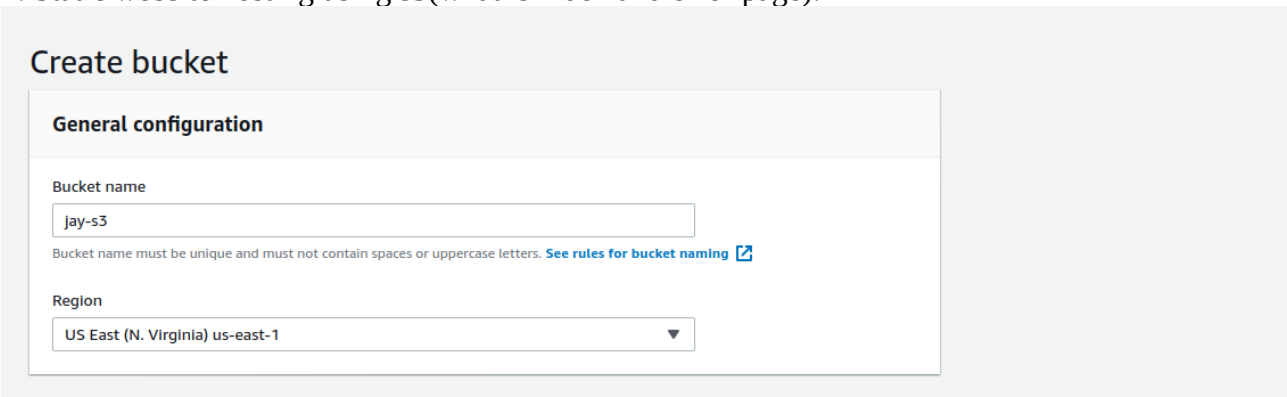
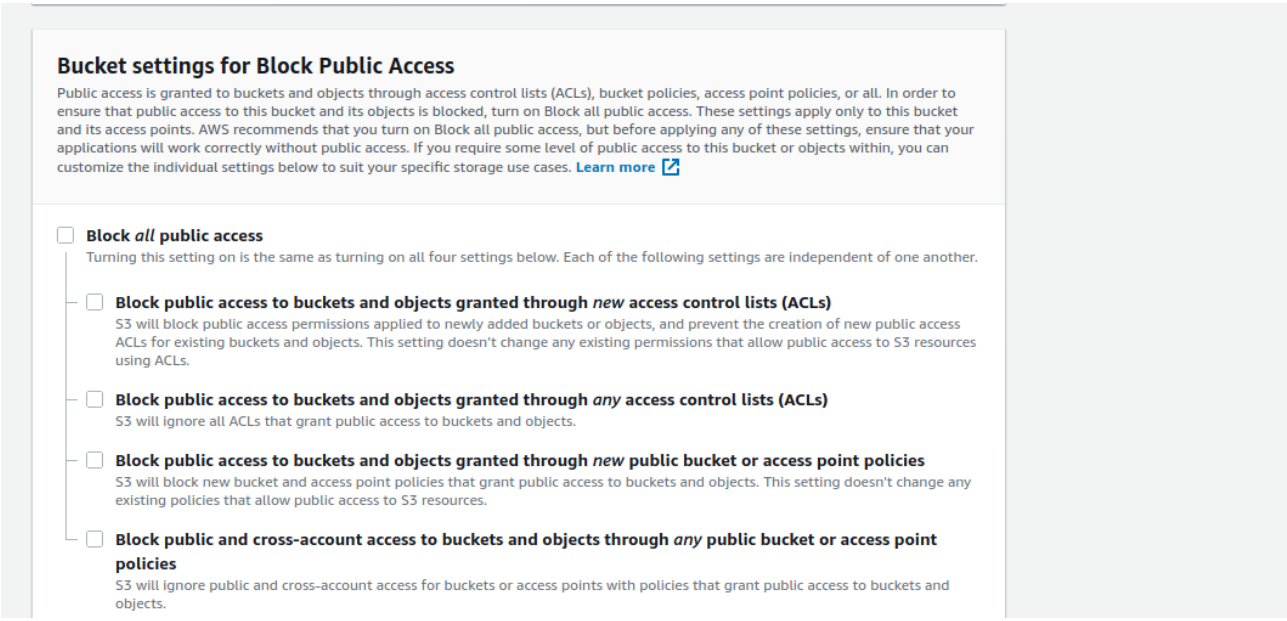
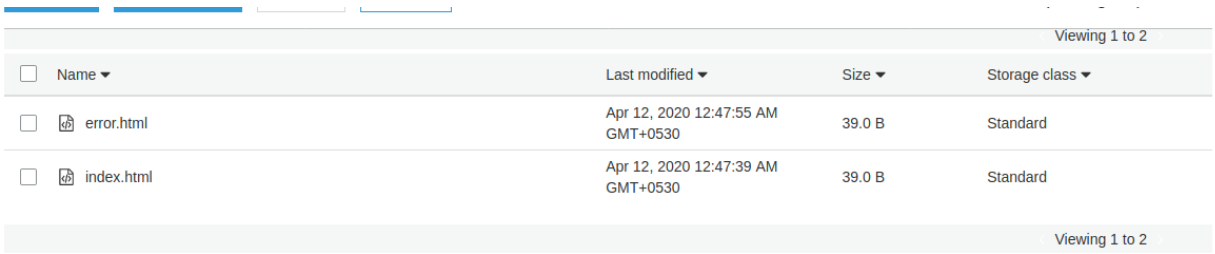
Doubt Resolving Session

1. Static website hosting using s3(what is index and error page).







Endpoint : http://jaypttn-s3.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

Bucket policy editor ARN: arn:aws:s3:::jaypttn-s3

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

Delete Cancel Save

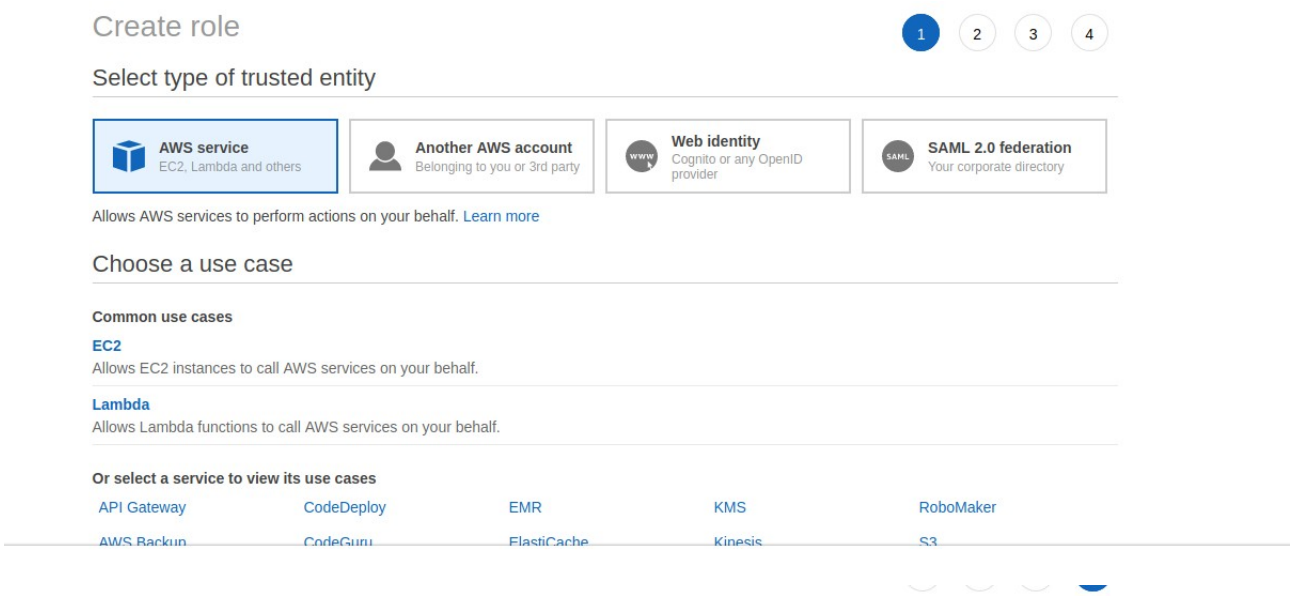
```
{
    "Id": "PolicyforWebsitecontent",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::jaypttn-s3/*"
        }
    ]
}
```

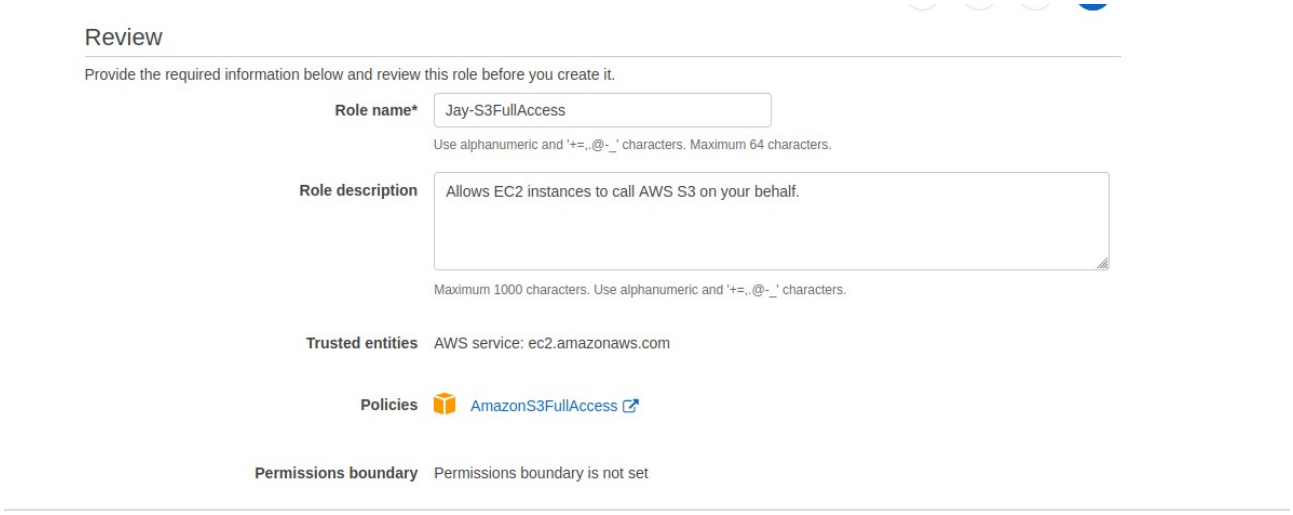
jaypttn-s3.s3-website-us-east-1.amazonaws.com

Hey index page of jaypttn-s3

Hey error page of jaypttn-s3

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





Visual editor | JSON

Import managed policy

Expand all | Collapse all

STS (1 action) 1 warning — Clone | Remove

Service: STS

Actions: Specify the actions allowed in STS — Switch to deny permissions

close

assume

- [x] AssumeRole
- [ ] AssumeRoleWithSAML
- [ ] AssumeRoleWithWebIdentity

Resources: Specify **role** resource ARN for the **AssumeRole** action.

Request conditions: Specify request conditions (optional)

Resource Gr... | enew.com @ ttn... | Global | Supp...

1 warning — Clone | Remove

Servic...

Actior...

Resource...

clos...

Any

Request conditions: Specify request conditions (optional)

Add ARN(s)

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. Learn more

Specify ARN for role — List ARNs manually

arn:aws:iam::187632318301:role/arn:aws:iam::187632318301:role/Jay-S3FullAccess

Account * 187632318301 — Any

Role name with path * arn:aws:iam::187632318301 — Any

Cancel | Add

Review policy

Name* Jay-Assume-Role

Use alphanumeric and '+=,.@-_' characters. Maximum 128 characters.

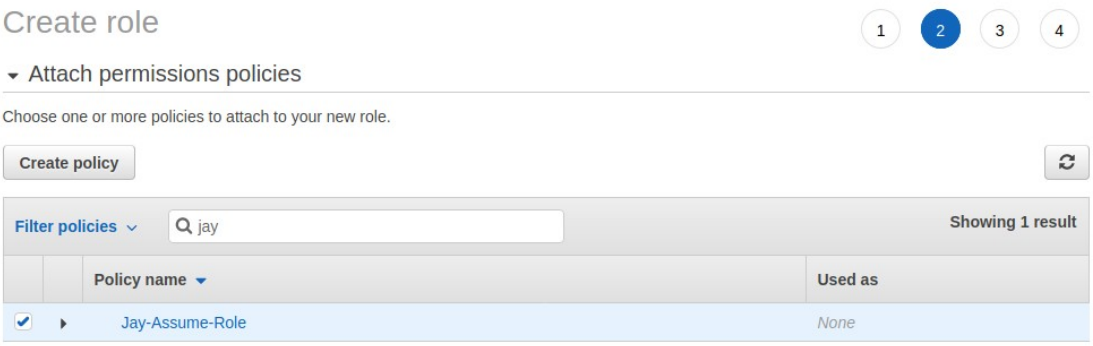
Description

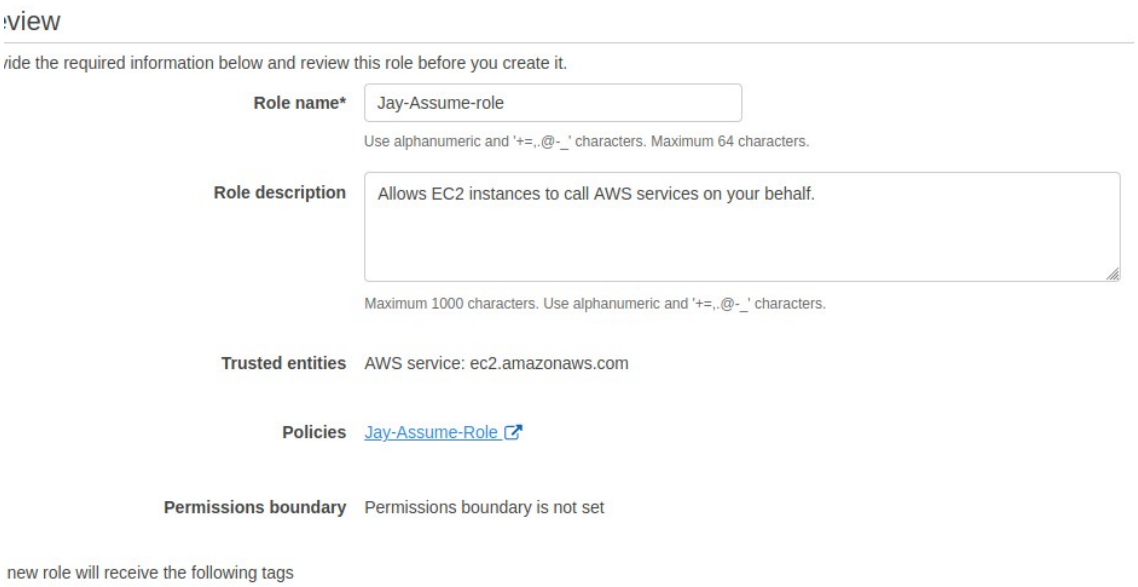
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

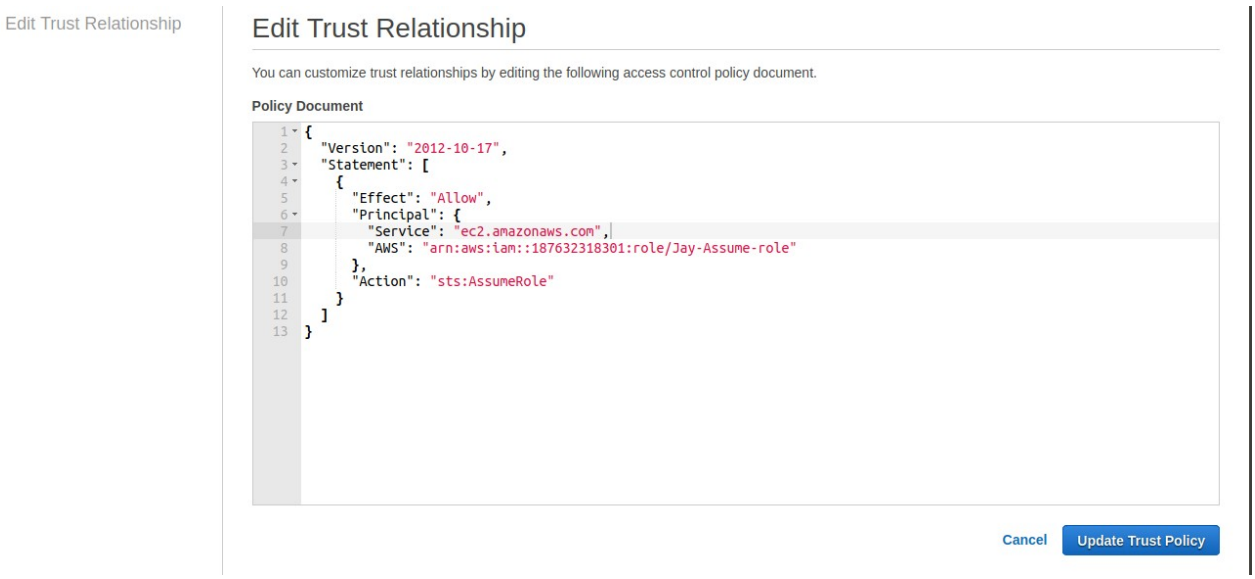
Summary

Filter

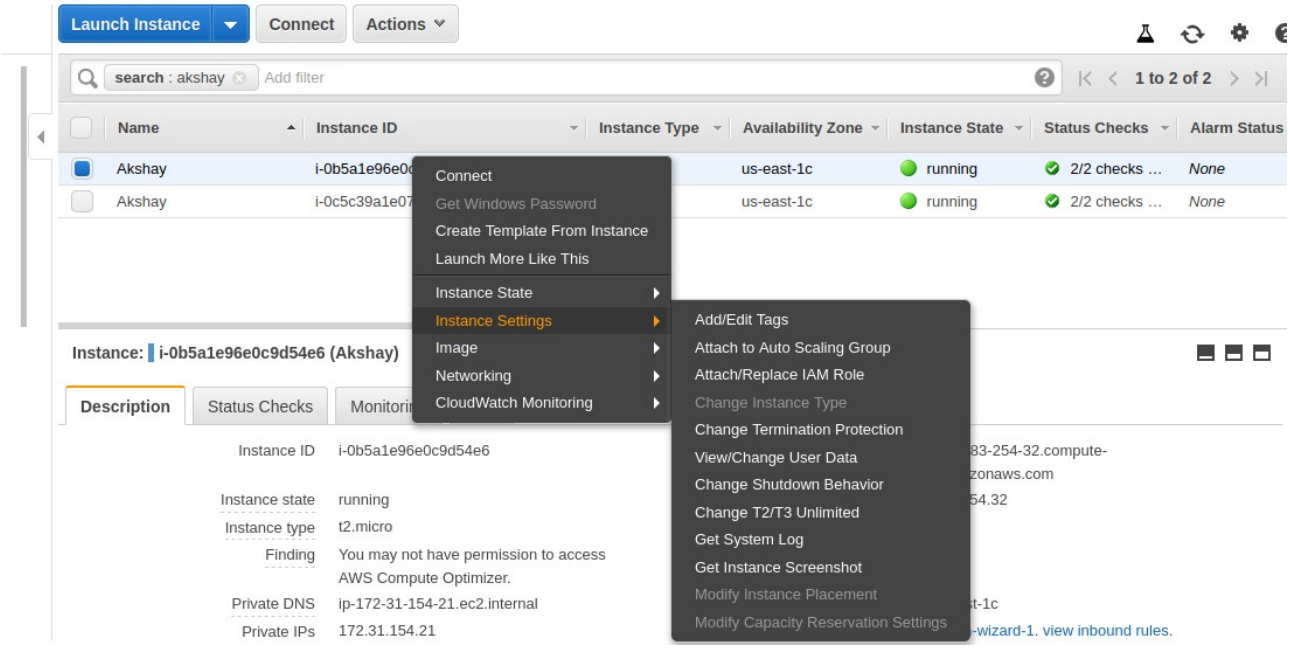
Service	Access level	Resource	Request condition
Allow (1 of 223 services) Show remaining 222			
STS	**Limited**: Write	RoleName | string like | Jay-S3FullAccess, Path | string like | arn:aws:iam::187632318301:role	None







Changed the trust relationship of s3fullaccess role to role can be assumed




```
ubuntu@ip-172-31-154-21:~$ aws sts assume-role --role-arn "arn:aws:iam::187632318301:role/Jay-S3FullAccess" --role-session-name "Jay"
{
    "Credentials": {
        "AccessKeyId": "ASIASXL6B65OY6MFMPPP",
        "SecretAccessKey": "CGvhUaybzYDTaKG12HtaVQAFWUp0wIYUU1QE1EIL",
        "SessionToken": "FwoGZXIvYXdzEGQaDFrYJaenhtPpCdzZZCKnAbdqO8ZOKRFJK/k1gCL0Kge3oILudtu83Tvz0iF10Z1FJR3I5OM0m4onfc3fM6ZBPo0vgimxgV8EaQRJ/aZ
2F0Gcj0d1OWfMGSYXO4fUMZYoX2u+FfM1MmdvwaJb02aDhQ88VYM58f8BJ++2rIv3wMp8Zguq1N8QCuqa5qmHSNtxDbsyuc2+bW2o85BBU6Kt6Ti95Fa03Y9Iu7ALtWp7x/N0jV5OvGOJKLa
08/IFMl2oUApEkl3cJJat2pbXVcwYVxwUvnQMxXoGmIRtqJNxq0Mboq4EVKYY++JpudE=",
        "Expiration": "2020-03-02T11:04:38Z"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROASXL6B65O37PKSJLNN:Jay",
        "Arn": "arn:aws:sts::187632318301:assumed-role/Jay-S3FullAccess/Jay"
    }
}
ubuntu@ip-172-31-154-21:~$ export AWS_ACCESS_KEY_ID=ASIASXL6B65OY6MFMPPP
ubuntu@ip-172-31-154-21:~$ export AWS_SECRET_ACCESS_KEY=CGvhUaybzYDTaKG12HtaVQAFWUp0wIYUU1QE1EIL
ubuntu@ip-172-31-154-21:~$ export AWS_SESSION_TOKEN=FwoGZXIvYXdzEGQaDFrYJaenhtPpCdzZZCKnAbdqO8ZOKRFJK/k1gCL0Kge3oILudtu83Tvz0iF10Z1FJR3I5OM0m4on
fc3fM6ZBPo0vgimxgV8EaQRJ/aZ2F0Gcj0d1OWfMGSYXO4fUMZYoX2u+FfM1MmdvwaJb02aDhQ88VYM58f8BJ++2rIv3wMp8Zguq1N8QCuqa5qmHSNtxDbsyuc2+bW2o85BBU6Kt6Ti95Fa0
3Y9Iu7ALtWp7x/N0jV5OvGOJKLa08/IFMl2oUApEkl3cJJat2pbXVcwYVxwUvnQMxXoGmIRtqJNxq0Mboq4EVKYY++JpudE=
ubuntu@ip-172-31-154-21:~$
```

```
ubuntu@ip-172-31-154-21:~$ aws s3 ls
2019-06-26 12:11:08 0testuser11
2018-04-20 16:59:22 187632318301-awsmacietrail-dataevent
2019-04-02 10:11:33 7testdemo
2019-03-11 04:51:59 abhimanyucftemplate
2020-03-01 18:54:15 abhishek-static
2019-03-04 06:55:23 abneesh1
2019-03-11 11:00:41 adityamun007
2020-03-01 15:41:46 aks-piv-buc
2020-02-26 16:26:29 akshaybuck1
2020-03-01 16:43:30 amankhandelwal1
2019-03-07 09:40:48 anmol-bootcamp19
2019-03-08 00:25:58 avcabc
2017-09-07 03:41:42 aws-codestar-us-east-1-187632318301
2017-09-07 04:23:01 aws-codestar-us-east-1-187632318301-codestartest2-app
2017-09-07 04:23:07 aws-codestar-us-east-1-187632318301-codestartest2-pipe
2017-09-07 03:41:48 aws-codestar-us-east-1-187632318301-codestarttest-pipe
2019-06-26 05:39:55 aws-lambda-trigger-ronozor
2020-02-28 03:56:49 ayush-public-bucket
2020-03-01 12:28:33 ayush-s3
2020-02-25 07:02:11 baban-123
2020-03-01 10:55:09 bucket-yash-1
2018-02-14 12:28:43 cf-templates-71mx96ojlvv5-us-east-1
2019-03-27 15:57:27 cfront1
2020-02-26 11:51:54 chirag-bucket-2
```

3. Block s3 access on the basis of

i. IP

Bucket policy editor ARN: arn:aws:s3:::jaypttn-s3

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

```
{
    "Version": "2012-10-17",
    "Id": "PolicyforWebsitecontent",
    "Statement": [
        {
            "Sid": "ToallowdenyIP",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::jaypttn-s3/*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "1.39.252.5/32"
                }
            }
        }
    ]
}
```

ii. Domain

```
{
   "Version": "2012-10-17",
   "Id": "PolicyforPublicWebsiteContent",
   "Statement": [
       {
           "Sid": "Allow get requests originating from jaypttn bucket url",
           "Effect": "Allow",
           "Principal": "*",
           "Action": "s3:GetObject",
           "Resource": "arn:aws:s3:::jaypttn/*",
           "Condition": {
               "StringLike": {
                       "aws:Referer": [
                       "http://jaypttn.s3-website-us-east-1.amazonaws.com/*"
                       ]
                   }
               }
           }

       ]
   }/*
```

iii. Pre-signed URL(Time based)

A presigned URL gives you access to the object identified in the URL, provided that the creator of the presigned URL has permissions to access that object. That is, if you receive a presigned URL to upload an object, you can upload the object only if the creator of the presigned URL has the necessary permissions to upload that object.

All objects and buckets by default are private. The presigned URLs are useful if you want your user/customer to be able to upload a specific object to your bucket, but you don't require them to have AWS security credentials or permissions. When you create a presigned URL, you must

provide your security credentials and then specify a bucket name, an object key, an HTTP method (PUT for uploading objects), and an expiration date and time. The presigned URLs are valid only for the specified duration.

```
{
    "Version": "2012-10-17",
    "Id": "PolicyforPublicWebsiteContent",
    "Statement": [
        {
            "Sid": "Pre-signed URL",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:Get*",
            "Resource": "arn:aws:s3:::jaypttn/*",
            "Condition": {
                "StringEquals": {
                    "s3:authtype": "REST-QUERY-STRING"
                }
            }
        }
    ]
}
```

4. ACL, Bucket policy, IAM Policy.

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.

5. Mount S3 to an EC2 instance.

```
jay@Jay-Patel:~ $
jay@Jay-Patel:~ $  sudo apt-get install automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-conf
ig
Reading package lists... Done
Building dependency tree
Reading state information... Done
automake is already the newest version (1:1.15.1-3ubuntu2).
automake set to manually installed.
autotools-dev is already the newest version (20180224.1).
autotools-dev set to manually installed.
fuse is already the newest version (2.9.7-1ubuntu1).
make is already the newest version (4.1-9.1ubuntu1).
make set to manually installed.
pkg-config is already the newest version (0.29.1-0ubuntu2).
pkg-config set to manually installed.
g++ is already the newest version (4:7.4.0-1ubuntu2.3).
g++ set to manually installed.
git is already the newest version (1:2.17.1-1ubuntu0.6).
libssl-dev is already the newest version (1.1.1-1ubuntu2.1~18.04.5).
libssl-dev set to manually installed.
The following additional packages will be installed:
  gir1.2-harfbuzz-0.0 icu-devtools libgraphite2-dev libharfbuzz-dev libharfbuzz-gobject0 libicu-dev libicu-le-hb-dev libicu-le-hb0 libiculx60
  libselinux1-dev libsepol1-dev
Suggested packages:
  libcurl4-doc libgnutls28-dev libidn11-dev libkrb5-dev librtmp-dev libssh2-1-dev libgraphite2-utils icu-doc
The following NEW packages will be installed:
  gir1.2-harfbuzz-0.0 icu-devtools libcurl4-gnutls-dev libfuse-dev libgraphite2-dev libharfbuzz-dev libharfbuzz-gobject0 libicu-dev
  libicu-le-hb-dev libicu-le-hb0 libiculx60 libselinux1-dev libsepol1-dev libxml2-dev
0 upgraded, 14 newly installed, 0 to remove and 1 not upgraded.
Need to get 11.1 MB of archives.
After this operation, 52.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 gir1.2-harfbuzz-0.0 amd64 1.7.2-1ubuntu1 [18.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 icu-devtools amd64 60.2-3ubuntu3.1 [179 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcurl4-gnutls-dev amd64 7.58.0-2ubuntu3.8 [294 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libsepol1-dev amd64 2.7-1 [324 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libselinux1-dev amd64 2.7-2build2 [149 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libfuse-dev amd64 2.9.7-1ubuntu1 [105 kB]
```

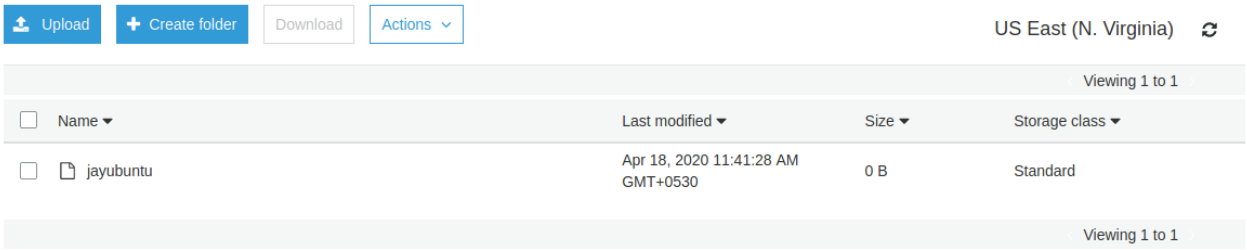
```
jay@Jay-Patel:~ $ git clone https://github.com/s3fs-fuse/s3fs-fuse.git
Cloning into 's3fs-fuse'...
remote: Enumerating objects: 53, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 5936 (delta 24), reused 27 (delta 12), pack-reused 5883
Receiving objects: 100% (5936/5936), 3.59 MiB | 472.00 KiB/s, done.
Resolving deltas: 100% (4106/4106), done.
jay@Jay-Patel:~ $
```

```
jay@Jay-Patel:~ $ cd s3fs-fuse/
jay@Jay-Patel:s3fs-fuse (master)$ ./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 ----------
jay@Jay-Patel:s3fs-fuse (master)$ ./configure --prefix=/usr --openssl
configure: error: unrecognized option: `--openssl'
Try `./configure --help' for more information
jay@Jay-Patel:s3fs-fuse (master)$ ./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++
```

```
jay@Jay-Patel:s3fs-fuse (master)$ which s3fs
/usr/bin/s3fs
jay@Jay-Patel:s3fs-fuse (master)$ sudo touch /etc/passwd
passwd   passwd-
jay@Jay-Patel:s3fs-fuse (master)$ sudo touch /etc/passwd
passwd   passwd-
jay@Jay-Patel:s3fs-fuse (master)$ sudo touch /etc/passwd-s3fs
jay@Jay-Patel:s3fs-fuse (master)$ sudo vim /etc/passwd-s3fs
jay@Jay-Patel:s3fs-fuse (master)$ sudo chmod 640 /etc/passwd-s3fs
jay@Jay-Patel:s3fs-fuse (master)$ cd /
```

```
jay@Jay-Patel:mnt $ sudo s3fs jaypttn -o use_cache=/tmp -o allow_other
-o uid=1001 -o mp_umask=1002 -o multireq_max=5 jaypttnlocal
jay@Jay-Patel:mnt $ ll
total 9
drwxr-xr-x  3 root root 4096 Apr 18 11:33 ./
drwxr-xr-x 24 root root 4096 Apr 13 00:42 ../
drwxrwxr-x  1 1001 root    0 Jan  1  1970 jaypttnlocal/
jay@Jay-Patel:mnt $ df -Th | grep jaypttnlocal
s3fs           fuse.s3fs  256T     0  256T   0% /mnt/jaypttnlocal
jay@Jay-Patel:mnt $
```

```
jay@Jay-Patel:jaypttnlocal $ ll
total 5
drwxrwxr-x 1 1001 root    0 Jan  1  1970 ./
drwxr-xr-x 3 root root 4096 Apr 18 11:33 ../
-rw-r--r-- 1 1001 root    0 Apr 18 11:41 jayubuntu
```



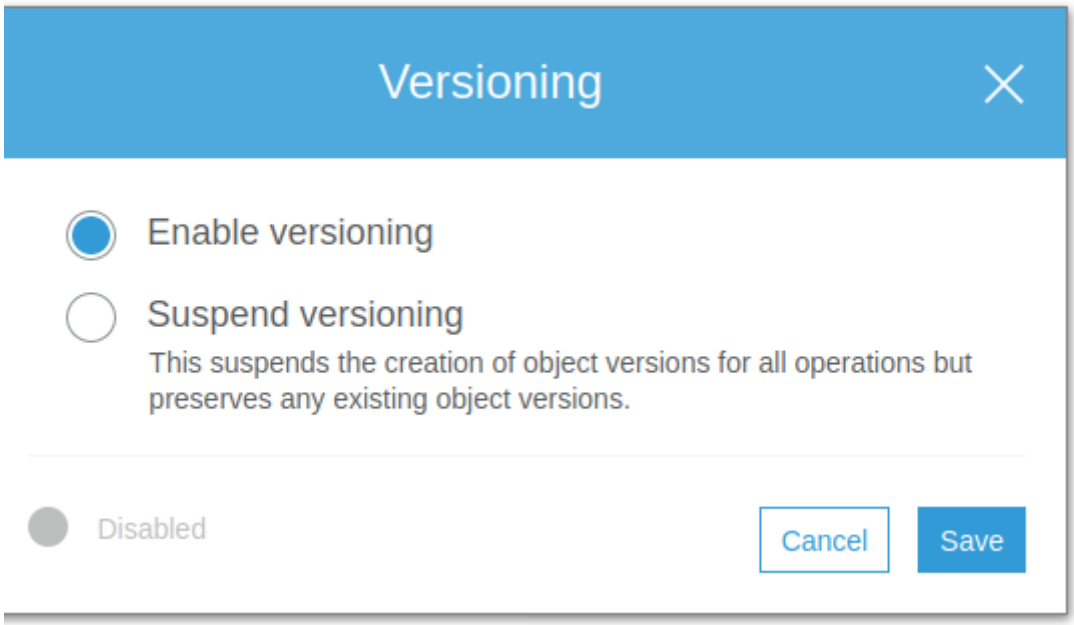
6. Change content type using s3.

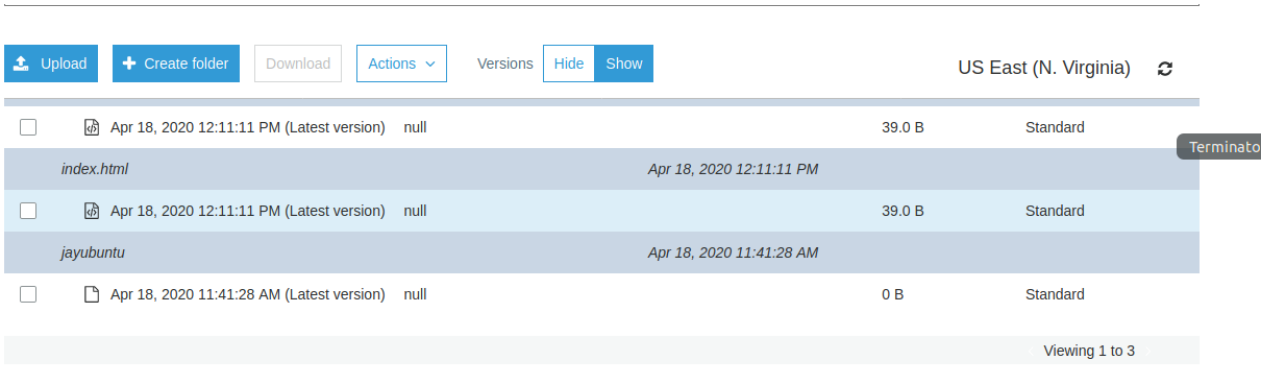
```
jay@Jay-Patel:~ $ aws s3api get-object --bucket jaypttn --key index.html error.html
{
    "AcceptRanges": "bytes",
    "LastModified": "2020-04-18T06:31:34+00:00",
    "ContentLength": 39,
    "ETag": "\"d63ae29a48c342cc6062abe8f5ac5dfa\"",
    "ContentType": "text/html",
    "Metadata": {}
}
jay@Jay-Patel:~ $
```

```
jay@Jay-Patel:~ $ aws s3 cp s3://jaypttn/ s3://jaypttn/  --exclude '*' --include '*.html' --no-guess-mime-type --content-type="text/plain" --met
adata-directive="REPLACE" --recursive
copy: s3://jaypttn/index.html to s3://jaypttn/index.html
copy: s3://jaypttn/error.html to s3://jaypttn/error.html
jay@Jay-Patel:~ $
```

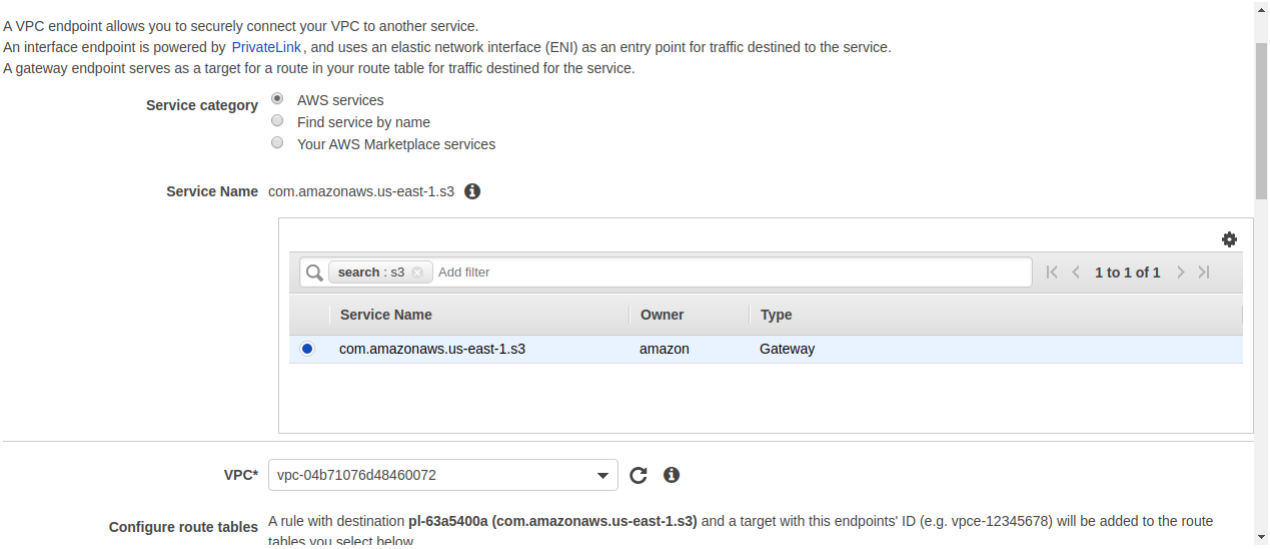
```
jay@Jay-Patel:~ $ aws s3api get-object --bucket jaypttn --key index.html jay.xtx
{
    "AcceptRanges": "bytes",
    "LastModified": "2020-04-18T06:41:11+00:00",
    "ContentLength": 39,
    "ETag": "\"d63ae29a48c342cc6062abe8f5ac5dfa\"",
    "ContentType": "text/plain",
    "Metadata": {}
}
jay@Jay-Patel:~ $
```

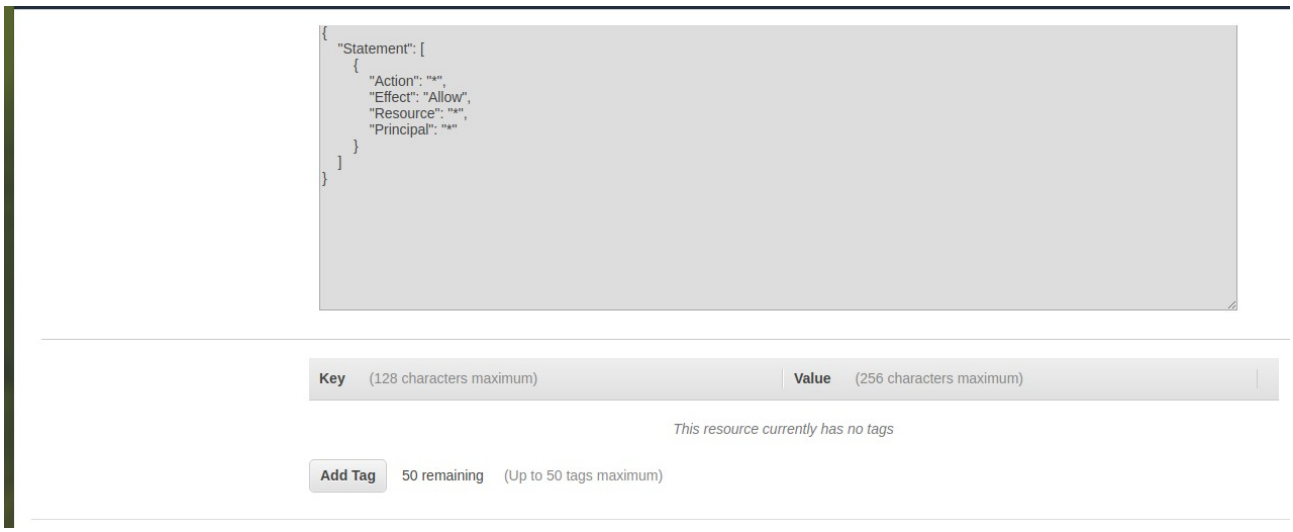
7. Retrive previous version of S3(enable versioning).





8. S3 VPC endpoint.





9. 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.

Configure CORS policy in s3