Blog

How to Mount Amazon S3 as a File System

Jun 27, 2017 10:17:14 AM

Posted by Gali Kovacs **Topics:** Data Tiering, S3, EBS, EFS, File Services, AWS, Object Storage

Amazon Simple Storage Service (Amazon S3) is generally used as highly durable and scalable data storage for images, videos, logs, big data and other static storage files. In addition to its popularity as a static storage service, some users want to use Amazon S3 storage as a file system mounted to either Amazon EC2, on-premises systems, or even client laptops. Doing so is complex but possible with the help of S3FS or other third-party tools.

In this article, we will show you how to mount an Amazon S3 bucket as file storage and discuss its advantages and drawbacks. We'll also show you how some NetApp cloud solutions can make it possible to have Amazon S3 mount as a file system while cutting down your overall storage costs on AWS.It is important to note that AWS does not recommend the use of Amazon S3 as a block-level file system.

Advantages of Mounting Amazon S3 as a File System

Mounting an Amazon S3 bucket as a file system means that you can use all your existing tools and applications to interact with the Amazon S3 bucket to perform read/write operations on files and folders. Can EC2 mount S3? Using this method enables multiple Amazon EC2 instances to concurrently mount and access data in Amazon S3, just like a shared file system.

Why use an Amazon S3 file system? Any application interacting with the mounted drive doesn't have to

For example, when creating a photo upload application, you can have it store data on a fixed path in a file system and when deploying you can mount an Amazon S3 bucket on that fixed path. This way, the application will write all files in the bucket without you having to worry about Amazon S3 integration at the application level. Another major advantage is to enable legacy applications to scale in the cloud since there are no source code changes required to use an Amazon S3 bucket as storage backend: the application can be configured to use a local path where the Amazon S3 bucket is mounted. This technique is also very helpful when you want to collect logs from various servers in a central location for archiving.

Scripting Options for Mounting a File System to Amazon S3

There are a few options you have for mounting Amazon S3 as a local drive on linux-based systems, which also support setups where you have Amazon S3 mount EC2.

- **S3FS-FUSE:** This is a free, open-source FUSE plugin and an easy-to-use utility which supports major Linux distributions & MacOS. S3FS also takes care of caching files locally to improve performance. This plugin simply shows the Amazon S3 bucket as a drive on your system.
- **ObjectiveFS:** ObjectiveFS is a commercial FUSE plugin which supports Amazon S3 and Google Cloud Storage backends. It claims to offer a full POSIX-compliant file system interface, which means that appends don’t need to rewrite entire files. It also promises performance comparable to a local drive.
- **RioFS:** RioFS is a lightweight utility written using C language. It is comparable to S3FS but has a few limitations: RioFS doesn’t support appending to file, doesn’t support fully POSIX-compliant file system interface, and it can’t rename folders.

How to Mount an Amazon S3 Bucket as a Drive with S3FS

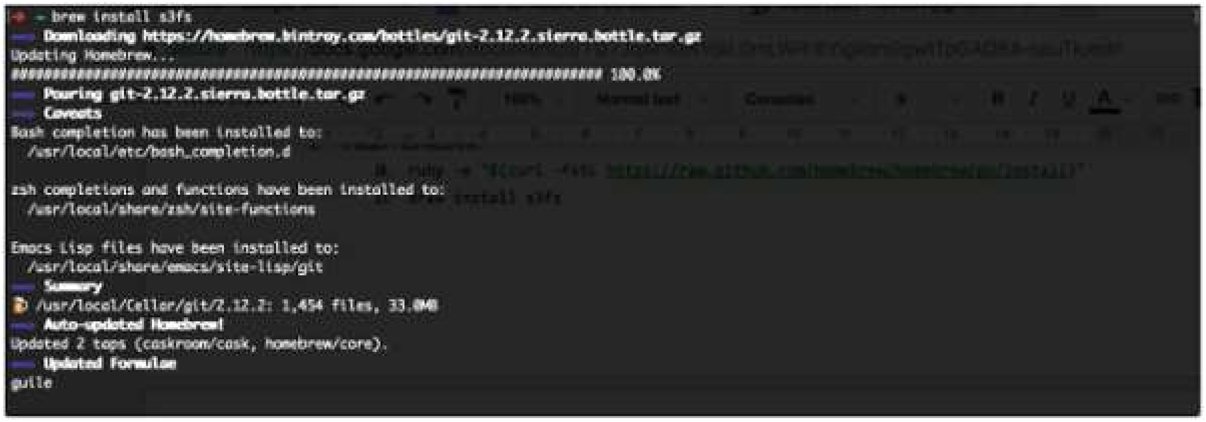
In this section, we’ll show you how to mount an Amazon S3 file system step by step. Mounting an Amazon S3 bucket using S3FS is a simple process: by following the steps below, you should be able to start experimenting with using Amazon S3 as a drive on your computer immediately.

Step 1: Installation

The first step is to get S3FS installed on your machine. please note that S3FS only supports Linux-based systems and MacOS.

1. ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"

2. Brew install s3fs, **as shown below.**



```
➜ ~ brew install s3fs
==> Downloading https://homebrew.bintray.com/bottles/git-2.12.2.sierra.bottle.tar.gz
Updating Homebrew...
######################################################################## 100.0%
==> Pouring git-2.12.2.sierra.bottle.tar.gz
==> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d

zsh completions and functions have been installed to:
  /usr/local/share/zsh/site-functions

Emacs Lisp files have been installed to:
  /usr/local/share/emacs/site-lisp/git
==> Summary
🍺 /usr/local/Cellar/git/2.12.2: 1,454 files, 33.0MB
==> Auto-updated Homebrew!
Updated 2 taps (caskroom/cask, homebrew/core).
==> Updated Formulae
guile
```

>> Installation steps for Ubuntu:

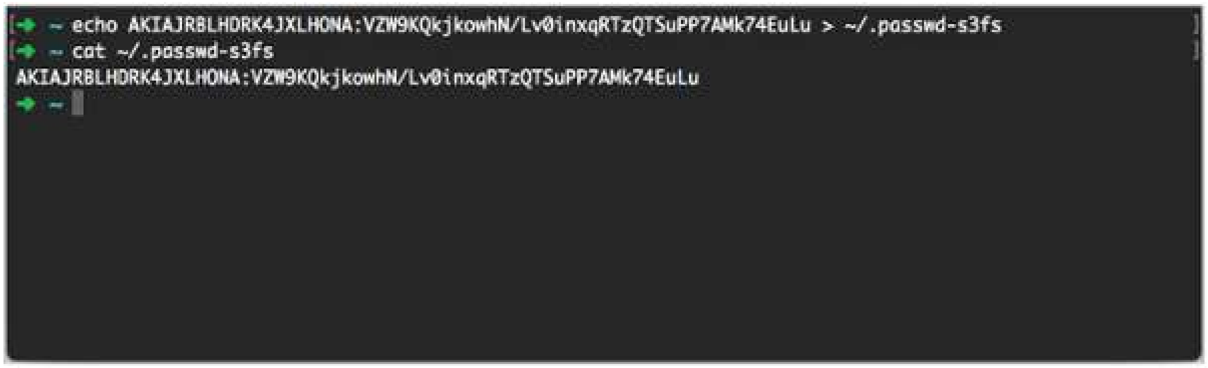
On Ubuntu 16.04, using apt-get, it can be installed by using the command below:

sudo apt-get install s3fs

Step 2: Configuration

1. Once S3FS is installed, set up the credentials as shown below:

echo ACCESS_KEY:SECRET_KEY > ~/.passwd-s3fs

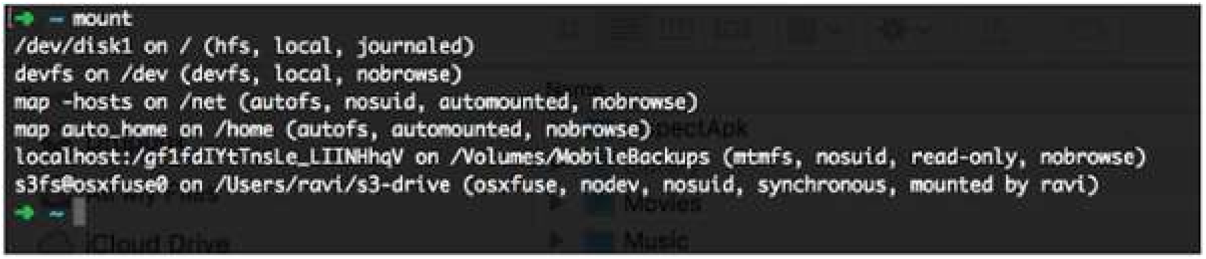


```
➜ ~ echo AKIAJRBLHDRK4JXLHONA:VZW9KQkjkowhN/Lv0inxqRTzQTSuPP7AMk74EuLu > ~/.passwd-s3fs
➜ ~ cat ~/.passwd-s3fs
AKIAJRBLHDRK4JXLHONA:VZW9KQkjkowhN/Lv0inxqRTzQTSuPP7AMk74EuLu
➜ ~
```

You will also need to set the right access permission for the passwd-s3fs file to run S3FS successfully.

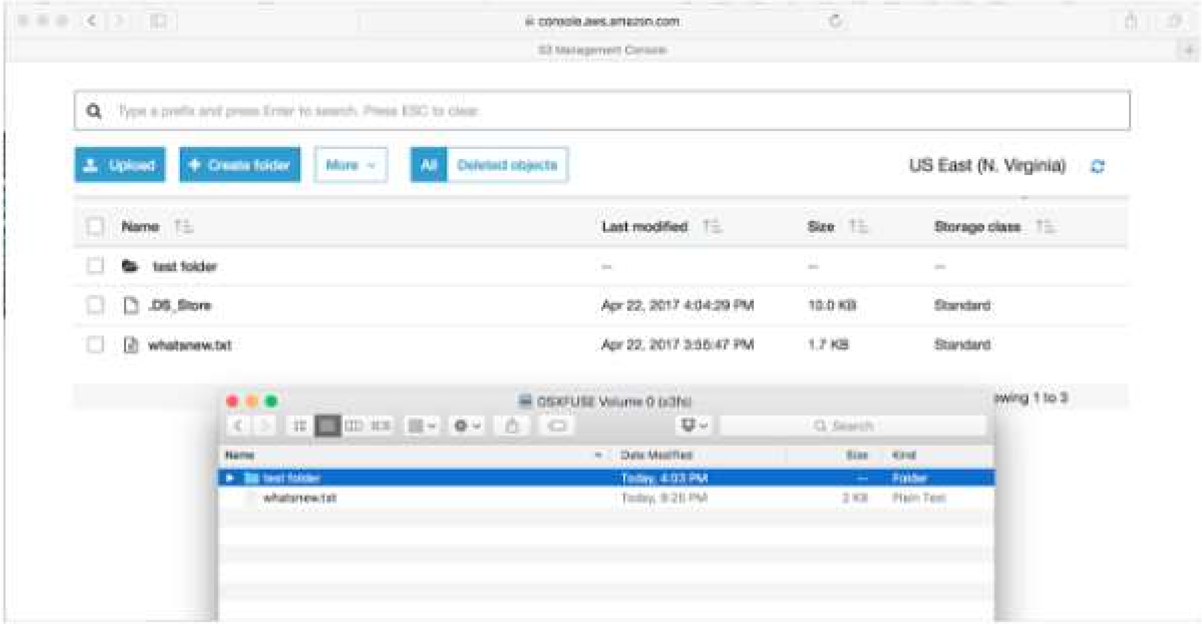
2. Now we're ready to mount the Amazon S3 bucket. Create a folder the Amazon S3 bucket will mount:

- **>>** mkdir ~/s3-drive
- **>>** s3fs <bucketname> ~/s3-drive
- **>>** You might notice a little delay when firing the above command: that's because S3FS tries to reach Amazon S3 internally for authentication purposes. If you don't see any errors, your S3 bucket should be mounted on the ~/s3-drive folder. To verify if the bucket successfully mounted, you can type "mount" on terminal, then check the last entry, as shown in the screenshot below:



```
➜  ~ mount
/dev/disk1 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
localhost:/gf1fdIYtTnsLe_LIINHhqV on /Volumes/MobileBackups (mtmfs, nosuid, read-only, nobrowse)
s3fs@osxfuse0 on /Users/ravi/s3-drive (osxfuse, nodev, nosuid, synchronous, mounted by ravi)
➜  ~ 
```

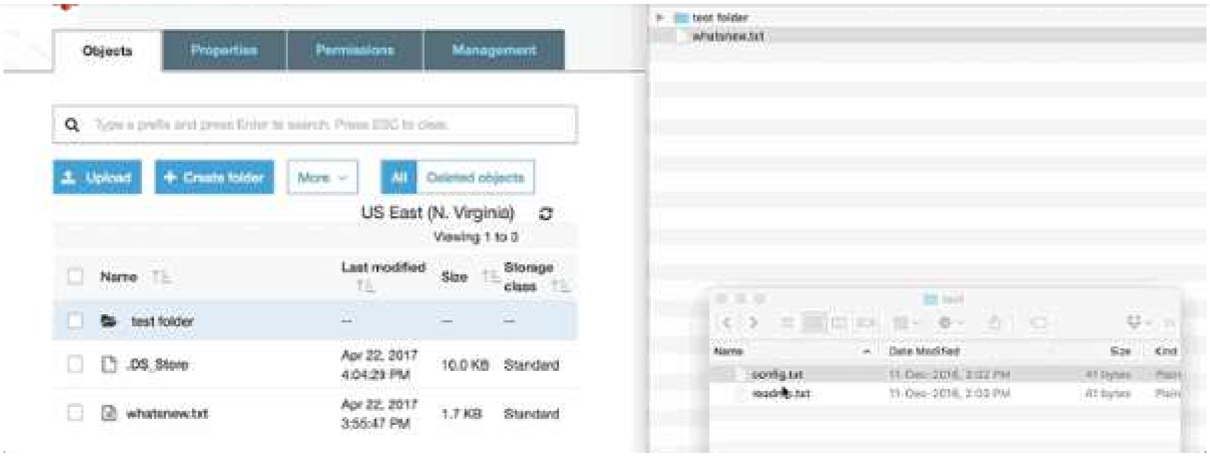
3. The previous command will mount the bucket on the Amazon S3-drive folder. Once mounted, you can interact with the Amazon S3 bucket same way as you would use any local folder.



Name	Last modified	Size	Storage class
test folder	--	--	--
.DS_Store	Apr 22, 2017 4:04:29 PM	10.0 KB	Standard
whatsnew.txt	Apr 22, 2017 3:56:47 PM	1.7 KB	Standard

Name	Date Modified	Size	Kind
test folder	Today, 4:03 PM	--	Folder
whatsnew.txt	Today, 9:20 PM	2 KB	Plain Text

In the screenshot above, you can see a bidirectional sync between MacOS and Amazon S3. The



Name	Last modified	Size	Storage class
test folder	--	--	--
.DS_Store	Apr 22, 2017 4:04:29 PM	10.0 KB	Standard
whatsnew.txt	Apr 22, 2017 3:55:47 PM	1.7 KB	Standard

Name	Date Modified	Size	Kind
config.txt	11-Dec-2016, 2:02 PM	41 bytes	Plain
readme.txt	11-Dec-2016, 2:03 PM	41 bytes	Plain

Considerations When Using Amazon S3 as a File System

Now that we've looked at the advantages of using Amazon S3 as a mounted drive, we should consider some of the points before using this approach.

- >> When you are using Amazon S3 as a file system, you might observe a network delay when performing IO centric operations such as creating or moving new folders or files. The performance depends on your network speed as well distance from Amazon S3 storage region.
- >> Since Amazon S3 is designed for atomic operations, files cannot be modified, they have to be completely replaced with modified files. This doesn't impact your application as long as it's creating or deleting files; however, if there are frequent modifications to a file, that means replacing the file on Amazon S3 repeatedly, which results in multiple put requests and, ultimately, higher costs.
- >> As files are transferred via HTTPS, whenever your application tries to access the mounted Amazon S3 bucket first time, there is noticeable delay. Future or subsequent access times can be delayed with local caching.
- >> Each object has a maximum size of 5GB. When considering costs, remember that Amazon S3 charges you for performing IO operations. The overall object might cost less, comparing General Purpose SSD to IOPS storage for example, but the cost for IO could be higher.
- >> NetApp can help cut Amazon AWS storage costs with powerful storage efficiencies and other benefits, more of which we'll discuss below.

NetApp Solutions for Using Amazon S3 for File System Storage

One way that NetApp offers you a shortcut in using Amazon S3 for file system storage is with Cloud

With data tiering to Amazon S3 Cloud Volumes ONTAP can send infrequently-accessed files to Amazon S3 (the cold data tier), where prices are lower than on Amazon EBS. When the data is needed for active use, it can be brought back up to Amazon EBS (the hot data tier), where the costs are higher but the performance is much better than on Amazon S3.

To use Amazon S3 as a file system storage, you could configure data tiering with Cloud Volumes ONTAP to create an NFS/CIFS share which has back-end storage in Amazon S3. From this Amazon S3-backed file share you could mount from multiple machines at the same time, effectively treating it as a regular file share. (Note that in this case that you would only be able to access the files over NFS/CIFS from Cloud Volumes ONTAP and not through Amazon S3.)

It is possible to move and preserve a file system in Amazon S3, from where the file system would remain fully usable and accessible. However, one consideration is how to migrate the file system to Amazon S3. One option would be to use Cloud Sync. Cloud Sync is NetApp's solution for fast and easy data migration, data synchronization, and data replication between NFS and CIFS file shares, Amazon S3, NetApp StorageGRID® Webscale Appliance, and more. Cloud Sync can also migrate and transfer data to and from Amazon EFS, AWS's native file share service.

Conclusion

From the steps outlined above you can see that it's simple to mount S3 bucket to EC2 instances, servers, laptops, or containers.

Mounting Amazon S3 as drive storage can be very useful in creating distributed file systems with minimal effort, and offers a very good solution for media content-oriented applications. But since you are billed based on the number of GET, PUT, and LIST operations you perform on Amazon S3, mounted Amazon S3 file systems can have a significant impact on costs, if you perform such operations frequently.

This mechanism can prove very helpful when scaling up legacy apps, since those apps run without any modification in their codebases. Having a shared file system across a set of servers can be beneficial when you want to store resources such as config files and logs in a central location. However, AWS does not recommend this due to the size limitation, increased costs, and decreased IO performance. But for some users the benefits of added durability in a distributed file system functionality may outweigh those considerations.

With NetApp, you might be able to mitigate the extra costs that come with mounting Amazon S3 as a

To offset the higher costs with using Amazon S3 as a file system, try a 30-day free trial of Cloud Volumes ONTAP on AWS and a free 14-day Cloud Sync trial today.

-

© 2018 NetApp, Inc.