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

Table of Contents

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	• AWS Account **Note** Accounts created within the past 24 hours might not yet have access to the services required for this tutorial. • 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.

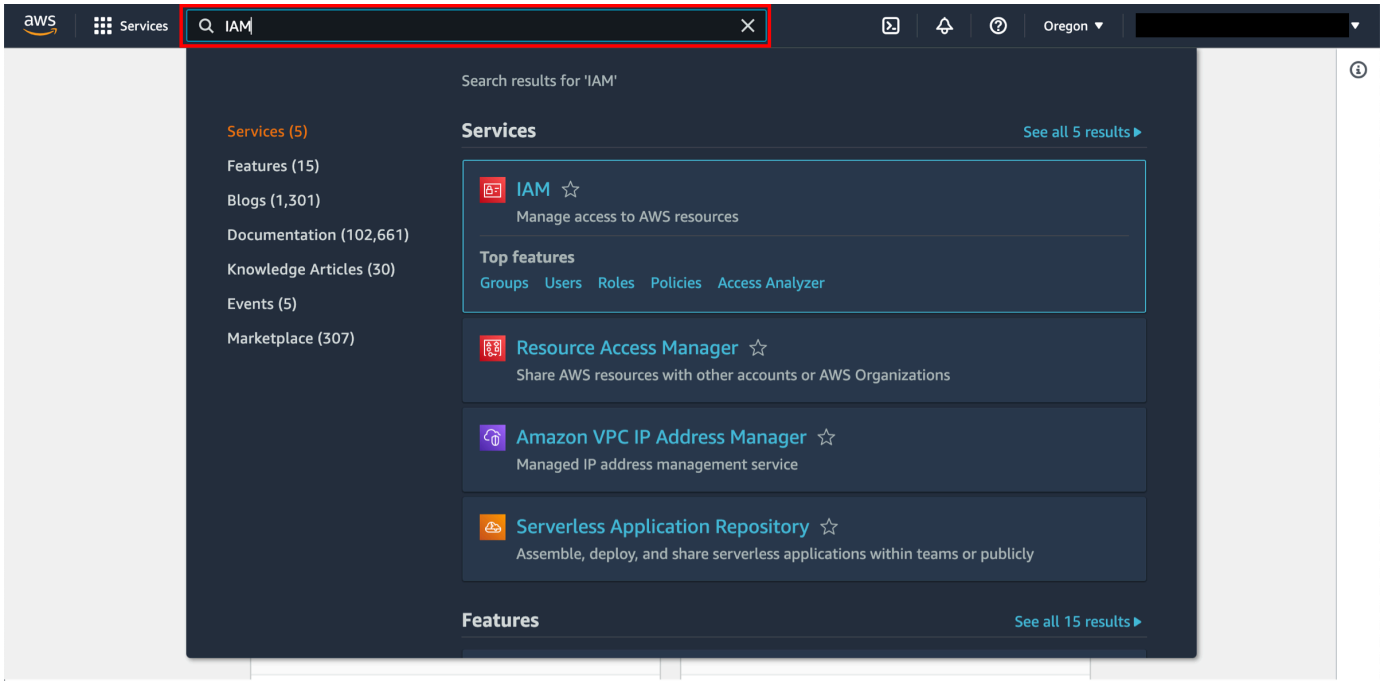
Implementation

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.

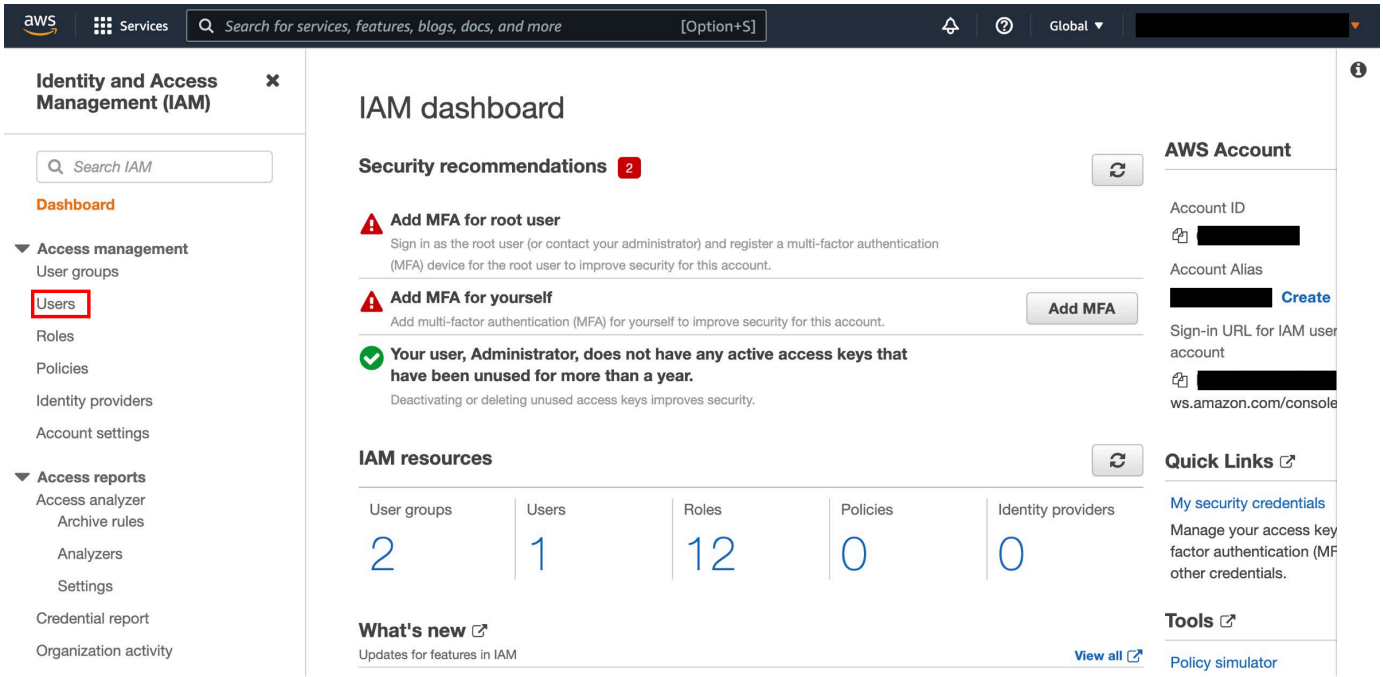
1. Sign in to the console

   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.



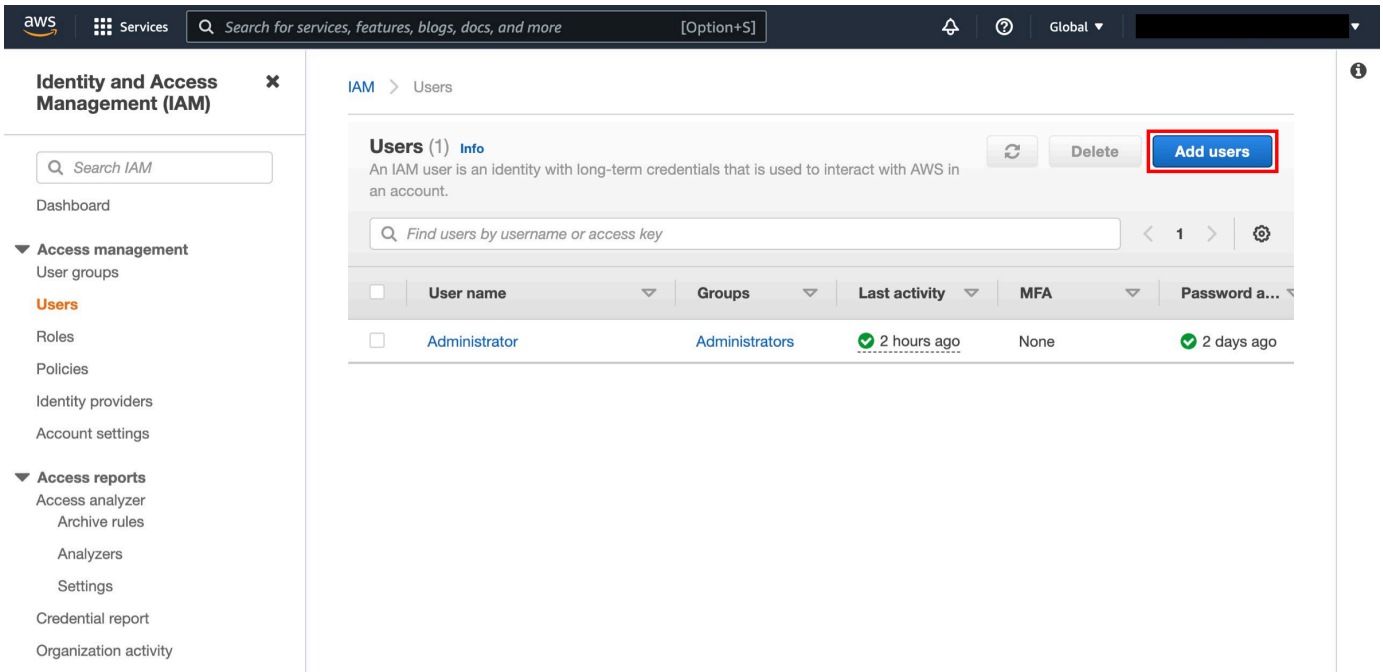
2. Choose Users

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



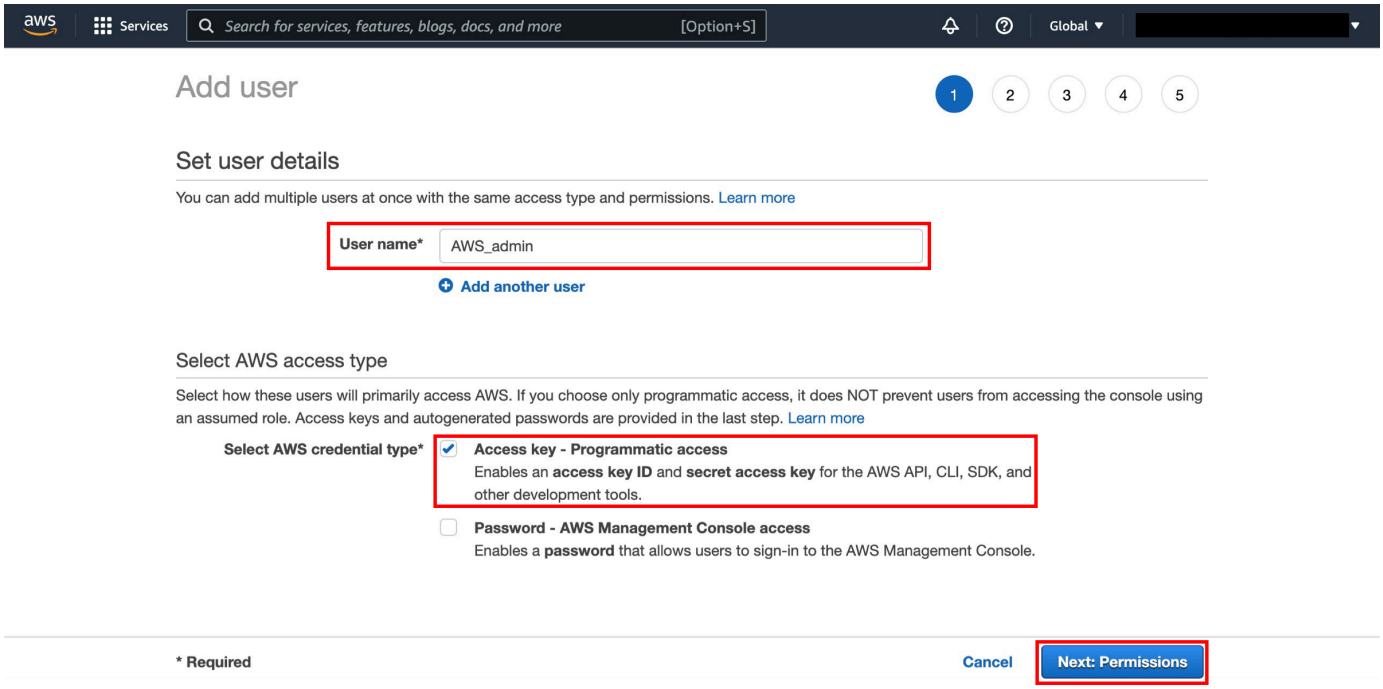
3. Create a user

Click the **Add user** button.



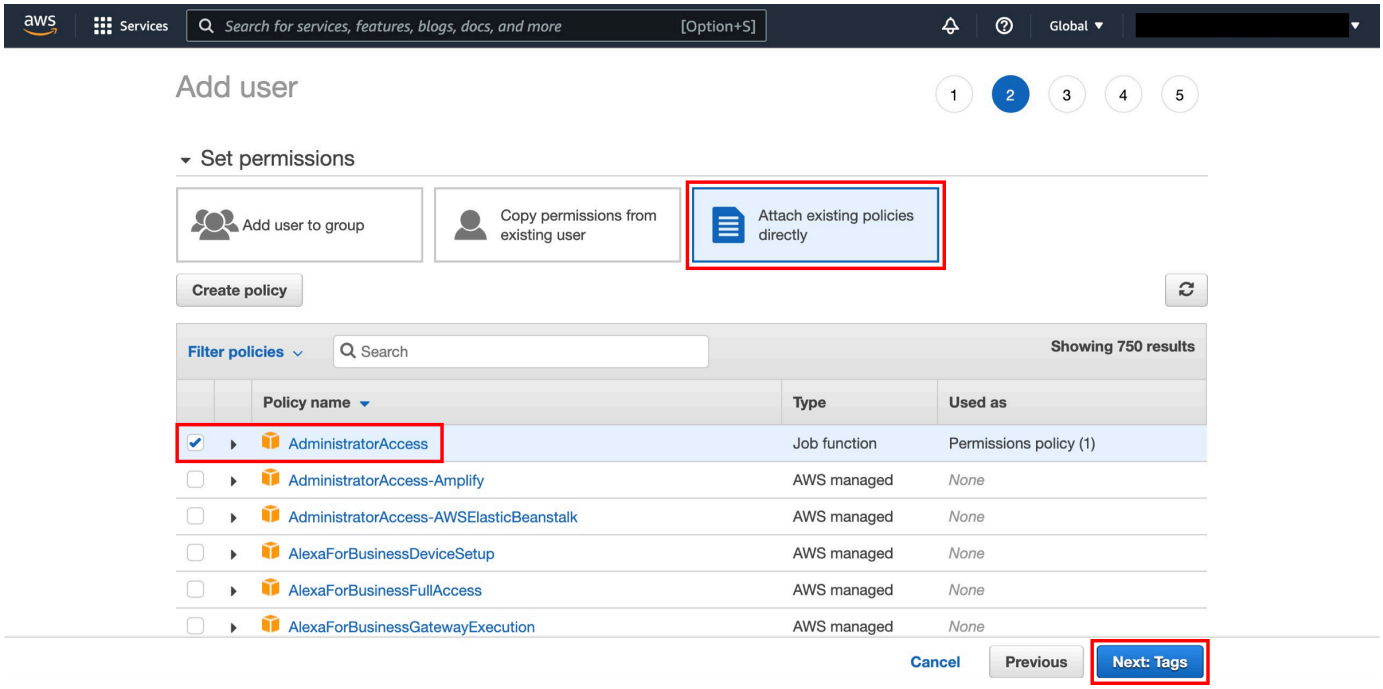
4. Specify user details

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.



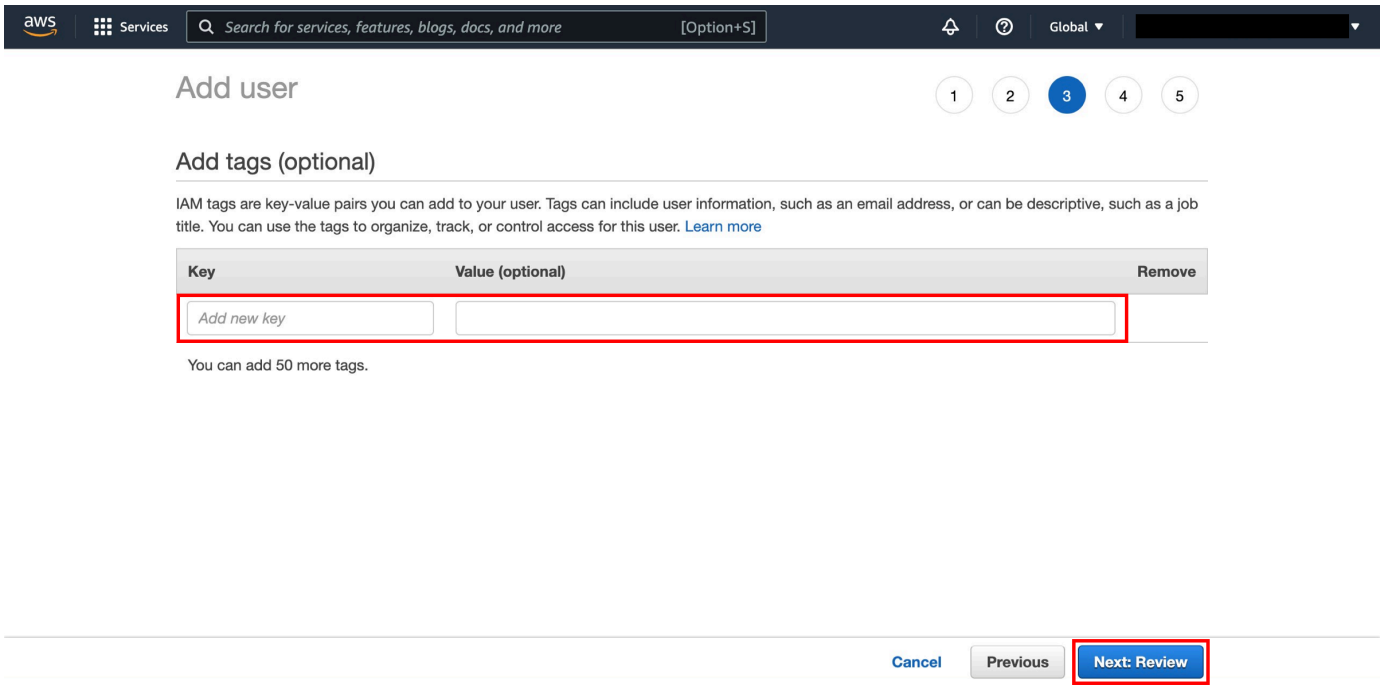
5. Add permissions

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



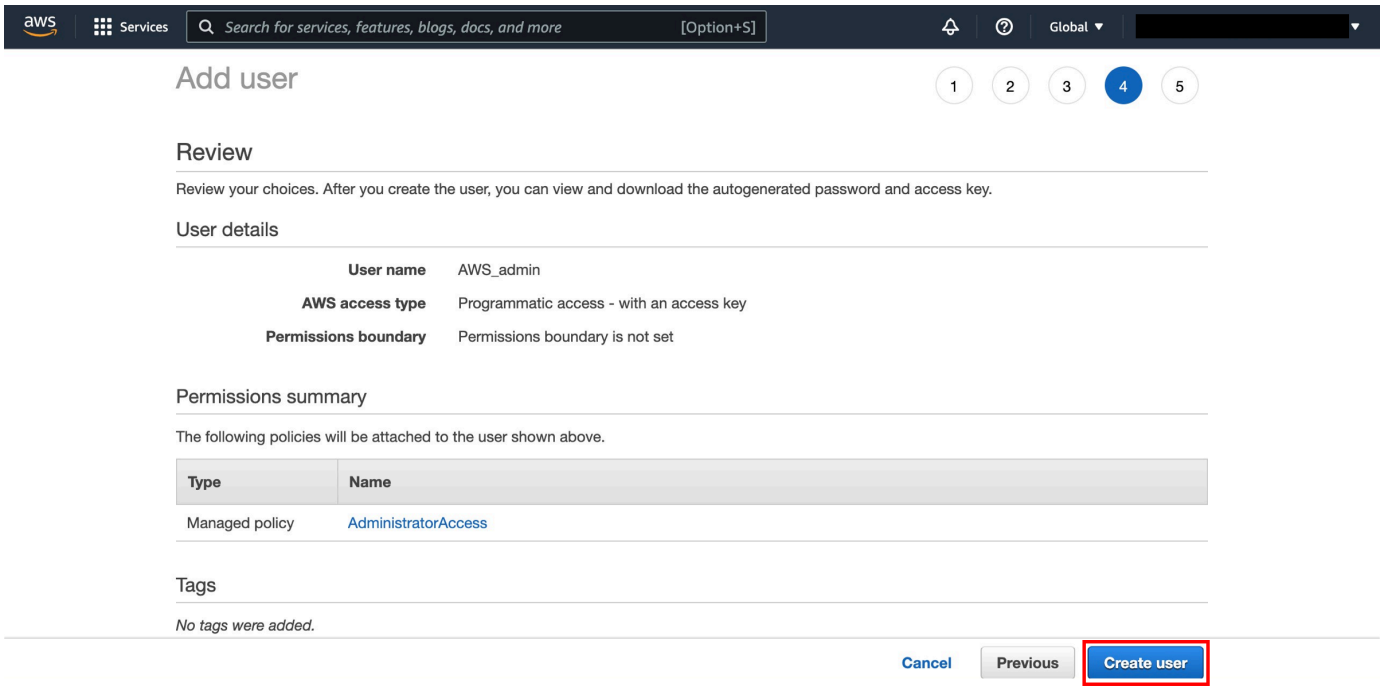
6. Add tags

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.



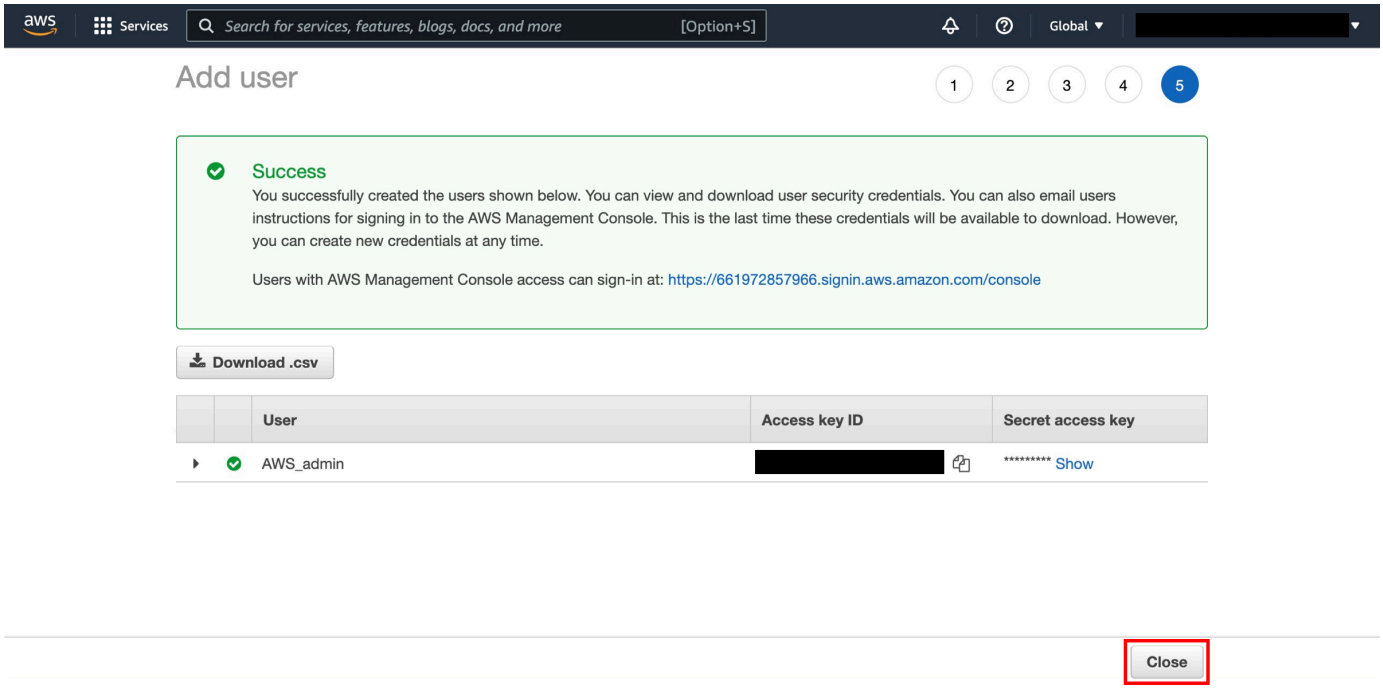
7. Review and create

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



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]Q
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 — 80×24
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 — 80×24
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 — 80×24
b8e856392176:~ adamglic$ aws s3 cp my-first-backup.bak s3://my-first-backup-buck
et/
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 — 80×24
b8e856392176:~ adamglic$ aws s3 cp my-first-backup.bak s3://my-first-backup-buck
et/
upload: ./my-first-backup.bak to s3://my-first-backup-bucket/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 — 80×24
b8e856392176:~ adamglic$ aws s3 rm s3://my-first-backup-bucket/my-first-backup.b
ak
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.