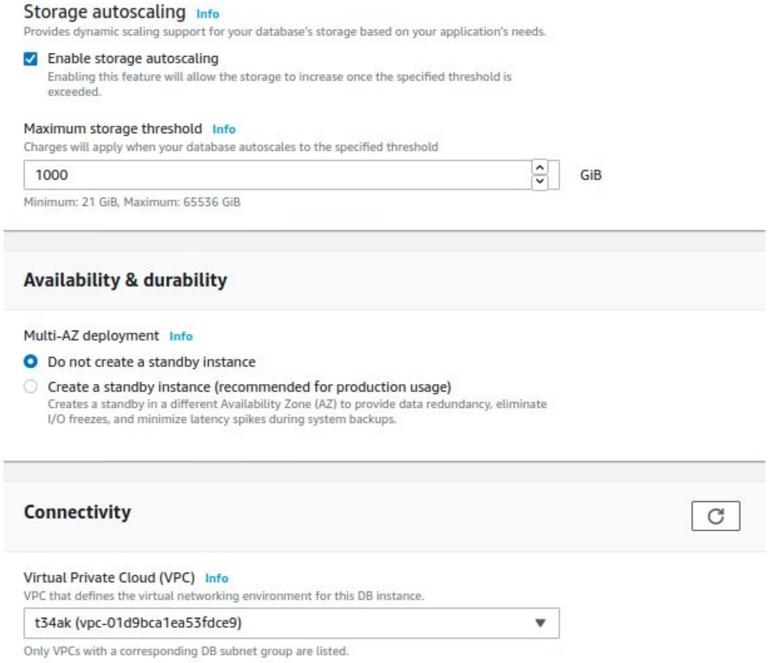
Include previous generation classes

Storage

Storage type Info
General Purpose (SSD)

Allocated storage
20 GiB
(Minimum: 20 GiB, Maximum: 65536 GiB) Higher allocated storage **may improve** IOPS performance.

Provisioning less than 100 GiB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Learn more



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: 65536 GiB

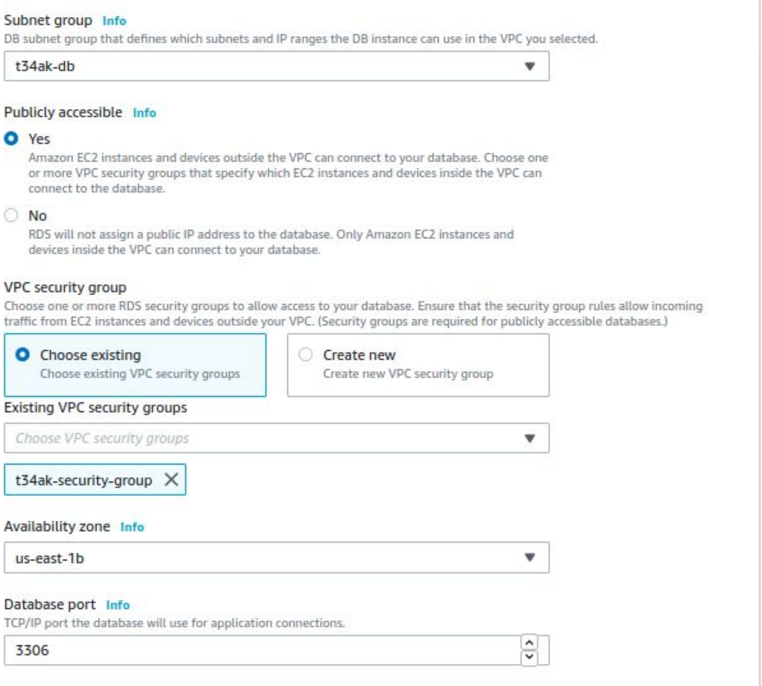
Availability & durability

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.
t34ak (vpc-01d9bca1ea53fdce9)
Only VPCs with a corresponding DB subnet group are listed.

Now we have to specify the subnet and the vpc



Subnet group Info

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

t34ak-db

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

t34ak-security-group ✕

Availability zone Info

us-east-1b

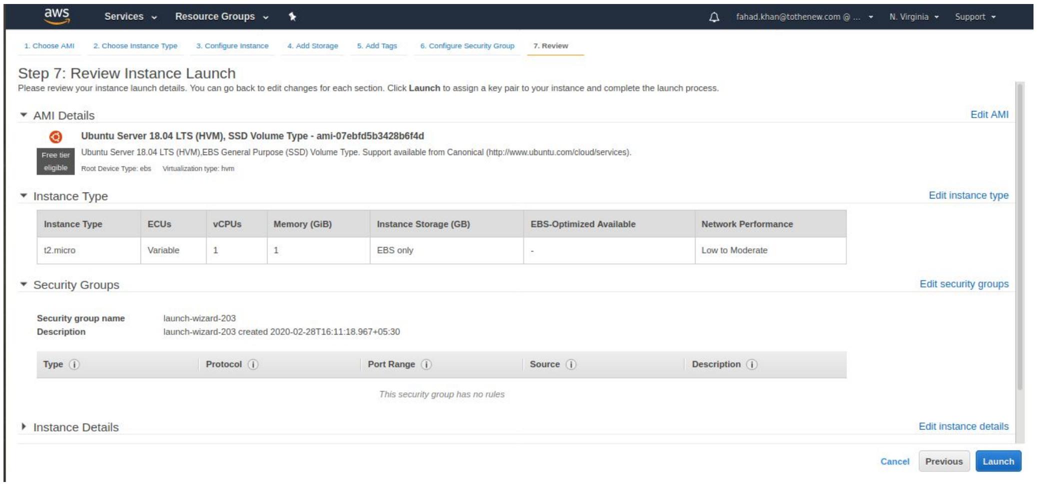
Database port Info

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

3306

Ques 4:-Mount S3 to an EC2 instance

Ans 4:- first we have to create the ec2 instance



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details — Edit AMI

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-07ebfd5b3428b6f4d

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root Device Type: ebs   Virtualization type: hvm

Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups — Edit security groups

Security group name launch-wizard-203
Description launch-wizard-203 created 2020-02-28T16:11:18.967+05:30

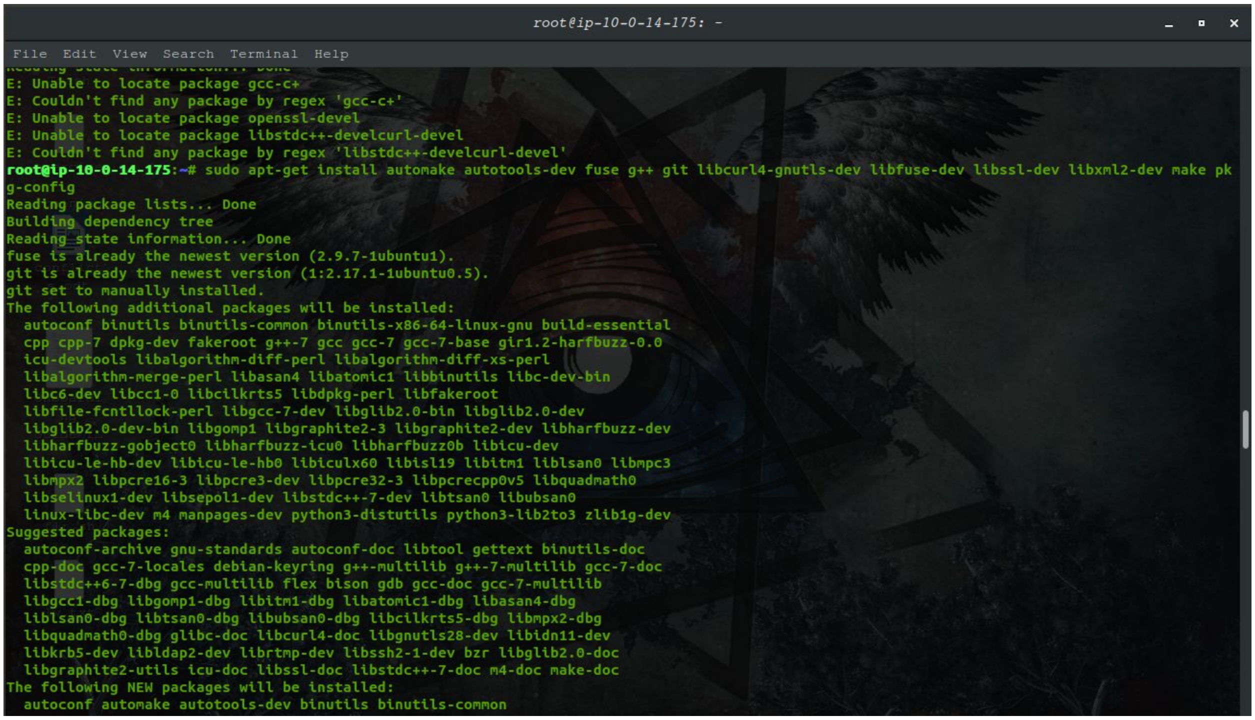
Type	Protocol	Port Range	Source	Description

This security group has no rules

Instance Details — Edit instance details

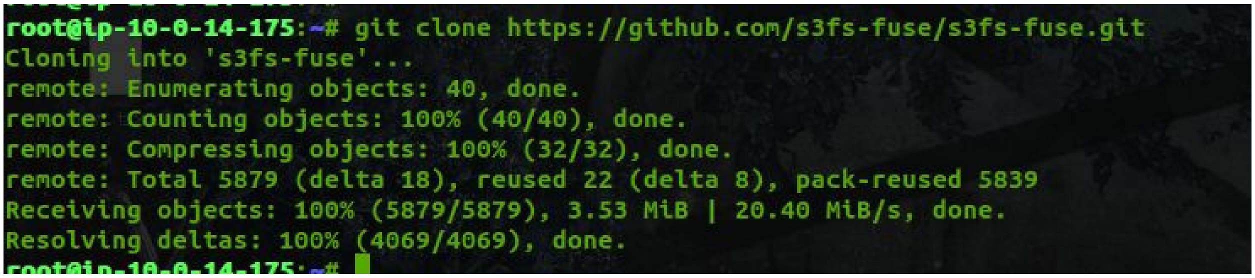
Now we have to ssh to that instance and install the following dependencies:-

```
sudo apt-get install automake autotools-dev fuse g++ git
libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config
```



```
Reading state information... Done
E: Unable to locate package gcc-c+
E: Couldn't find any package by regex 'gcc-c+'
E: Unable to locate package openssl-devel
E: Unable to locate package libstdc++-develcurl-devel
E: Couldn't find any package by regex 'libstdc++-develcurl-devel'
root@ip-10-0-14-175:~# sudo apt-get install automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pk
g-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
fuse is already the newest version (2.9.7-1ubuntu1).
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 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 libglib2.0-bin libglib2.0-dev
  libglib2.0-dev-bin libgomp1 libgraphite2-3 libgraphite2-dev libharfbuzz-dev
  libharfbuzz-gobject0 libharfbuzz-icu0 libharfbuzz0b libicu-dev
  libicu-le-hb-dev libicu-le-hb0 libiculx60 libisl19 libitm1 liblsan0 libmpc3
  libmpx2 libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5 libquadmath0
  libselinux1-dev libsepol1-dev libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev m4 manpages-dev python3-distutils python3-lib2to3 zlib1g-dev
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc libtool gettext binutils-doc
  cpp-doc gcc-7-locales debian-keyring g++-multilib g++-7-multilib gcc-7-doc
  libstdc++6-7-dbg gcc-multilib flex bison gdb gcc-doc gcc-7-multilib
  libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan4-dbg
  liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libmpx2-dbg
  libquadmath0-dbg glibc-doc libcurl4-doc libgnutls28-dev libidn11-dev
  libkrb5-dev libldap2-dev librtmp-dev libssh2-1-dev bzr libglib2.0-doc
  libgraphite2-utils icu-doc libssl-doc libstdc++-7-doc m4-doc make-doc
The following NEW packages will be installed:
  autoconf automake autotools-dev binutils binutils-common
```

Clone s3fs source code from git

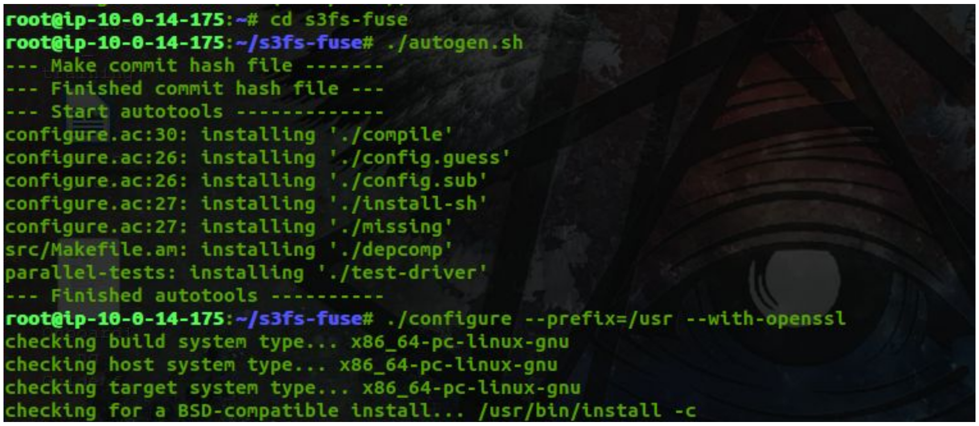


```
root@ip-10-0-14-175:~# git clone https://github.com/s3fs-fuse/s3fs-fuse.git
Cloning into 's3fs-fuse'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 5879 (delta 18), reused 22 (delta 8), pack-reused 5839
Receiving objects: 100% (5879/5879), 3.53 MiB | 20.40 MiB/s, done.
Resolving deltas: 100% (4069/4069), done.
root@ip-10-0-14-175:~#
```

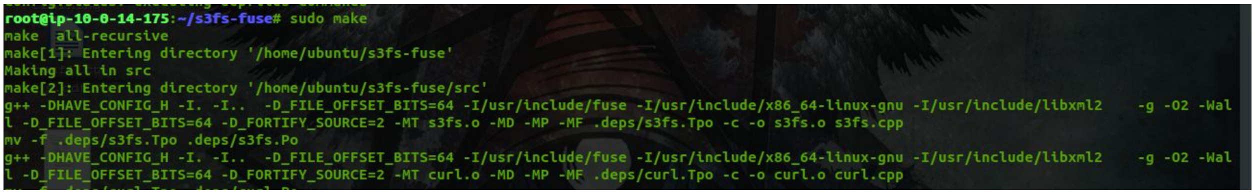
Now change to source code directory, and compile and install the code with the following commands:

```
cd s3fs-fuse
./autogen.sh
./configure --prefix=/usr --with-openssl
```

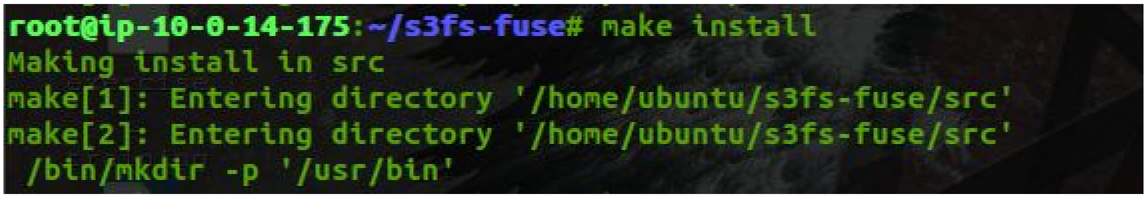
```
make
sudo make install
```



```
root@ip-10-0-14-175:~# cd s3fs-fuse
root@ip-10-0-14-175:~/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 ----------
root@ip-10-0-14-175:~/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
```



```
root@ip-10-0-14-175:~/s3fs-fuse# sudo 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 -Wal
l -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -MT 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/include/fuse -I/usr/include/x86_64-linux-gnu -I/usr/include/libxml2    -g -O2 -Wal
l -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -MT curl.o -MD -MP -MF .deps/curl.Tpo -c -o curl.o curl.cpp
```



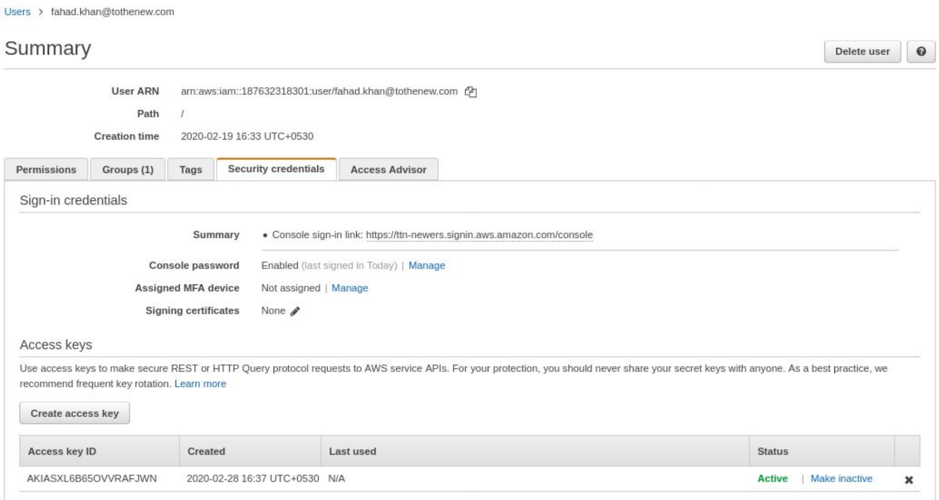
```
root@ip-10-0-14-175:~/s3fs-fuse# make install
Making install in src
make[1]: Entering directory '/home/ubuntu/s3fs-fuse/src'
make[2]: Entering directory '/home/ubuntu/s3fs-fuse/src'
 /bin/mkdir -p '/usr/bin'
```

Next step is to get the access key and secret key through AWS console

We first have to go to the IAM

Then we have to find the user

Then we have to generate the secret key and download it to the local host



Users > fahad.khan@tothenew.com

Summary

User ARN arn:aws:iam::187632318301:user/fahad.khan@tothenew.com

Path /

Creation time 2020-02-19 16:33 UTC+0530

Sign-in credentials

Summary • Console sign-in link: https://ttn-newers.signin.aws.amazon.com/console

Console password Enabled (last signed in Today) | Manage

Assigned MFA device Not assigned | Manage

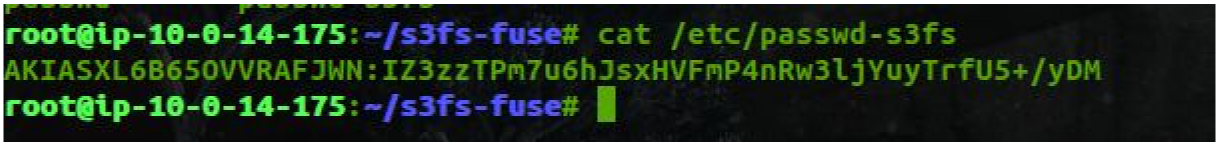
Signing certificates None

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

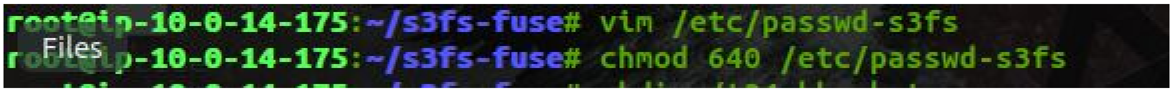
Access key ID	Created	Last used	Status	
AKIASXL6B65OVVRAFJWN	2020-02-28 16:37 UTC+0530	N/A	Active | Make inactive	✖

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



```
root@ip-10-0-14-175:~/s3fs-fuse# cat /etc/passwd-s3fs
AKIASXL6B65OVVRAFJWN:IZ3zzTPm7u6hJsxHVFmP4nRw3ljYuyTrfU5+/yDM
root@ip-10-0-14-175:~/s3fs-fuse#
```

Now change the permission of file.
chmod 640 /etc/passwd-s3fs

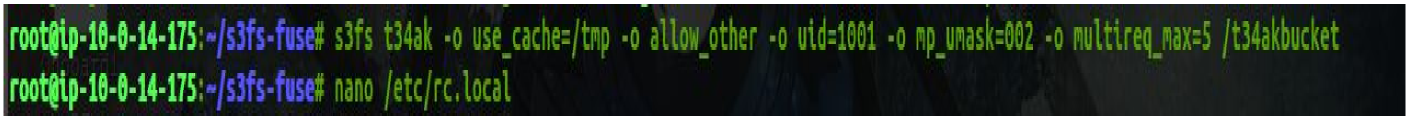


```
root@ip-10-0-14-175:~/s3fs-fuse# vim /etc/passwd-s3fs
root@ip-10-0-14-175:~/s3fs-fuse# chmod 640 /etc/passwd-s3fs
```

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

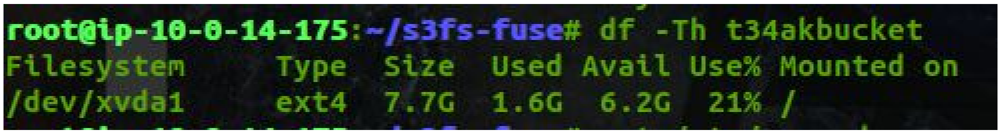
s3fs t34ak -o use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /t34akbucket

Now we have to make the entry of this in the rc.local so that it will be mounted on every restart



```
root@ip-10-0-14-175:~/s3fs-fuse# s3fs t34ak -o use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /t34akbucket
root@ip-10-0-14-175:~/s3fs-fuse# nano /etc/rc.local
```

Now we have to check the mounted bucket by df -Th t34akbucket command



```
root@ip-10-0-14-175:~/s3fs-fuse# df -Th t34akbucket
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/xvda1     ext4  7.7G  1.6G  6.2G  21% /
```

It shows that it is mounted on the system

Ques 5.what is parameter group and option group

Ans5. 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 6. ACL, Bucket policy, IAM Policy.

Ans 6. 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.

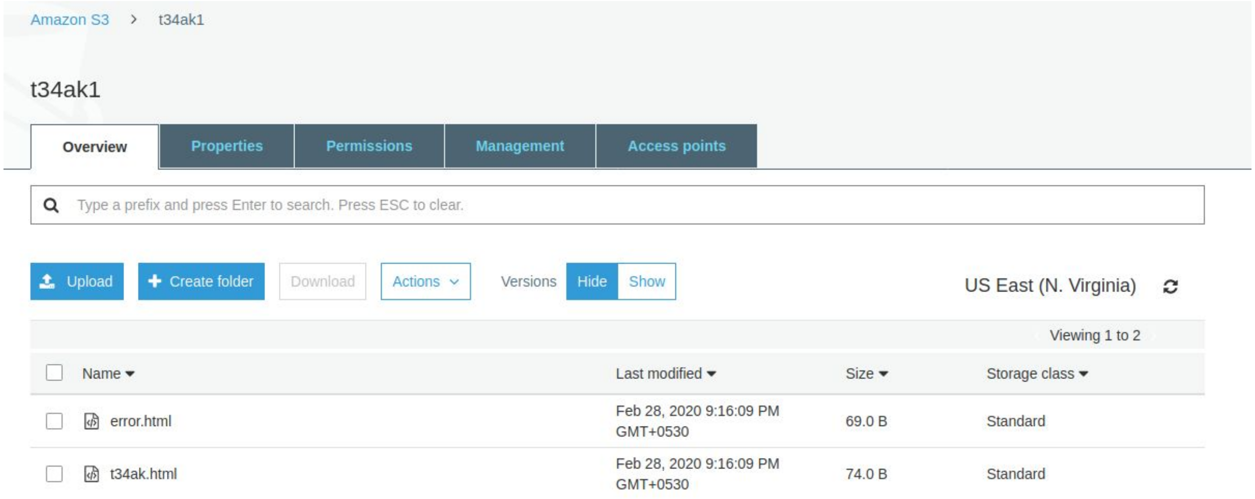
An S3 ACL is a sub-resource that’s attached to every S3 bucket and object. It defines which AWS accounts or groups are granted access and the type of access. When you create a bucket or an object, Amazon S3 creates a default ACL that grants the resource owner full control over the resource.

Ques 7:-Change content type using S3, PDF rendering and downloading.

Ans 7:-

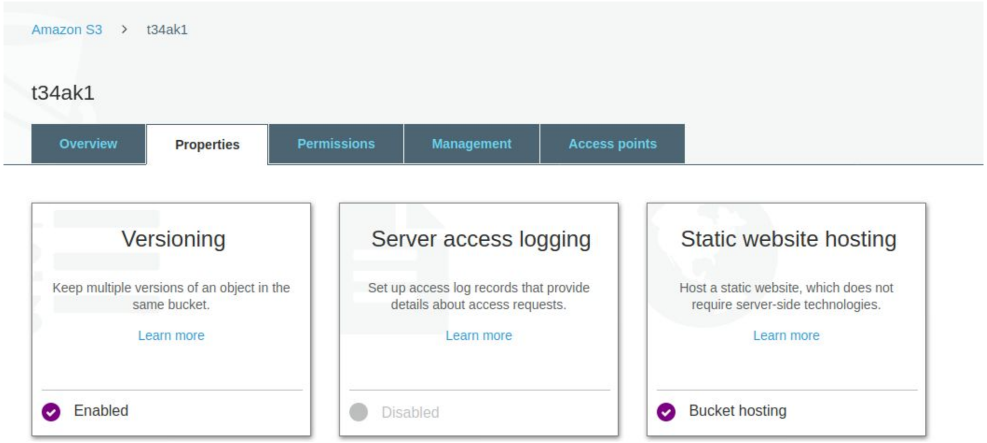
Ques 8:-Retrieve previous version of S3, enable versioning.

Ans 8:- First we have to make a bucket then we have to



Name	Last modified	Size	Storage class
error.html	Feb 28, 2020 9:16:09 PM GMT+0530	69.0 B	Standard
t34ak.html	Feb 28, 2020 9:16:09 PM GMT+0530	74.0 B	Standard

Then we have to enable the versoning under properties tab



Versioning

Keep multiple versions of an object in the same bucket.

Enabled

Server access logging

Set up access log records that provide details about access requests.

Disabled

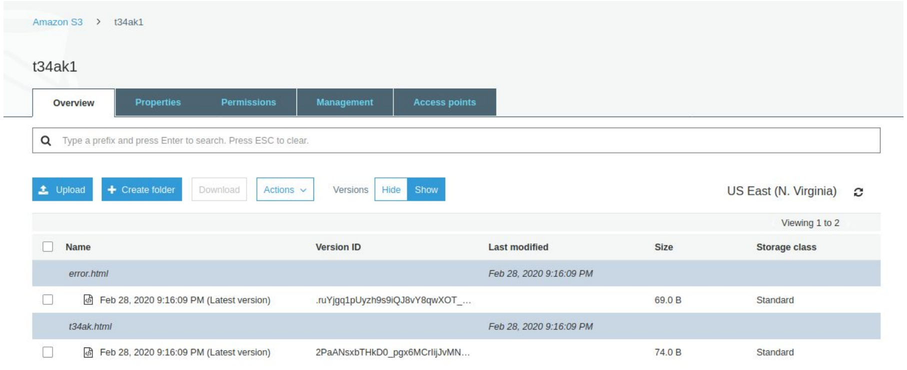
Static website hosting

Host a static website, which does not require server-side technologies.

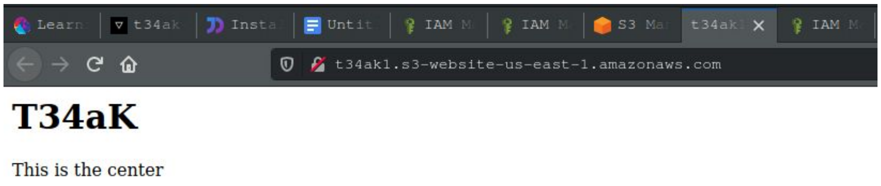
Bucket hosting

Now to check the versioning is working or not we have to modify the t34ak.html page then re upload it.

Before uploading the modified version



Name	Version ID	Last modified	Size	Storage class
error.html		*Feb 28, 2020 9:16:09 PM*		
□ Feb 28, 2020 9:16:09 PM (Latest version)	.ruYjgq1pUyzh9s9iQJ8vY8qwXOT_...		69.0 B	Standard
t34ak.html		*Feb 28, 2020 9:16:09 PM*		
□ Feb 28, 2020 9:16:09 PM (Latest version)	2PaANsxbTHkD0_pgx6MCrlijJvMN...		74.0 B	Standard

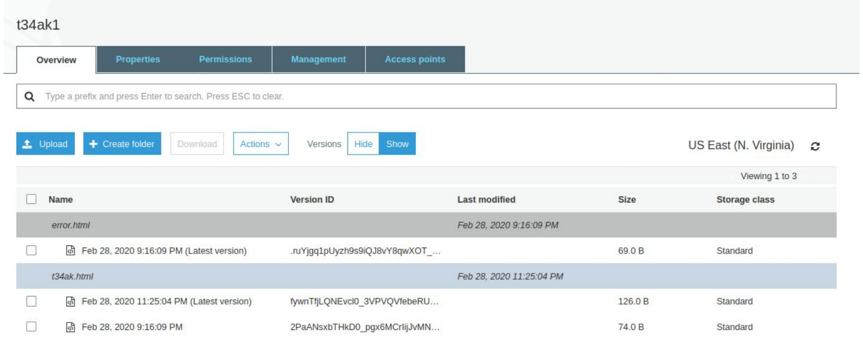


t34ak1.s3-website-us-east-1.amazonaws.com

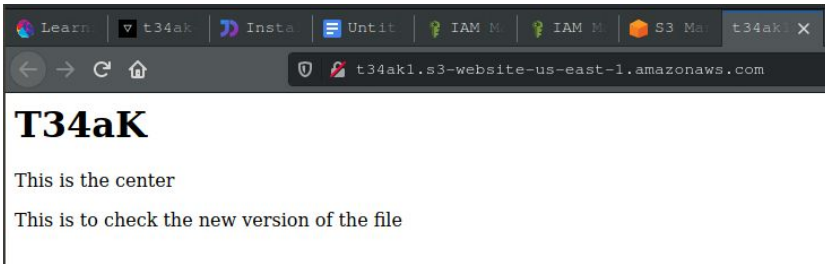
T34aK

This is the center

After editing and reuploading the t34ak.html page



□	Name	Version ID	Last modified	Size	Storage class
	error.html		*Feb 28, 2020 9:16:09 PM*		
□	Feb 28, 2020 9:16:09 PM (Latest version)	.ruYjgq1pUyzh9s9iQJ8vY8qwXOT_...		69.0 B	Standard
	t34ak.html		*Feb 28, 2020 11:25:04 PM*		
□	Feb 28, 2020 11:25:04 PM (Latest version)	fywnTfjLQNEvcl0_3VPVQVfebeRU...		126.0 B	Standard
□	Feb 28, 2020 9:16:09 PM	2PaANsxbTHkD0_pgx6MCrIijJvMN...		74.0 B	Standard



T34aK

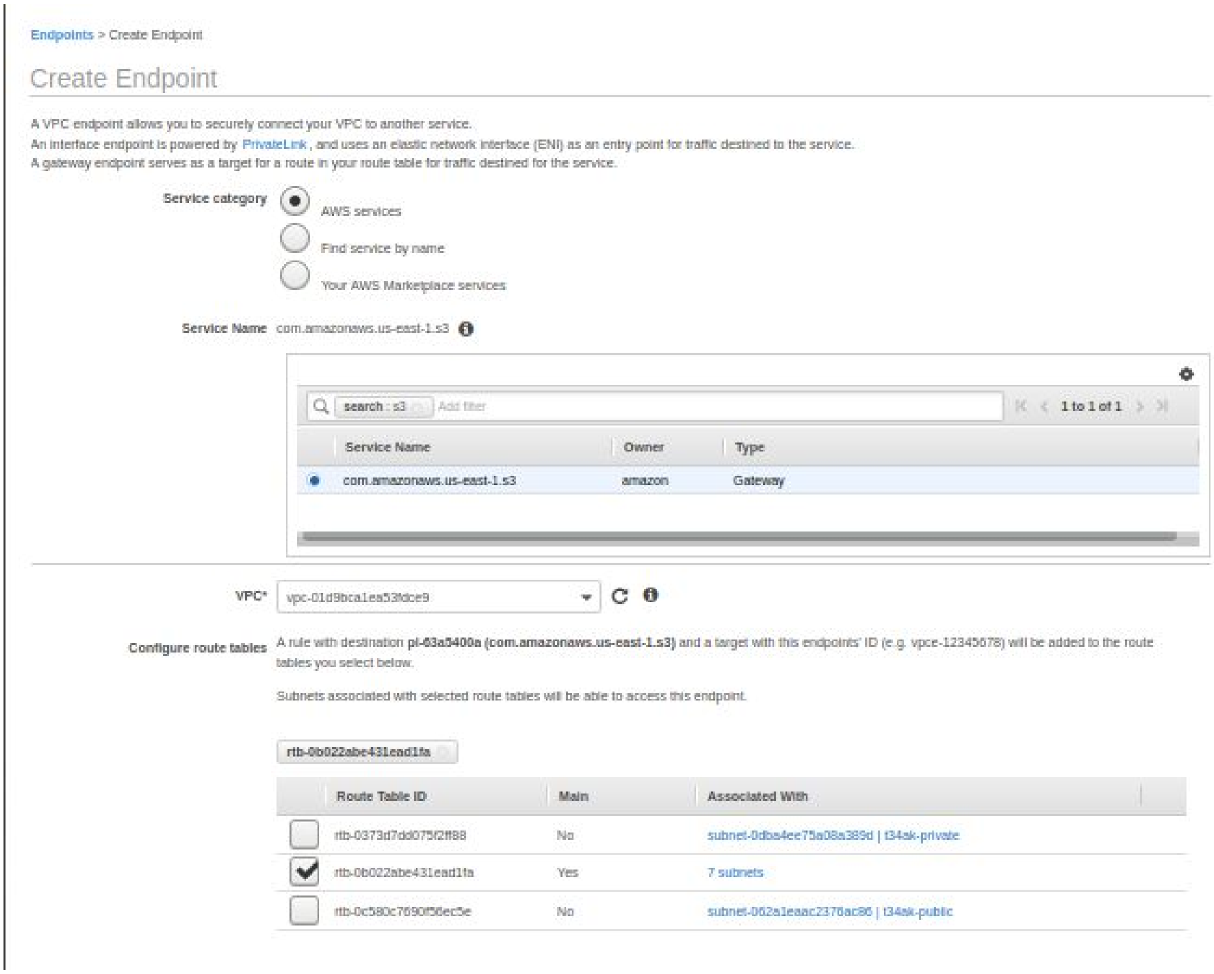
This is the center

This is to check the new version of the file

We can see that the updated version is detected and the page is also updated

Ques 9:-S3 VPC endpoint

Ans 9:- To create the s3 ENDPOINT we have to go on VPC service under which we have to chhoose the create endpoint



Endpoints > Create 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

search : s3 Add filter — 1 to 1 of 1

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

VPC* vpc-01d9bca1ea53fdce9

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.

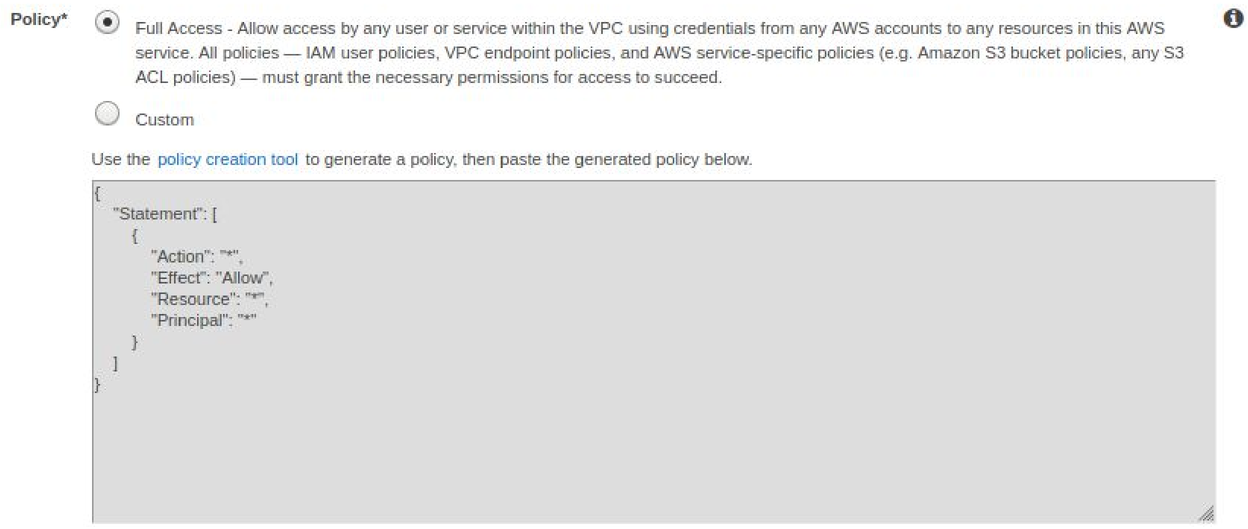
Subnets associated with selected route tables will be able to access this endpoint.

rtb-0b022abe431ead1fa

	Route Table ID	Main	Associated With
□	rtb-0373d7dd075f2ff88	No	subnet-0dba4ee75a08a389d | t34ak-private
☑	rtb-0b022abe431ead1fa	Yes	7 subnets
□	rtb-0c580c7690f56ec5e	No	subnet-062a1eaac2376ac86 | t34ak-public

Then we have to select the s3 service of which we have to make the endpoint

Now we are going to provide the full access to the policy for the ENDPOINT.

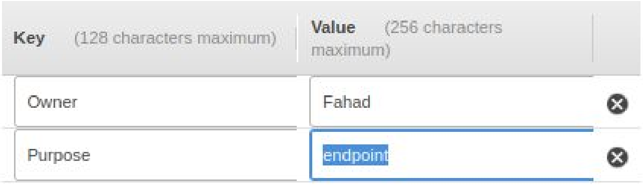


Policy* ● Full Access - Allow access by any user or service within the VPC using credentials from any AWS accounts to any resources in this AWS service. All policies — IAM user policies, VPC endpoint policies, and AWS service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must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

○ Custom

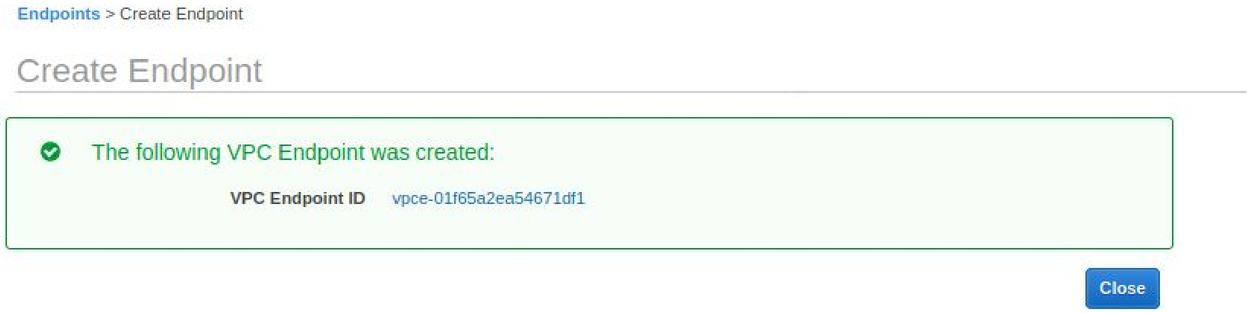
Use the policy creation tool to generate a policy, then paste the generated policy below.

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



Key (128 characters maximum)	Value (256 characters maximum)
Owner	Fahad
Purpose	endpoint

EPOINT si created now



Endpoints > Create Endpoint

Create Endpoint

The following VPC Endpoint was created:

VPC Endpoint ID vpce-01f65a2ea54671df1

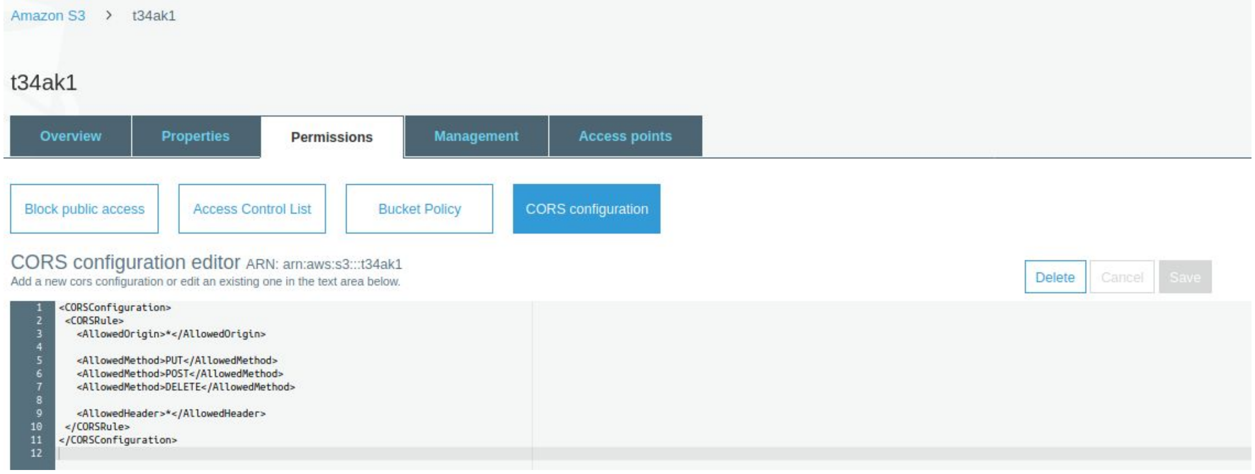
Ques 10:-CORS, enable CORS for a specific website

Ans 10:-CORS(Cross Origin Resource Sharing) is a way in which the applications that are loaded in one domain to interact with resources in a different domain.

To enable CORS we have to go to the bucket then we have to go to the permissions tab then select the CORS configuration

Now we have to specify the policy and in this policy i have listed that from origin is allowed from anywhere so it can access GET POST DELETE request from anywhere.

```
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
        <AllowedOrigin>*</AllowedOrigin>
        <AllowedMethod>PUT</AllowedMethod>
        <AllowedMethod>POST</AllowedMethod>
        <AllowedMethod>DELETE</AllowedMethod>
        <AllowedHeader>*</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```



t34ak1

CORS configuration editor ARN: arn:aws:s3:::t34ak1

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

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>*</AllowedOrigin>

   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>

   <AllowedHeader>*</AllowedHeader>
 </CORSRule>
</CORSConfiguration>
```