

To Interact with S3 from Amazon CLI:

Install the AWS CLI on Windows

1. Download the appropriate MSI installer.
 - [Download the AWS CLI MSI installer for Windows \(64-bit\)](#)
 - [Download the AWS CLI MSI installer for Windows \(32-bit\)](#)
2. Note
3. The 64-bit version of the AWS CLI does not currently work with Windows Server 2008 (version 6.0.6002). Please use the 32-bit installer with this version of Windows.
4. Run the downloaded MSI installer.
5. Follow the instructions that appear.

Confirm the installation

To confirm the installation, use the `aws --version` command at a command prompt (open the START menu and search for “cmd” if you’re not sure how to find the command prompt).

64 Bit

The CLI installs to C:\Program Files\Amazon\AWSCLI

```
1 C:\Program Files\Amazon\AWSCLI>aws --version
```

```
2 aws-cli/1.7.24 Python/2.7.9 Windows/8
```

32 Bit

The CLI installs to C:\Program Files (x86)\Amazon\AWSCLI

```
1 C:\Program Files (x86)\Amazon\AWSCLI>aws --version
```

```
2 aws-cli/1.7.24 Python/2.7.9 Windows/7
```

To be able to connect to your AWS account, you'll need one more step.

For Windows, the...

```
1 aws configure
```

...command is the fastest way to set up your AWS CLI installation. Through *aws configure*, the AWS CLI will prompt you for four pieces of information. The AWS Access Key ID and AWS Secret Access Key are your account credentials. Those you will definitely need to provide. You can probably leave the other two – region and output format – as default for the time being.

```
1 aws configure
```

```
2 AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
3 AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

```
4 Default region name [None]: us-west-2
```

```
5 Default output format [None]: json
```

That should be it. Try out the following command from your cmd prompt and, if you have any s3 buckets, you should see them listed

```
1 aws s3 ls
```

Now to work with S3-

Managing Buckets

High-level `aws s3` commands support commonly used bucket operations, such as creating, removing, and listing buckets.

Creating Buckets

Use the `aws s3 mb` command to create a new bucket. Bucket names must be unique and should be DNS compliant. Bucket names can contain lowercase letters, numbers, hyphens and periods. Bucket names can only start and end with a letter or number, and cannot contain a period next to a hyphen or another period.

```
$ aws s3 mb s3://bucket-name
```

Removing Buckets

To remove a bucket, use the `aws s3 rb` command.

```
$ aws s3 rb s3://bucket-name
```

By default, the bucket must be empty for the operation to succeed. To remove a non-empty bucket, you need to include the `--force` option.

```
$ aws s3 rb s3://bucket-name --force
```

This will first delete all objects and subfolders in the bucket and then remove the bucket.

Note

If you are using a versioned bucket that contains previously deleted—but retained—objects, this command will *not* allow you to remove the bucket.

Listing Buckets

To list all buckets or their contents, use the `aws s3 ls` command. Here are some examples of common usage.

The following command lists all buckets.

```
$ aws s3 ls
      CreationTime Bucket
      -----
2013-07-11 17:08:50 my-bucket
2013-07-24 14:55:44 my-bucket2
```

The following command lists all objects and folders (prefixes) in a bucket.

```
$ aws s3 ls s3://bucket-name
```

```
Bucket: my-bucket
```

```
Prefix:
```

LastWriteTime	Length	Name
-----	-----	----
		PRE path/
2013-09-04 19:05:48	3	MyFile1.txt

The following command lists the objects in *bucket-name*/path (in other words, objects in *bucket-name* filtered by the prefix path).

```
$ aws s3 ls s3://bucket-name/path
```

```
Bucket: my-bucket
```

```
Prefix: path/
```

LastWriteTime	Length	Name
-----	-----	----
2013-09-06 18:59:32	3	MyFile2.txt

Managing Objects

The high-level `aws s3` commands make it convenient to manage Amazon S3 objects as well. The object commands include `aws s3 cp`, `aws s3 ls`, `aws s3 mv`, `aws s3 rm`, and `sync`. The `cp`, `ls`, `mv`, and `rm` commands work similarly to their Unix counterparts and enable you to work seamlessly across your local directories and Amazon S3 buckets. The `sync` command synchronizes the contents of a bucket and a directory, or two buckets.

Note

All high-level commands that involve uploading objects into an Amazon S3 bucket (`aws s3 cp`, `aws s3 mv`, and `aws s3 sync`) automatically perform a multipart upload when the object is large.

Failed uploads cannot be resumed when using these commands. If the multipart upload fails due to a timeout or is manually cancelled by pressing

CTRL+C, the AWS CLI cleans up any files created and aborts the upload. This process can take several minutes.

If the process is interrupted by a kill command or system failure, the in-progress multipart upload remains in Amazon S3 and must be cleaned up manually in the AWS Management Console or with the [s3api abort-multipart-upload](#) command.

The `cp`, `mv`, and `sync` commands include a `--grants` option that can be used to grant permissions on the object to specified users or groups. You set the `--grants` option to a list of permissions using following syntax:

```
--grants Permission=Grantee_Type=Grantee_ID  
        [Permission=Grantee_Type=Grantee_ID ...]
```

Each value contains the following elements:

- *Permission* – Specifies the granted permissions, and can be set to `read`, `readacl`, `writeacl`, or `full`.
- *Grantee_Type* – Specifies how the grantee is to be identified, and can be set to `uri`, `emailaddress`, or `id`.
- *Grantee_ID* – Specifies the grantee based on *Grantee_Type*.
 - `uri` – The group's URI. For more information, see [Who Is a Grantee?](#)
 - `emailaddress` – The account's email address.
 - `id` – The account's canonical ID.

For more information on Amazon S3 access control, see [Access Control](#).

The following example copies an object into a bucket. It grants `read` permissions on the object to everyone and `full` permissions (`read`, `readacl`, and `writeacl`) to the account associated with `user@example.com`.

```
$ aws s3 cp file.txt s3://my-bucket/ --grants  
read=uri=http://acs.amazonaws.com/groups/global/AllUsers  
full=emailaddress=user@example.com
```

To specify a non-default storage class (`REDUCED_REDUNDANCY` or `STANDARD_IA`) for objects that you upload to Amazon S3, use the `--storage-class` option:

```
$ aws s3 cp file.txt s3://my-bucket/ --storage-class REDUCED_REDUNDANCY
```

