

```
aws_secret_access_key=XXXXXXXXXXXXXXXXXXXXHUGGXXXX
```

```
[user2]
```

```
aws_access_key_id=AKIAIXXXXXXXXXXDHBXXXX
```

```
aws_secret_access_key=je7MtXXXXXXXXBF/XXXXXXX/XXXXXXXKEY
```

```
~/.aws/config
```

```
[default]
```

```
region=us-west-2
```

```
output=json
```

```
[profile user2]
```

```
region=us-east-1
```

```
output=text
```

Note: The AWS credentials file uses a different naming format than the CLI config file for named profiles. Do not include the 'profile ' prefix when configuring a named profile in the AWS credentials file.

The following settings are supported.

**aws\_access\_key\_id** – AWS access key.

**aws\_secret\_access\_key** – AWS secret key.

**aws\_session\_token** – AWS session token. A session token is only required if you are using temporary security credentials.

**region** – AWS region.

**output** – output format (json, text, or table)

## AWS CLI NAMED PROFILE

You can configure additional, named profiles by using the --profile option.

The AWS CLI supports named profiles stored in the config and credentials files. You can configure additional profiles by using aws configure with the --profile option or by adding entries to the config and credentials files.

```
satish@devops:~/aws$ aws configure --profile testuser
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

## Precedence for AWS CLI

The above configuration values have the following precedence:

- Command line options
- Environment variables
- Configuration file

## AWS CLI SUB COMANDS

1. LIST-----> List common configuration sources
2. GET-----> get the value of single configure value
3. SET -----> set the value of single configure var to see the list of configuration data  
:~/aws\$ aws configure list -----> It will give output default aws profile  
:~/aws\$ aws configure list --profile testuser

```
satish@devops:~/aws$ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "AWS-CLI",
            "Arn": "arn:aws:iam::256189000252:user/AWS-CLI",
            "CreateDate": "2017-06-06T10:25:21Z",
            "UserId": "AIDAJOJDFDEFX26CC3HMK"
        },
        {
            "Path": "/",
            "UserName": "cli_user",
            "Arn": "arn:aws:iam::256189000252:user/cli_user",
            "CreateDate": "2017-06-06T05:49:13Z",
            "UserId": "AIDAIZ7UTLVMTED4JUDRK"
        },
        {
            "Path": "/",
            "UserName": "devadmin",
            "Arn": "arn:aws:iam::256189000252:user/devadmin",
            "CreateDate": "2017-04-24T18:27:06Z",
            "UserId": "AIDAIX3EYXXE6JHOGZTIA"
        },
        {
            "Path": "/",
            "UserName": "testuser",
            "Arn": "arn:aws:iam::256189000252:user/testuser",
            "CreateDate": "2017-06-09T05:12:37Z",
            "UserId": "AIDAIM2I7PFLZCQDXOH6A"
        }
    ]
}
```

arn-->The Amazon Resource Name (ARN) specifying the group.

arn:partition:service:region:account:resource

## Manage Users & Security Groups from AWS CLI

### ❖ User Creation

Syntax

```
Devops: ~/.aws$ aws iam create-user --user-name testuser2
```

```
satish@devops:~/aws$ aws iam create-user --user-name testuser2
{
  "User": {
    "CreateDate": "2017-06-09T10:19:22.508Z",
    "UserId": "AIDAISIF707H4WARZ5RCY",
    "UserName": "testuser2",
    "Path": "/",
    "Arn": "arn:aws:iam::256189000252:user/testuser2"
  }
}
```

### ❖ Creating User & Password with Profile

```
Devops: ~/.aws$ aws iam create-login-profile --user-name testuser -password
*****
```

```
satish@devops:~/aws$ aws iam create-login-profile --user-name testuser --password
rd devops@2468#
{
  "LoginProfile": {
    "PasswordResetRequired": false,
    "UserName": "testuser",
    "CreateDate": "2017-06-09T10:17:12.926Z"
  }
}
```

### ❖ Creating Access key & SecretAccess key for User

```
satish@devops:~/aws$ aws iam create-access-key --user-name testuser
{
  "AccessKey": {
    "SecretAccessKey": "hKgqu4BNCsaDjFMaepnDQd77Z4X4DZeanThLkga2",
    "AccessKeyId": "AKIAIM3MCXZV72K7DYBQ",
    "Status": "Active",
    "CreateDate": "2017-06-09T10:21:39.644Z",
    "UserName": "testuser"
  }
}
```

### ❖ Security Group Creation

```
satish@devops:~/.aws$ aws ec2 create-security-group --group-name testgroup --description "testgroups"
{
  "GroupId": "sg-160c376d"
}
```

### ❖ Inbound Rules for Security Group

```
:~/aws$ aws ec2 authorize-security-group-ingress --group-name testgroup --
protocol tcp --port 8080 --port 22 --cidr 0.0.0.0/0
```

```
:~/aws$ aws ec2 authorize-security-group-ingress --group-name testgroup --protocol tcp --port 8080 --port 22 --cidr 0.0.0.0/0
```

### ❖ Creating Key Pair

```
devops:~/aws$ aws ec2 create-key-pair --key-name test-key --query 'KeyMaterial' --output text > test.pem
```

```
~/aws$ aws ec2 create-key-pair --key-name test-key --query 'KeyMaterial' --
output text > test.pem
```

### ❖ we can also create key pair by giving for only Region also

```
~/aws$ aws -region us-west-2 ec2 create-key-pair --key-name test-key3 --query
'KeyMaterial' --output text > test.pem
```

## Setup & Manage EC2

### ❖ Create Instances

```
:~/aws$ aws ec2 run-instances --image-id ami-4836a428 --count 1 --instance-type
t2.micro --key-name test-key --security-groups testgroup
```

```
satish@devops:~/aws$ aws ec2 run-instances --image-id ami-4836a428 --count 1 --instance-type t2.micro --key-name test-key --security-groups testgroup
{
    "ReservationId": "r-0251550944839989f",
    "Groups": [],
    "Instances": [
        {
            "ClientToken": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "Placement": {
                "GroupName": "",
                "AvailabilityZone": "us-west-2b",
                "Tenancy": "default"
            },
            "ProductCodes": [],
            "ImageId": "ami-4836a428",
            "StateTransitionReason": "",
            "Monitoring": {
                "State": "disabled"
            },
            "EbsOptimized": false,
            "LaunchTime": "2017-06-09T12:06:31.000Z",
            "RootDeviceName": "/dev/xvda",
            "VirtualizationType": "hvm",
            "SourceDestCheck": true,
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "PrivateIpAddress": "172.31.26.236",
            "SubnetId": "subnet-36135251",
        }
    ]
}
```

## ❖ Starting instances

```
@devops:~$ aws ec2 start-instances --instance-ids i-06f132d68ec51c93f
```

```
satish@devops:~/aws$ aws ec2 start-instances --instance-ids i-06f132d68ec51c93f
{
    "StartingInstances": [
        {
            "InstanceId": "i-06f132d68ec51c93f",
            "CurrentState": {
                "Name": "pending",
                "Code": 0
            },
            "PreviousState": {
                "Name": "pending",
                "Code": 0
            }
        }
    ]
}
```

## ❖ Stop instances

```
devops:~$ aws ec2 stop-instances --instance-ids i-06f132d68ec51c93f
```

```

satish@devops:~$ aws ec2 stop-instances --instance-ids i-06f132d68ec51c93f
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Name": "stopping",
                "Code": 64
            },
            "InstanceId": "i-06f132d68ec51c93f",
            "PreviousState": {
                "Name": "stopping",
                "Code": 64
            }
        }
    ]
}

```

Code -> (integer)

The low byte represents the state. The high byte is an opaque internal value and should be ignored.

- 0 : pending
- 16 : running
- 32 : shutting-down
- 48 : terminated
- 64 : stopping
- 80 : stopped

### ❖ Terminate Instances

```
$ aws ec2 terminate-instances --instance-ids i-564845gt655
```

## Snapshot & Volume

### ❖ Snapshot Creation

```
@devops:~/.aws$ aws ec2 create-snapshot --volume-id vol-04bc9d39b50e4549a --description " testuser snapshot"
```

```

satish@devops:~$ aws ec2 create-snapshot --volume-id vol-04bc9d39b50e4549a --description " testuser snapshot"
{
    "Description": " testuser snapshot",
    "Encrypted": false,
    "StartTime": "2017-06-12T09:28:47.000Z",
    "OwnerId": "256189000252",
    "VolumeId": "vol-04bc9d39b50e4549a",
    "SnapshotId": "snap-08f359bbaa3ea8703",
    "State": "pending",
    "Progress": "",
    "VolumeSize": 8
}

```

## Manage Volume Creation

### Visualpath Training & Consulting

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

## ❖ Describe Volumes

```
@devops:~/aws$ aws ec2 describe-volumes --output table
```

```
@devops:~/aws$ aws ec2 describe-volumes --output table
```

```
@devops:~/aws$ aws ec2 describe-volumes --filters Name=volume-id,Values=vol-04bc9d39b50e4549a --output table
```

## ❖ Creating Volume

```
@devops:~/aws$ aws ec2 create-volume --size 8 --region us-west-2 --availability-zone us-west-2b --volume-type gp2
```

```
satish@devops:~/aws$ aws ec2 create-volume --size 8 --region us-west-2 --availability-zone us-west-2b --volume-type gp2
{
    "CreateTime": "2017-06-12T10:39:28.314Z",
    "Size": 8,
    "VolumeType": "gp2",
    "VolumeId": "vol-00b0533a0c2d009ab",
    "SnapshotId": "",
    "Iops": 100,
    "State": "creating",
    "Encrypted": false,
    "AvailabilityZone": "us-west-2b"
}
```

--volume-type (string)

Possible values:

- standard
- io1 -- Provisioned IOPS SSD
- gp2 -- General Purpose SSD
- sc1 --Cold HDD
- st1 --Throughput Optimized HDD

Default value: standard (Magnetic Volumes)

## ❖ Attach Volume

```
devops:~/aws$ aws ec2 attach-volume --volume-id vol-00b0533a0c2d009ab --instance-id i-06f132d68ec51c93f --device /dev/sdf
```

```
satish@devops:~/aws$ aws ec2 attach-volume --volume-id vol-00b0533a0c2d009ab --instance-id i-06f132d68ec51c93f --device /dev/sdf
{
  "VolumeId": "vol-00b0533a0c2d009ab",
  "Device": "/dev/sdf",
  "InstanceId": "i-06f132d68ec51c93f",
  "State": "attaching",
  "AttachTime": "2017-06-12T10:52:18.791Z"
}
```

### ❖ De-Attach the volume

```
devops:~/aws$ aws ec2 detach-volume --volume-id vol-1234567890abcdef0 --force
```

*Note: Before detach the volume we should be unmount the volume within your Operating System.*

## Elastic IP

### ❖ Elastic IP Address Allocation

```
@devops:~/aws$ aws ec2 allocate-address
```

```
satish@devops:~/aws$ aws ec2 allocate-address
{
  "PublicIp": "52.32.86.20",
  "AllocationId": "eipalloc-2cb89e16",
  "Domain": "vpc"
}
```

### ❖ Attaching Elastic IP to instances

```
@devops:~/aws$ aws ec2 associate-address --instance-id i-06f132d68ec51c93f --
allocation-id eipalloc-2cb89e16
```

```
satish@devops:~/aws$ aws ec2 associate-address --instance-id i-06f132d68ec51c93f --allocation-id eipalloc-2cb89e16
{
  "AssociationId": "eipassoc-1f9d2622"
}
```

### ❖ Get list of elastic IP's

```
$ aws ec2 describe-addresses
```

### ❖ Disassociate Elastic IP

```
@devops:~/aws$ aws ec2 disassociate-address --association-id eipassoc-cf962df2
```

```
@devops:~/aws$ aws ec2 disassociate-address --association-id eipassoc-cf962df2
```

### ❖ Delete Elastic IP

```
devops:~/aws$ aws ec2 release-address --allocation-id eipalloc-2cb89e16
```

## 20. S3cmd

**DOCUMENTED BY VENU RUPANI.**

**Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

s3md is a command line utility used for creating s3 buckets, uploading, retrieving and managing data to Amazon s3 storage.

### Install s3cmd Package

s3cmd is available in default rpm repositories for CentOS, RHEL and Ubuntu systems, you can install it using simply executing following commands on your system.

#### On Ubuntu/Debian:

```
$ sudo apt-get install s3cmd
```

#### Install Latest s3cmd from Source: -

If you are not getting latest version of s3cmd using above package managers, You can install last s3cmd version on your system using source code.

Visit this url (<http://ufpr.dl.sourceforge.net/project/s3tools/s3cmd/>)

or

Use below command to download latest version of s3cmd.

```
$ wget http://ufpr.dl.sourceforge.net/project/s3tools/s3cmd/1.6.1/s3cmd-1.6.1.tar.gz
```

```
$ tar xzf s3cmd-1.6.1.tar.gz
```

Now install it using below command with source files.

```
$ cd s3cmd-1.6.1
```

```
$ sudo python setup.py install
```

#### Configure s3cmd Environment: -

In order to configure s3cmd we would require Access key and Secret key of your S3 Amazon account.

After getting key files, use below command to configure s3cmd.

NOTE: - Before executing below command create a user with full permissions of s3 access in your aws account.

```
$ s3cmd --configure
```

After press, the above command please enter the key file information like given below.

Enter new values or accept defaults in brackets with Enter.

Refer to user manual for detailed description of all options.

#### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

*Access key and Secret key are your identifiers for Amazon S3*

*Access Key: xxxxxxxxxxxxxxxxxxxxxxxxx*

*Secret Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

**Encryption password is used to protect your files from reading by unauthorized persons while in transfer to S3**

*Encryption password: xxxxxxxxxxxx*

*Path to GPG program [/usr/bin/gpg] :*

When using secure HTTPS protocol all communication with Amazon S3 servers is protected from 3rd party eavesdropping. This method is slower than plain HTTP and can't be used if you're behind a proxy

*Use HTTPS protocol [No] : Yes*

**New settings:**

*Access Key: xxxxxxxxxxxxxxxxxxxxxxxxx*

*Secret Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

*Encryption password: xxxxxxxxxxxx*

*Path to GPG program: /usr/bin/gpg*

*Use HTTPS protocol: True*

*HTTP Proxy server name:*

*HTTP Proxy server port: 0*

*Test access with supplied credentials? [Y/n] Y*

*Please wait, attempting to list all buckets...*

*Success. Your access key and secret key worked fine :-)*

*Now verifying that encryption works...*

*Success. Encryption and decryption worked fine :-)*

*Save settings? [y/N] y*

*Configuration saved to '/root/.s3cfg'*

Now setup is ready to create s3 buckets, uploading, retrieving and managing data to Amazon s3 storage.

**Uses of s3cmd Command Line: -**

**Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

Once configuration is successfully completed. Now find below command details to how to manage s3 buckets using commands.

## 1. List All S3 Bucket

Use following command to list all s3 buckets in your aws account.

```
$*s3cmd ls
```

## 2. Creating New Bucket

To create a new bucket in Amazon s3 use below command. It will create bucket named **venudevops** (create your bucket name as your wish) in aws account.

```
$ s3cmd mb s3://venudevops
```

```
Bucket 's3://venudevops/' created
```

## 3. Uploading file in Bucket

Below command will upload file **devops.txt** to s3 bucket using s3cmd command.

```
$ s3cmd put devops.txt s3://venudevops/
devops.txt -> s3://venudevops/devops.txt [1 of 1]
190216 of 190216 100% in 0s 1668.35 kB/s done
```

## 4. Uploading Directory in Bucket

If we need to upload entire directory use **-r** to upload it recursively like below.

```
$ s3cmd put -r backup s3://venudevops/
backup/devops1.txt -> s3://venudevops/backup/devops1.txt [1 of 2]
9984 of 9984 100% in 0s 18.78 kB/s done
backup/devops2.txt -> s3://venudevops/backup/devops2.txt [2 of 2]
0 of 0 0% in 0s 0.00 B/s done
```

Make sure you are not adding trailing slash in upload directory named backup (eg: backup/), else it will upload only content of backup directory only.

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
$ s3cmd put -r backup/ s3://venudevops/  
  
backup/devops1.txt -> s3://venudevops/devops1.txt [1 of 2]  
 9984 of 9984 100% in 0s 21.78 kB/s done  
backup/devops2.txt -> s3://venudevops/devops2.txt [2 of 2]  
 0 of 0 0% in 0s 0.00 B/s done
```

## 5. List Data of S3 Bucket

List the objects of s3 bucket using **ls** switch with s3cmd.

```
$ s3cmd ls s3://venudevops/  
  
DIR s3://venudevops/backup/  
2017-06-03 10:58 190216 s3://venudevops/file.txt
```

## 6. Download Files from Bucket

Sometimes if we need to download files from s3 bucket, Use following commands to download it.

```
$ s3cmd get s3://venudevops/devops.txt  
  
s3://venudevops/devops.txt -> ./devops.txt [1 of 1]  
4 of 4 100% in 0s 10.84 B/s done
```

## 7. Remove Data of S3 Bucket

To remove files or folder from s3 bucket use following commands.

Removing file from s3 bucket

```
$ s3cmd del s3://venudevops/devops.txt  
  
File s3://venudevops/devops.txt deleted  
  
Removing directory from s3 bucket
```

```
$ s3cmd del s3://venudevops/backup
```

```
| File s3://venudevops/backup deleted |
```

## 8. Remove S3 Bucket

If we don't need s3 bucket any more, we can simply delete it using following command. Before removing bucket make sure its empty.

```
$ s3cmd rb s3://venudevops
'
ERROR: S3 error: 409 (BucketNotEmpty): The bucket you tried to delete is not empty
```

Read more about s3cmd sync <http://s3tools.org/s3cmd>

## 21. s3cmd-sync

**DOCUMENTED BY VENU RUPANI.**

### 1. Syncing Files from Local => S3 Bucket

For example I want to sync my local directory **/root/mydir/** to S3 bucket directory

**s3://venudevops/mydir/** where **venudevops** is bucket name. I have created some new files in **/root/mydir/** and sync to s3 bucket using following command.

```
$ s3cmd sync /root/mydir/ s3://venudevops/mydir/
```

[Sample Output]

```
/root/mydir/index.php -> s3://venudevops/mydir/index.php [1 of 2]
 397 of 397 100% in      0s      4.02 kB/s  done
/root/mydir/readme.html -> s3://venudevops/mydir/readme.html [2 of 2]
```

**Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
9202 of 9202 100% in 0s 103.62 kB/s done
```

```
Done. Uploaded 9599 bytes in 0.3 seconds, 27.92 kB/s
```

**Note:** Do not forgot to add trailing slash (/) in local directory path when specifying s3 bucket with full directory path.

To keep preserve file attributes like date/time etc use **-p** or **--preserve** parameter like below

```
$ s3cmd sync /root/mydir/ --preserve s3://venudevops/mydir/
```

If we want to sync only newly created file on source use **--skip-existing** parameter. It will skip all files which already exists on destination either its modified on source.

```
# s3cmd sync /root/mydir/ --skip-existing s3://venudevops/mydir/
```

If you want to delete all files from s3 bucket which has removed from local use **--delete-removed** parameter.

```
$ s3cmd sync /root/mydir/ --delete-removed s3://venudevops/mydir/
```

## 2. Syncing Files from S3 Bucket => Local Directory

For this example I am again using same folder and bucket used above. To test this i have put some extra files in s3 bucket (s3://venudevops/mydir/) and executed following command to sync all files to local directory.

```
$ s3cmd sync s3://venudevops/mydir/ /root/mydir/
```

### [Sample Output]

```
s3://venudevops/mydir/logo.jpg -> /root/mydir/logo.jpg [2 of 3]
```

```
7219 of 7219 100% in 0s 125.28 kB/s done
```

```
s3://venudevops/mydir/user.php -> /root/mydir/user.php [3 of 3]
```

```
40380 of 40380 100% in 0s 596.33 kB/s done
```

```
Done. Downloaded 47599 bytes in 0.3 seconds, 184.40 kB/s
```

We can also used **--preserve**, **--skip-existing** and **--delete-removed** parameters during syncing files from S3 bucket to Local directory as followings.

```
$ s3cmd sync s3://venudevops/mydir/ --preserve /root/mydir/
```

```
$ s3cmd sync s3://venudevops/mydir/ --skip-existing /root/mydir/
```

```
$ s3cmd sync s3://venudevops/mydir/ --delete-removed /root/mydir/
```

Read more about s3cmd sync <http://s3tools.org/s3cmd-sync>

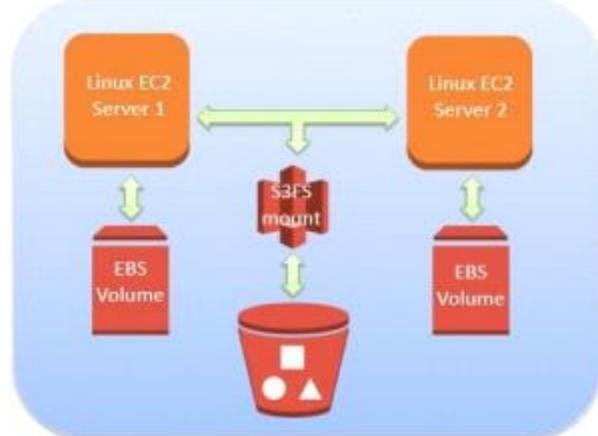
### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

## 22. Mount an Amazon S3 Bucket to a Local Linux File System

**DOCUMENTED BY PRAVEEN.**

**S3FS** is a FUSE (File System in User Space) will mount Amazon S3 as a local file system. S3FS has an ability to manipulate Amazon S3 bucket in many useful ways. If you wish to access your Amazon S3 bucket without mounting it on your server, you can use s3cmd command line utility to manage S3 bucket.



Block diagram of S3FS

### Install prerequisite packages

```
yum install automake fuse fuse-devel gcc-c++ git libcurl-devel libxml2-devel  
make openssl-devel
```

### Download, compile and install the latest version of FUSE module

```
cd /usr/src/  
yum install wget unzip  
wget http://www.techmarksolutions.co.uk/downloads/fuse-2.8.4.tar.gz  
tar xzvf fuse-2.8.4.tar.gz  
cd fuse-2.8.4  
./configure --prefix=/usr/
```

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
make  
make install  
export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig/  
ldconfig
```

#Manually load the FUSE module for the first time

```
modprobe fuse
```

## Verify FUSE installation

```
# pkg-config --modversion fuse
```

## Download, compile and install the latest version of S3FS

```
cd /usr/src/  
wget https://github.com/s3fs-fuse/s3fs-fuse/archive/master.zip  
  
unzip master.zip  
  
cd s3fs-fuse-master  
  
. ./autogen.sh  
  
. ./configure --prefix=/usr --with-openssl  
make  
make install
```

## Verify S3FS installation

```
# s3fs -version
```

## Create an s3fs password file for storing your AWS Access Key ID and Secret Access Key.

[\*Create IAM User and Create S3 Bucket steps are given below\*]

The default location for the s3fs password file can be created:

- using a .passwd-s3fs file in the users home directory (i.e. `~/.passwd-s3fs`)
- using the system-wide `/etc/passwd-s3fs` file
- (AccessKeyId:SecretAccessKey)

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
# cd  
# vi /etc/passwd-s3fs (AccessKeyId:SecretAccessKey)  
# cd /etc
```

### Change the permissions of password file

```
# chmod 400 passwd-s3fs  
# cd /usr/src
```

### Create a directory for mounting the s3fs file system

```
# mkdir s3mnt
```

### Mount bucket to specific directory

```
# s3fs s3fs-aws /usr/src/s3mnt/ -o passwd_file=/etc/passwd-s3fs  
# cd s3mnt/
```

### Create directories and files in s3mnt

```
# mkdir 1 2 3 4 5 6
```

### Check s3 bucket in aws

### Verify the s3fs mounted file system.

```
grep s3fs /etc/mtab  
(*s3fs /usr/src/s3mnt fuse.s3fs rw,nosuid,nodev 0 0*)  
  
df -Th /usr/src/s3mnt/  
(*Filesystem      Type            Size  Used Avail Use% Mounted on  
s3fs            fuse.s3fs       256T     0   256T    0% /usr/src/s3mnt*)
```

### Manually unmount the virtual drive using the umount command

```
# umount /usr/src/s3mnt/
```

### Create IAM User

Create an IAM user that will be used to access S3 from your EC2 instances.

- Login to AWS web console
- Open **IAM | Users**
- Select **Create New User**
- Enter a user name

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

1. s3user
  - Copy generated access keys
  - Set a complicated/long password for the user

## Create S3 Bucket

- Login to AWS web console
- Select **Services S3**
- Create a S3 bucket

1.s3fs-aws

- Add bucket policy to allow IAM User and VPC Endpoint access to all files inside the bucket.

## Commands and details

### 1. Modprobe

**modprobe** is a Linux program originally written by Rusty Russell and used to add a loadable kernel module (LKM) to the Linux kernel or to remove a LKM from the kernel. It is commonly used indirectly: udev relies upon **modprobe** to load drivers for automatically detected hardware.

### 2. Ldconfig

**ldconfig** is a program that is used to maintain the shared library cache. This cache is typically stored in the file /etc/ld.so.cache and is used by the system to map a shared library name to the location of the corresponding shared library file

### 3. Make and Make install

**make** connects the libs to the source and creates the required links and sets it up for the final phase it also parses the human readable to machine readable

**make install** This is the final phase where the compiler will create the binary files and moves all required files / executable and associated libs to their appropriate directories.

### 4. ./autogen.sh

provides automatic build system preparation

auto tools for preparing a build for compilation, verifying functionality, and overcoming common build preparation issues.

### 5. pkg-config --modversion

Requests that the version information of the libraries specified on the command line be displayed. If *pkg-config* can find all the libraries on the command line, each library's version string is printed to stdout, one version per line. In this case *pkg-config* exits successfully. If one or more libraries is unknown, *pkg-config* exits with a nonzero code, and the contents of stdout are undefined.

## **S3FS Description**

[S3fs](#) is a direct mapping of S3 to a filesystem paradigm. Files are mapped to objects. Filesystem metadata (e.g., ownership and file modes) are stored inside the object's meta data. Filenames are keys, with "/" as the delimiter to make listing more efficient, etc.

That's significant because it means there is nothing terribly magical about a bucket being read/written to by s3fs, and in fact you can mount any bucket with s3fs to explore it as a filesystem.

s3fs's main advantage is its simplicity. There are however a few gotchas:

If you're using s3fs to access a bucket it didn't create and have objects in it that have directory-like components in their names (e.g., mypath/myfile), you'll need to create a dummy directory in order to see them (e.g., mkdir mypath).

The project seems to be "regretware". The last open source release was in August 2008. Since then the author seems to have continued all development of new features (e.g., encryption, compression, multi-user access) as a commercial license (subcloud), and with that inherent conflict of interest the future of the GPLed licensed open source version is uncertain.

## **Advantages of S3FS**

If you're using s3fs to access a bucket it didn't create and have objects in it that have directory-like components in their names (e.g., mypath/myfile), you'll need to create a dummy directory in order to see them (e.g., mkdir mypath).

No embedded documentation. Probably another side-effect of the proprietary version, though the available options are documented no the web site.

Inherits S3's limitations: no file can be over 5GB, and you can't partially update a file so changing a single byte will re-upload the entire file.

Inherits S3's performance characteristics: operation on many small files are very efficient (each is a separate S3 object after all)

Though S3 supports partial/chunked downloads, s3fs doesn't take advantage of this so if you want to read just one byte of a 1GB file, you'll have to download the entire GB.

s3fs supports a disk cache, which can be used to mitigate this limitation

## **Features**

- large subset of POSIX including reading/writing files, directories, symlinks, mode, uid/gid, and extended attributes
- compatible with Amazon S3, Google Cloud Storage, and other S3-based object stores

### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

- large files via multi-part upload
- renames via server-side copy
- optional server-side encryption
- data integrity via MD5 hashes
- in-memory metadata caching
- local disk data caching
- user-specified regions, including Amazon GovCloud
- authenticate via v2 or v4 signatures

## **S3FS Limitations**

### **UID/GID**

Now, there is no full UID/GID support. All files will be owned by root. Additionally, if you allow others to access the bucket (using the -o allow\_other option), others can also remove files.

### **Excessive Time-outs**

Currently s3fs can hang the CPU if you have lots of time-outs. This is not a fault of s3fs but rather libcurl. This happens when you try to copy thousands of files in 1 session, it doesn't happen when you upload hundreds of files or less.

### **Moving Large Files**

Moving, renaming, or erasing files may take considerable time since the whole file needs to be accessed first. A workaround could be to use s3fs's cache support with the -o use\_cache=<directory> mount option.

### **File Size**

S3FS has a file size limit of 64GB for the current version (limited by s3fs, not Amazon).

### **Directory Support**

Prior to configuring s3fs on my test system, I used the AWS Management Console to create several directories and to upload some files to the target S3 bucket I intended to mount (idevelopment-software). After mounting the bucket to the local file system, I was unable to see any of the directories. I could see the files I uploaded to the root directory of the bucket, but was unable to see any of the directories. I went back to the AWS Management Console and even to S3Fox and verified that the directories did indeed exist.

It turns out that S3 does not have a native concept of a folder (i.e. directory). It is up to each S3 client tool (AWS Console, S3Fox, s3cmd, s3fs) to implement their own strategy for handling

### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).