

Hands-on tutorials

Batch Upload Files to Amazon S3 Using the AWS CLI



Batch Upload Files to Amazon S3 Using the AWS CLI: Hands-on tutorials

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Batch Upload Files to Amazon S3 Using the AWS CLI	i
Overview	1
Implementation	2
Conclusion	11

Batch Upload Files to Amazon S3 Using the AWS CLI

AWS experience	Beginner
Time to complete	10 minutes
Cost to complete	Free Tier eligible
Requires	<ul style="list-style-type: none">• AWS Account <div data-bbox="862 632 1507 947"><p>Note</p><p>Accounts created within the past 24 hours might not yet have access to the services required for this tutorial.</p></div> <ul style="list-style-type: none">• Recommended browser: The latest version of Chrome or Firefox
Last updated	Aug 9, 2022

Overview

In this how-to guide, we are going to help you use the AWS Command Line Interface (AWS CLI) to access Amazon Simple Storage Service (Amazon S3). We will do this so you can easily build your own scripts for backing up your files to the cloud and easily retrieve them as needed. This will make automating your backup process faster, more reliable, and more programmatic. You can use this information to build a scheduled task (or cron job) to handle your backup operations.

Note

This guide builds upon the concepts from the [Store and Retrieve a File with Amazon S3](#) how-to guide. If you haven't done that guide yet, you should complete it first.

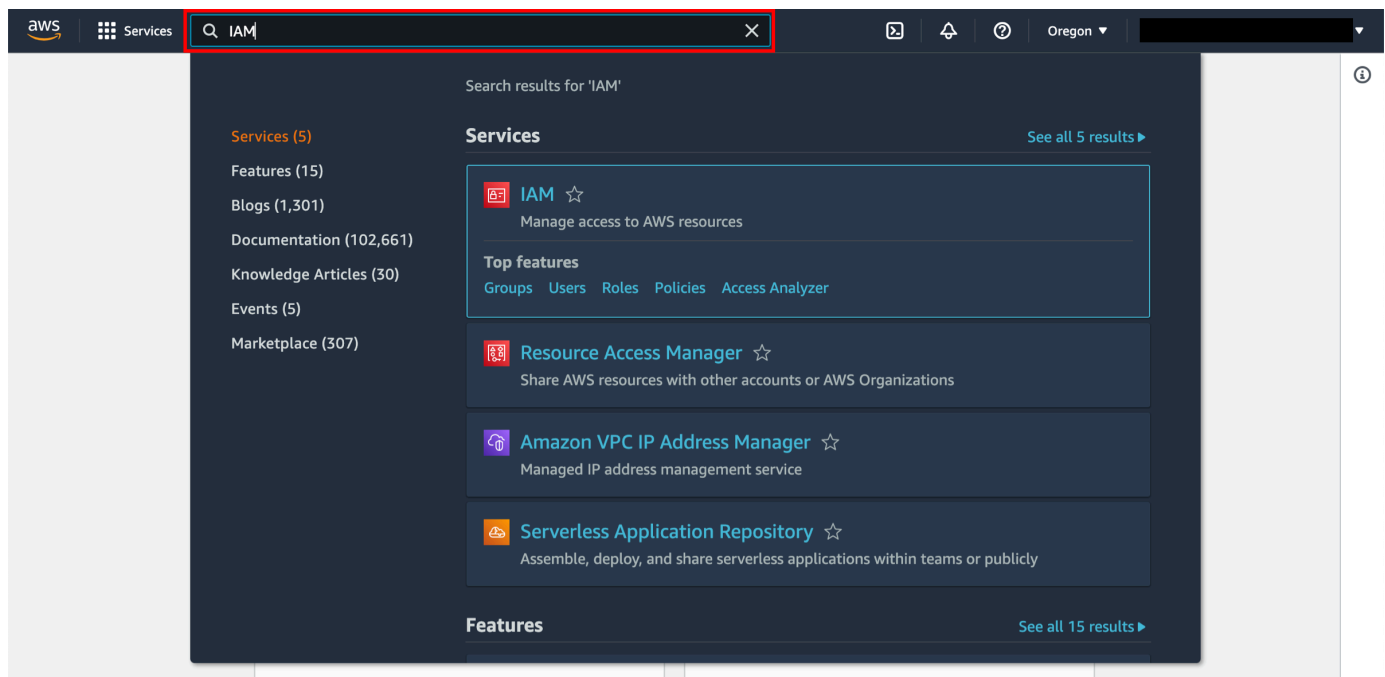
Implementation

Step 1: Create an AWS IAM User

In this step, you will use the IAM service to create a user account with administrative permission. In later steps, you will use this user account to securely access AWS services using the AWS CLI.

1. Sign in to the console

Click on the [AWS Management Console home](#) to open the console in a new browser window, so you can keep this step-by-step guide open. When this screen loads, enter your user name and password to get started. Then type IAM in the search bar and select **IAM** to open the Identity and Access Management dashboard.



2. Choose Users

From the AWS Identity and Access Management dashboard, click on Users on the left side.

The screenshot shows the AWS IAM dashboard. On the left is a navigation menu for Identity and Access Management (IAM) with options like Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity), and What's new. The main content area is titled 'IAM dashboard' and features 'Security recommendations' with two items: 'Add MFA for root user' and 'Add MFA for yourself', both with red warning icons. A green checkmark indicates that the user 'Administrator' does not have any active access keys unused for more than a year. Below this is a table of 'IAM resources':

User groups	Users	Roles	Policies	Identity providers
2	1	12	0	0

On the right side, there are sections for 'AWS Account' (Account ID, Account Alias, Sign-in URL), 'Quick Links' (My security credentials, Policy simulator), and 'Tools' (Policy simulator).

3. Create a user

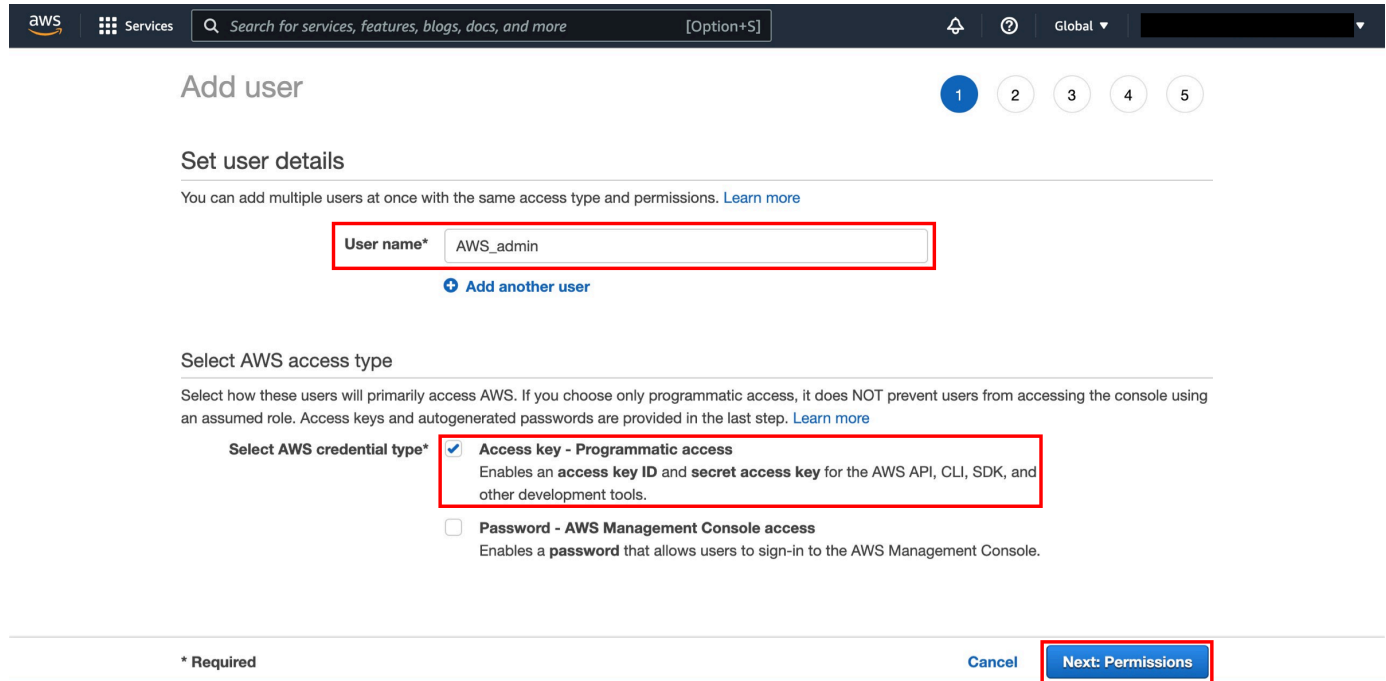
Click the **Add user** button.

The screenshot shows the 'Users' page in the AWS IAM console. The breadcrumb navigation is 'IAM > Users'. The page title is 'Users (1) Info'. Below the title is a description: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' There are 'Delete' and 'Add users' buttons, with the 'Add users' button highlighted with a red box. A search bar is present with the placeholder text 'Find users by username or access key'. Below the search bar is a table of users:

	User name	Groups	Last activity	MFA	Password a...
<input type="checkbox"/>	Administrator	Administrators	2 hours ago	None	2 days ago

4. Specify user details

Enter a user name in the textbox next to **User name:** (we'll use **AWS_Admin** for this example) and select **Programmatic access** in the Select AWS Access Type section. Click the **Next: Permissions** button.



The screenshot shows the AWS IAM console 'Add user' wizard. The top navigation bar includes the AWS logo, 'Services', a search bar, and a 'Global' dropdown. The main heading is 'Add user' with a progress indicator showing five steps, with step 2 'Set user details' selected. Below the heading, there is a note: 'You can add multiple users at once with the same access type and permissions. [Learn more](#)'. A text input field for 'User name*' contains 'AWS_admin'. Below it is a blue '+ Add another user' button. The 'Select AWS access type' section has a note: 'Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)'. Under 'Select AWS credential type*', there are two options: 'Access key - Programmatic access' (checked) and 'Password - AWS Management Console access' (unchecked). The 'Access key' option is highlighted with a red box. At the bottom, there is a '* Required' label, a 'Cancel' button, and a 'Next: Permissions' button highlighted with a red box.

5. Add permissions

Click on **Attach existing policies directly** option. Select **AdministratorAccess** then click **Next: Tags**.

Add user 1 2 3 4 5

▼ Set permissions

Add user to group
 Copy permissions from existing user
 Attach existing policies directly

Filter policies Showing 750 results

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (1)
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None

6. Add tags

IAM tags are key-value pairs you can add to your user. We'll skip this step for this example. Click the **Next: Review** button.

Add user 1 2 3 4 5

Add tags (optional)

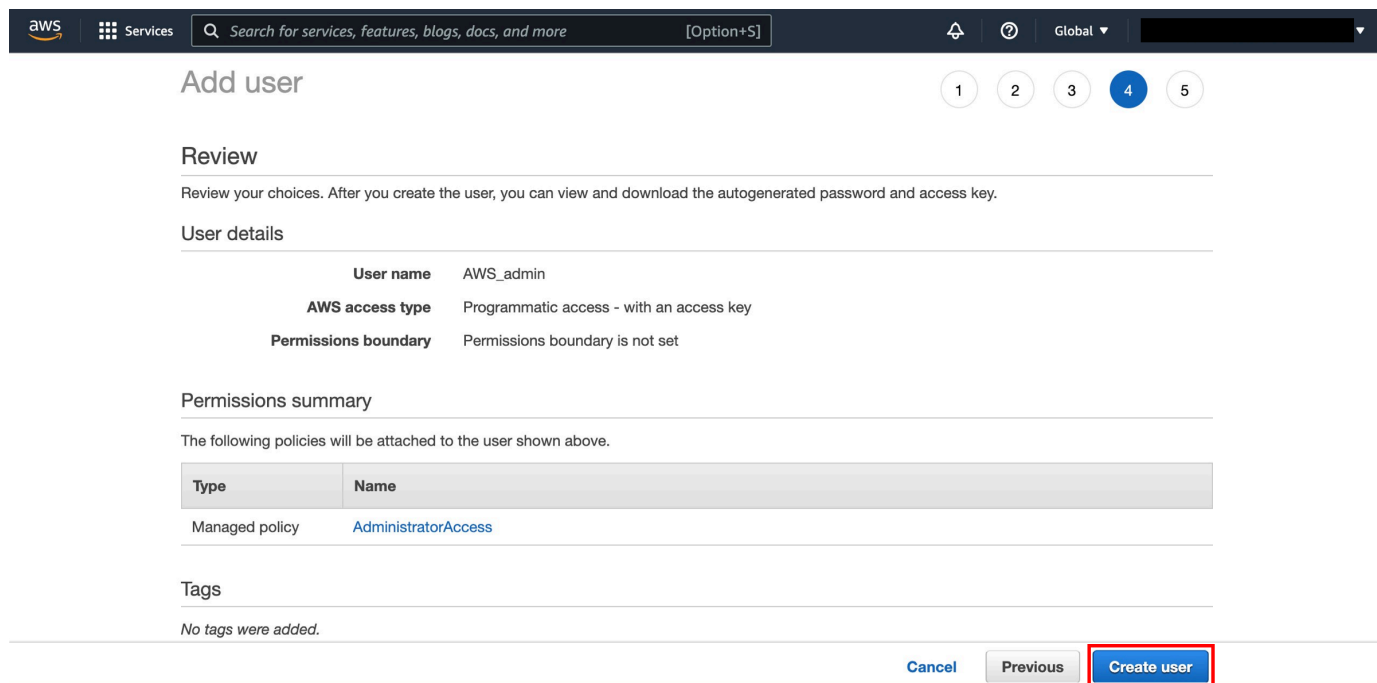
IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	<input type="button" value="Remove"/>

You can add 50 more tags.

7. Review and create

Take this opportunity to review that all settings are correct. When you are ready, click on **Create user**.



Add user 1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	AWS_admin
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess

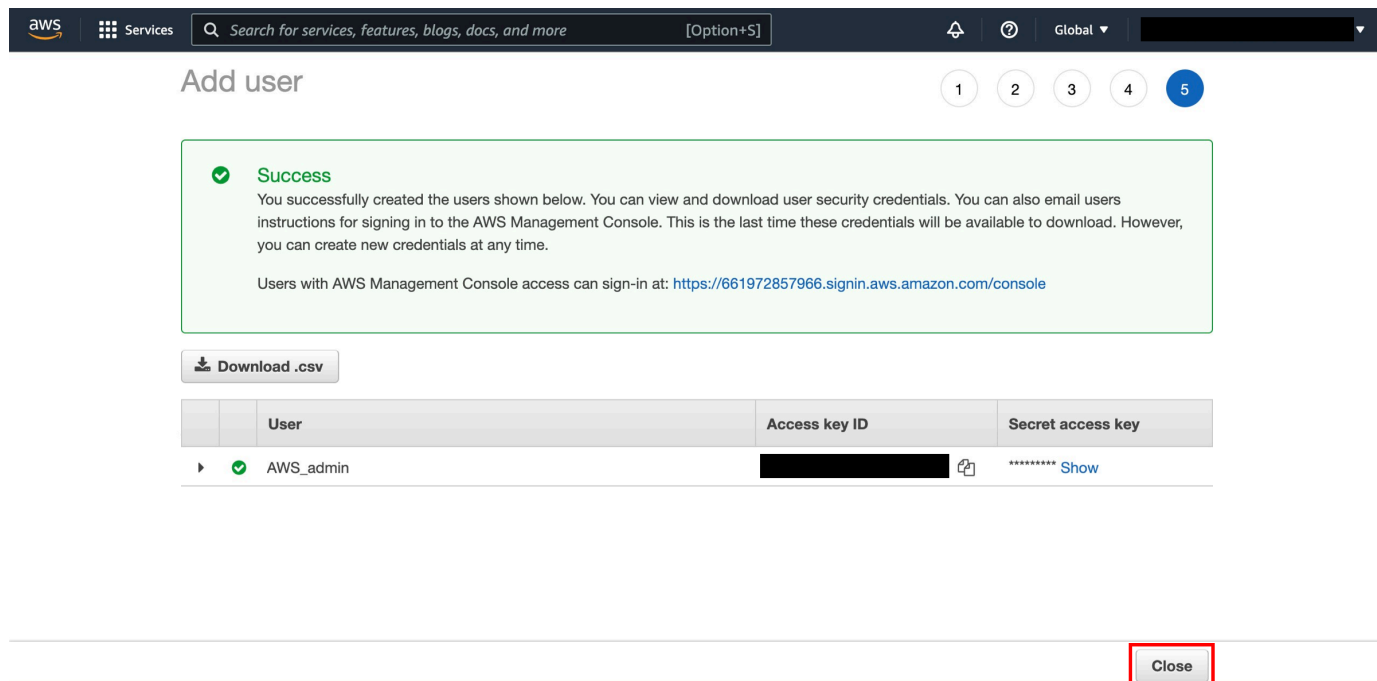
Tags

No tags were added.

Cancel Previous **Create user**

8. Review and create

Click the **Download Credentials** button and save the `credentials.csv` file in a safe location (you'll need this later in step 3) and then click the **Close** button.



Add user 1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://661972857966.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key
AWS_admin	[REDACTED]	***** Show


Close

Step 2: Install and configure the AWS CLI

Now that you have your IAM user, you need to install the AWS CLI. For instructions, select the tab that corresponds to your operating system.

Windows


1. Download and run the Windows installer ([64-bit](#), [32-bit](#)).

 **Note**

Users of Windows Server 2008 v6.0.6002 will need to use a different install method, listed in the [AWS Command Line Interface User Guide](#).


2. Open a command prompt by pressing the Windows Key + r to open the run box and enter cmd and press the OK button.
3. Type **aws configure** and press enter. When prompted, enter the following:

AWS Access Key ID [None]: Enter the **Access Key Id** from the **credentials.csv** file you downloaded earlier

 **Note**

This should look something like **AKIAIOSFODNN7EXAMPLE**

AWS Secret Access Key [None]: Enter the **Secret Access Key** from the **credentials.csv** file you downloaded earlier

 **Note**

This should look something like **je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY**

Default region name [None]: Enter **us-east-1**

Default output format [None]: Enter **json**

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\Users\adamglic>aws configure
AWS Access Key ID [None]: AK[REDACTED]
AWS Secret Access Key [None]: 2U[REDACTED]S
Default region name [None]: us-east-1
Default output format [None]: json

D:\Users\adamglic>
```

macOS / Linux

1. Follow [these directions](#) for installing the AWS CLI bundled installer.
2. **MacOS users:** Open a terminal window by pressing **Command + Space** and typing **terminal** in the search window. Then press **enter** to open the terminal window.

Linux users: Open a terminal window.

3. Type **aws configure** and press **enter**. Enter the following when prompted:

AWS Access Key ID [None]: Enter the **Access Key Id** from the **credentials.csv** file you downloaded earlier

Note

This should look something like **AKIAIOSFODNN7EXAMPLE**

AWS Secret Access Key [None]: Enter the **Secret Access Key** from the **credentials.csv** file you downloaded earlier

Note

This should look something like **je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY**

Default region name [None]: Enter **us-east-1**

Default output format [None]: Enter **json**

```
adamglic — bash — 80x24
Last login: Fri Dec 11 10:42:06 on ttys000
b8e856392176:~ adamglic$ aws configure
AWS Access Key ID [None]: AK[REDACTED]Q
AWS Secret Access Key [None]: 2U[REDACTED]S
Default region name [None]: us-east-1
Default output format [None]: json
b8e856392176:~ adamglic$
```

Step 3: Using the AWS CLI with Amazon S3

In this step, you will use the AWS CLI to create a bucket in Amazon S3 and copy a file to the bucket.

1. Create an S3 bucket

Creating a bucket is optional if you already have a bucket created that you want to use. To create a new bucket named my-first-backup-bucket type:

```
aws s3 mb s3://my-first-backup-bucket
```

Note

Bucket naming has some restrictions; one of those restrictions is that bucket names must be globally unique (for example, two different AWS users can not have the same bucket name); because of this, if you try the command above you will get a BucketAlreadyExists error.

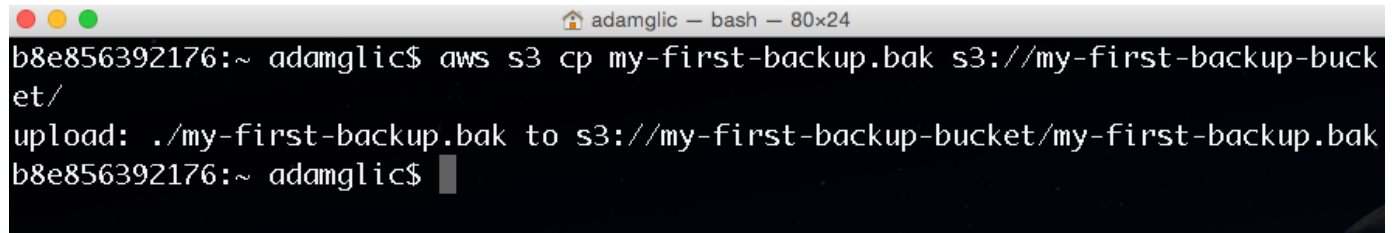
```
adamglic — bash — 80x24
b8e856392176:~ adamglic$ aws s3 mb s3://my-first-backup-bucket
make_bucket: s3://my-first-backup-bucket/
b8e856392176:~ adamglic$
```

2. Upload files to Amazon S3

To upload the file **my first backup.bak** located in the local directory (C:\users) to the S3 bucket **my-first-backup-bucket**, you would use the following command:

```
aws s3 cp "C:\users\my first backup.bak" s3://my-first-backup-bucket/
```

Or, use the original syntax if the filename contains no spaces.

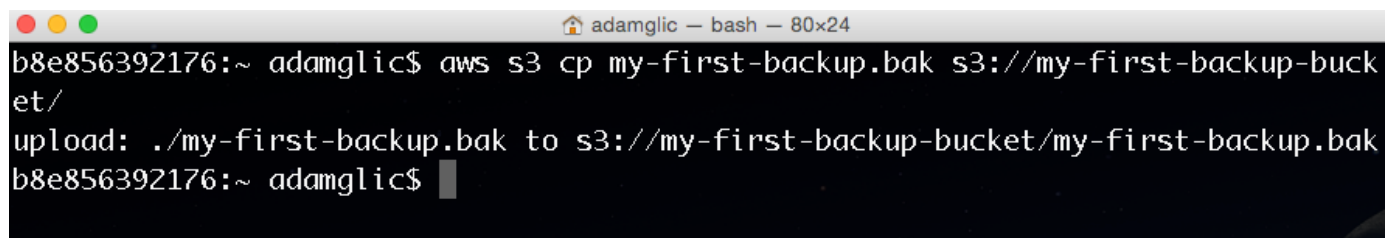


```
adamglic — bash — 80x24
b8e856392176:~ adamglic$ aws s3 cp my-first-backup.bak s3://my-first-backup-bucket/
upload: ./my-first-backup.bak to s3://my-first-backup-bucket/my-first-backup.bak
b8e856392176:~ adamglic$
```

3. Download files from Amazon S3

To download **my-first-backup.bak** from S3 to the local directory we would reverse the order of the commands as follows:

```
aws s3 cp s3://my-first-backup-bucket/my-first-backup.bak ./
```

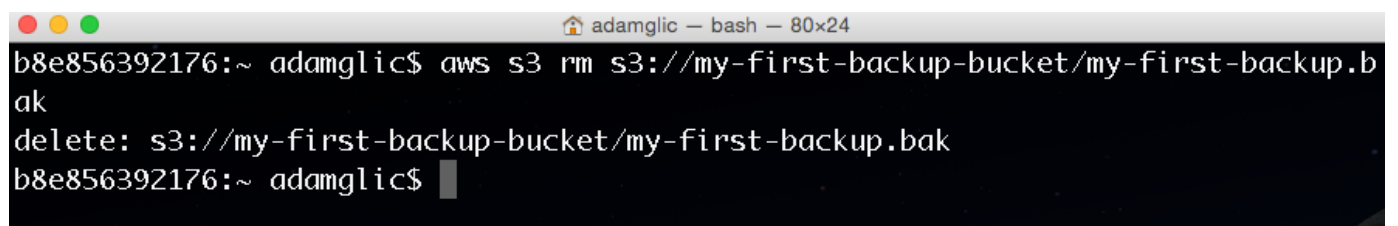


```
adamglic — bash — 80x24
b8e856392176:~ adamglic$ aws s3 cp s3://my-first-backup-bucket/my-first-backup.bak ./
upload: s3://my-first-backup-bucket/my-first-backup.bak to ./my-first-backup.bak
b8e856392176:~ adamglic$
```

4. Delete files from Amazon S3

To delete **my-first-backup.bak** from your **my-first-backup-bucket** bucket, use the following command:

```
aws s3 rm s3://my-first-backup-bucket/my-first-backup.bak
```



```
adamglic — bash — 80x24
b8e856392176:~ adamglic$ aws s3 rm s3://my-first-backup-bucket/my-first-backup.bak
delete: s3://my-first-backup-bucket/my-first-backup.bak
b8e856392176:~ adamglic$
```

Conclusion

Congratulations! You have set up an IAM user, configured your machine for use with the AWS Command Line Interface, and learned how to create, copy, retrieve, and delete files from the cloud.