

# Terraform Setup

# The Setup Process

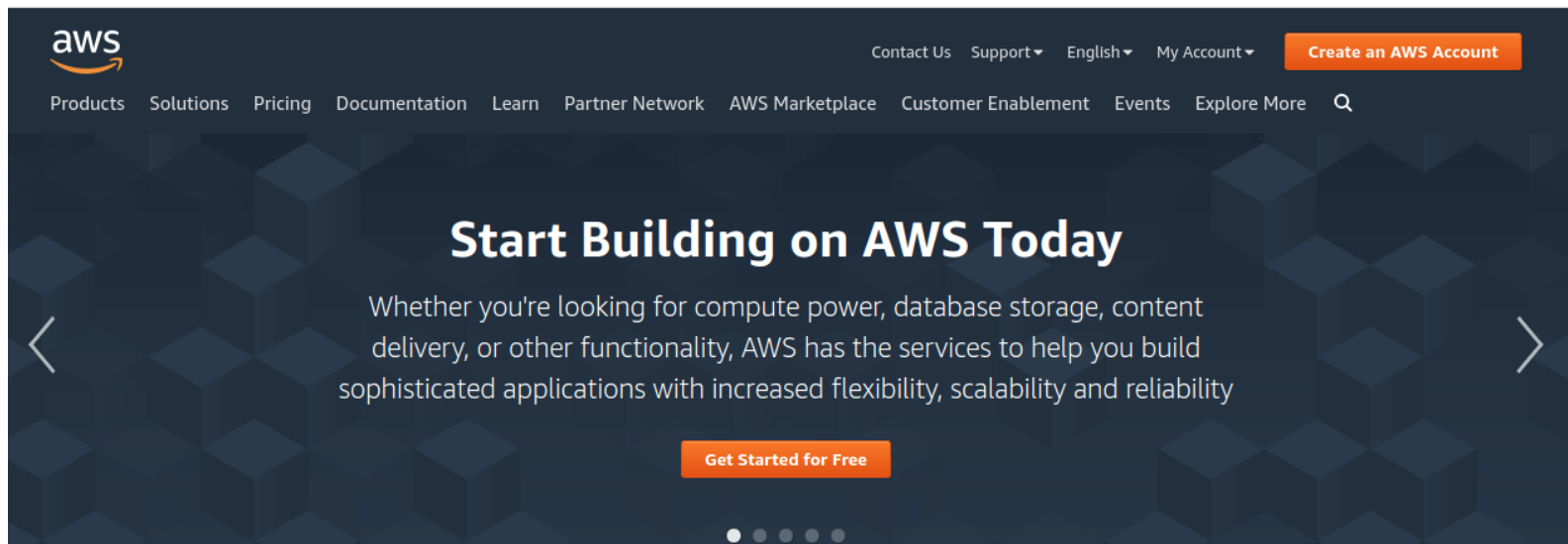
- ◆ This slide deck demonstrates the process in lab01-01
- ◆ Not all of the setup steps will apply to your configuration
- ◆ The steps of the setup process are:
  - Getting your AWS account or ID created or configured
  - Setting up an AWS IAM user for the class work
  - Installing the AWS CLI (command line interface)
  - Configuring your AWS credentials on your local machine
  - Downloading and installing terraform
  - Running the "Hello World" terraform test script

# Your AWS Account

- ◆ You need access to an AWS account for this class
- ◆ This setup document will cover the following three cases:
  - You are creating a new AWS account for this class
  - You are going to be using your own AWS account for the class
  - You will be using a class provided AWS ID

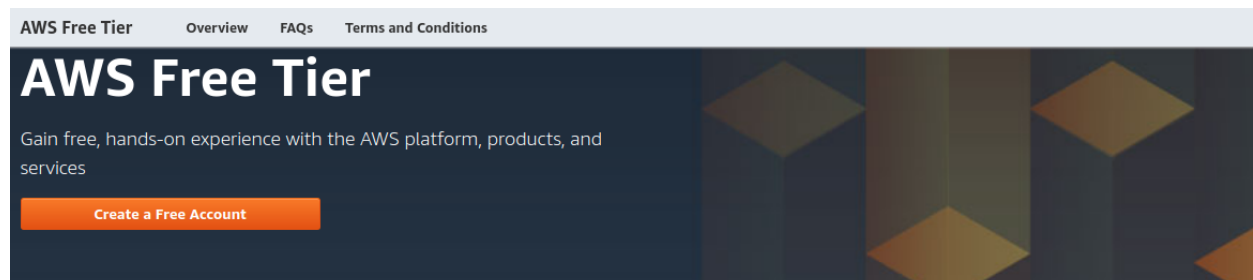
# Step One: New Account

- ◆ If you already have an AWS account or an ID, you can skip to step two
- ◆ You can create a new free-tier AWS account by going to: <https://aws.amazon.com/>
- ◆ Selecting the “*Create an AWS Account*” button will walk you through the process of setting up and AWS account
  - You will need a credit or debit card to set up the account
  - You will also need to use an email address that has not been used to set up an AWS account



# Step One: Free Tier Accounts

- ◆ Your account is not "Free" but allows you to access some AWS products and services for free
- ◆ *If you are new to AWS, read the details of the Free Tier Account*
  - If you use AWS resources that are *not* in the free tier, you could incur charges
  - **Keeping your resource usage within the limits of the free tier is YOUR responsibility**



## Types of offers

Explore more than 100 products and start building on AWS using the Free Tier. Three different types of free offers are available depending on the product used. See below for details on each product.



### Always free

These free tier offers do not expire and are available to all AWS customers



### 12 months free

Enjoy these offers for 12-months following your initial sign-up date to AWS



### Trials

Short-term free trial offers start from the date you activate a particular service

# Step One: Free Tier Usage Rates

- ◆ You are allowed a certain amount of free AWS resource usage
- ◆ **It is your responsibility to ensure you do not exceed these limits**
  - The instructor will provide pointers on how to ensure your AWS resources in class are cleaned up

**Free Tier details**

**Filter by:**  
[Clear all filters](#)

▼ **Tier Type**

- ☐ Featured
- ☐ 12 Months Free
- ☐ Always Free
- ☐ Trials

▼ **Product Categories**

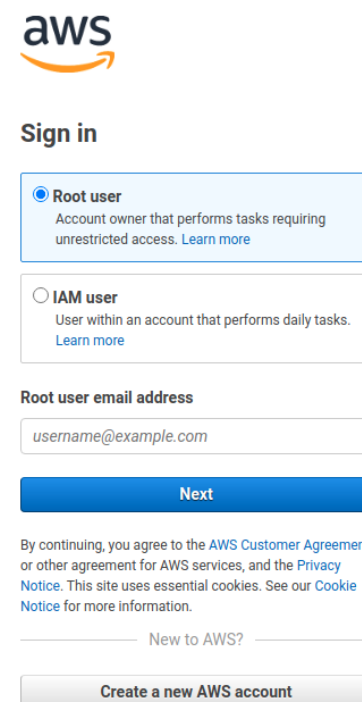
- ☐ Analytics
- ☐ Application Integration
- ☐ Business Productivity
- ☐ Compute
- ☐ Customer Engagement
- ☐ Database

Search free tier products

COMPUTE	STORAGE	DATABASE
<p>Free Tier 12 MONTHS FREE</p> <p><b>Amazon EC2</b></p> <p><b>750 Hours</b></p> <p>per month</p> <p>Resizable compute capacity in the Cloud.</p> <p>750 hours per month of Linux, RHEL, or SLES</p>	<p>Free Tier 12 MONTHS FREE</p> <p><b>Amazon S3</b></p> <p><b>5 GB</b></p> <p>of standard storage</p> <p>Secure, durable, and scalable object storage infrastructure.</p> <p>5 GB of Standard Storage</p>	<p>Free Tier 12 MONTHS FREE</p> <p><b>Amazon RDS</b></p> <p><b>750 Hours</b></p> <p>per month of db.t2.micro database usage (applicable DB engines)</p> <p>Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, Oracle BYOL, or SQL Server.</p>

# Step Two: Administrative User

- ◆ This does not apply to those who are using a class provided AWS ID
  - If you are using your own account, you may have already done this
- ◆ When you are logging in with your email, you are the root user
  - You should never use this account for day-to-day operations
  - It should be only used for billing related work
- ◆ Instead, AWS recommends setting up an administrative IAM user



The screenshot shows the AWS Sign in page. At the top is the AWS logo. Below it is the 'Sign in' heading. There are two radio button options: 'Root user' (selected) and 'IAM user'. The 'Root user' option has a description: 'Account owner that performs tasks requiring unrestricted access. [Learn more](#)'. The 'IAM user' option has a description: 'User within an account that performs daily tasks. [Learn more](#)'. Below these options is a text input field for 'Root user email address' with the placeholder text 'username@example.com'. A blue 'Next' button is below the input field. At the bottom, there is a link to the AWS Customer Agreement and Privacy Notice, a link to the Cookie Notice, a 'New to AWS?' link, and a 'Create a new AWS account' button.

aws

Sign in

☒ **Root user**  
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**  
User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

[New to AWS?](#)

Create a new AWS account

# Step Two: Creating the User

- ◆ Go to the IAM service and create a new user
- ◆ Ensure the user has both console and programmatic access
- ◆ Set the password to what you want
- ◆ Disable the "Require Password Reset" option
- ◆ Select "Next"

## Add user



### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[+ Add another user](#)

### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type\* ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

- Console password\* ☐ Autogenerated password  
☒ Custom password

☐ Show password

Require password reset ☐ **User must create a new password at next sign-in**  
Users automatically get the `IAMUserChangePassword` policy to allow them to change their own password.



# Step Two: Adding Permissions

- ◆ Select the option to *attach existing policies directly*
- ◆ Select the *AdministratorAccess* policy
  - You may have to search for it

Add user

1 2 3 4 5

▼ Set permissions



Add user to group



Copy permissions from existing user



Attach existing policies directly

Create policy



Filter policies ▼

Search

Showing 681 results

	Policy name ▼	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (2)
<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
<input type="checkbox"/>	AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None

# Step Two: Review the User

- ◆ Press "Next" until you get to the `Review` screen and ensure your user configuration looks like the screenshot
  - If not, go back and make the necessary changes
  - If it matches, press "Create User"

## Add user



## Review

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

## User details

<b>User name</b>	AdminUser
<b>AWS access type</b>	Programmatic access and AWS Management Console access
<b>Console password type</b>	Custom
<b>Require password reset</b>	No
<b>Permissions boundary</b>	Permissions boundary is not set

## Permissions summary

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

[Cancel](#)

[Previous](#)

[Create user](#)

# Step Two: Success Screen

- ◆ Once the user has been created, you should see screen below
  - Bookmark the URL for AWS Management Console access
  - YoHu don't need to download the .csv file, just click on "Close"

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://roddavison.signin.aws.amazon.com/console>



Download .csv

	User	Access key ID	Secret access key	Email login instructions
▶	✓ AdminUser	AKIA6KD2BMBQWHCXXKWQ	***** Show	Send email

Close

# Step Two: Login as the Administrative User

- ◆ Log out as the root user and login with the new Admin account ID
  - Use the URL you bookmarked



## Sign in as IAM user

Account ID (12 digits) or account alias

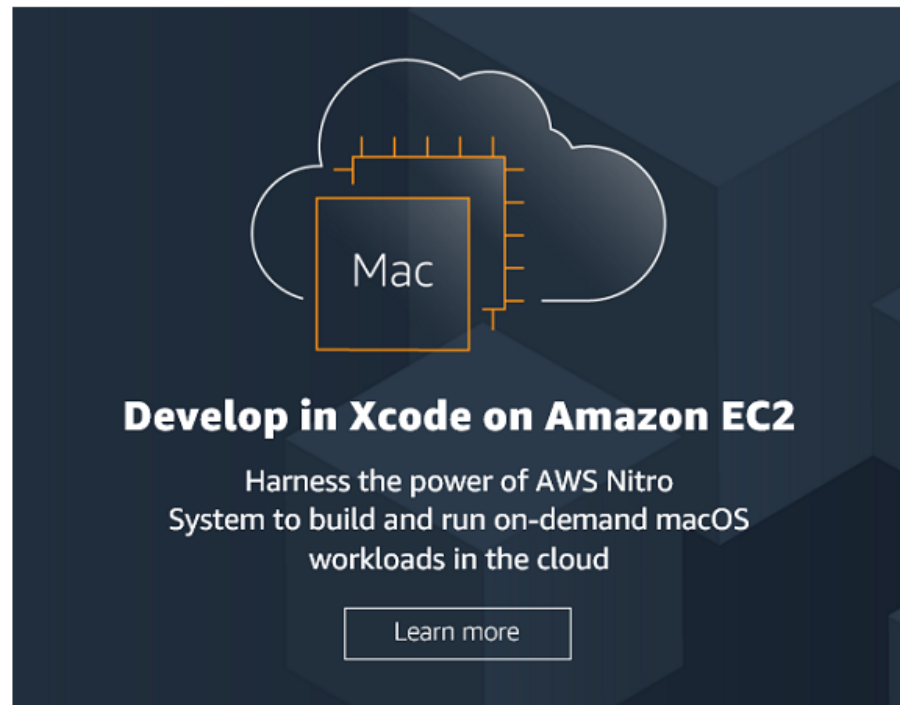
IAM user name

Password

[Sign in](#)

[Sign in using root user email](#)

[Forgot password?](#)



## Step Two: Create a Developer User

- ◆ For classwork, we will create a developer account which will not have full administrative access
  - This follows AWS recommendations for best account management practices
- ◆ The first few steps are the same as for creating the AdminUser account
  - Ensure that the user has both console and programmatic access
  - Programmatic access is needed to run terraform code
  - Console access allow visual confirmation of the results of running the Terraform code
- ◆ I have called this user "Dev"
  - You can either give `Dev` the same administration permissions as your AdminUser
  - Or you can give the AWS permissions shown on the next slide
  - You can always change this later

# Step Two: Restrictive Permissions

## Review

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

## User details

User name	Dev
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

## Permissions summary

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

Type	Name
Managed policy	<a href="#">AmazonS3FullAccess</a>
Managed policy	<a href="#">AmazonEC2FullAccess</a>
Managed policy	<a href="#">AmazonDynamoDBFullAccess</a>
Managed policy	<a href="#">IAMFullAccess</a>
Managed policy	<a href="#">CloudWatchFullAccess</a>
Managed policy	<a href="#">AmazonRDSDataFullAccess</a>

## Step Two: Download Access Keys

- ◆ You will need to download the .csv files since we need the AWS access keys to set up terraform access to AWS
- ◆ Alternatively, you can just copy them from the display and store them yourself in a text file
- ◆ Logout of the AdminUser account

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://roddavison.signin.aws.amazon.com/console>

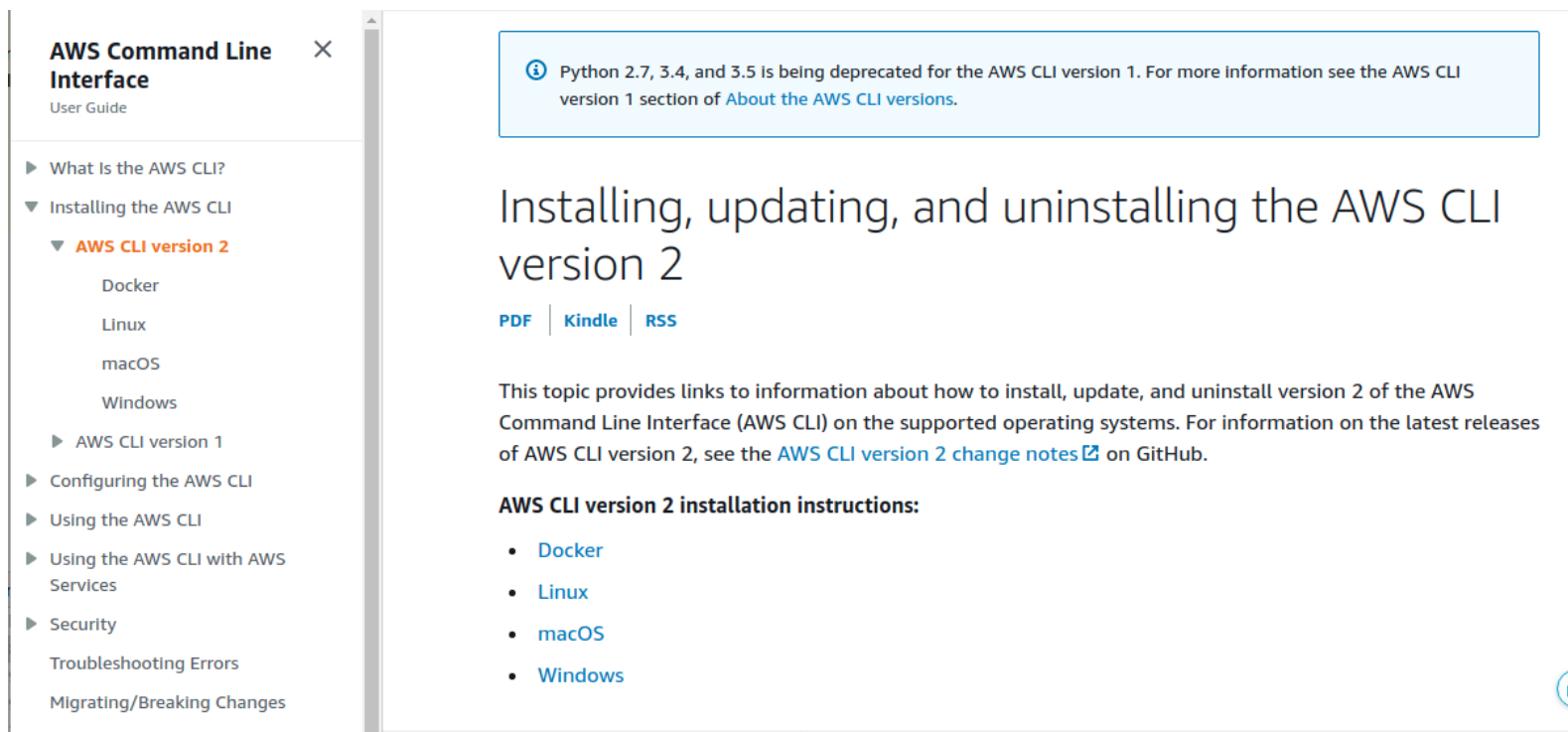


Download .csv

	User	Access key ID	Secret access key	Email login instructions
▶	✓ Dev	AKIA6KD2BMBQU5MYW3YK	NSQWeW51Rbch0W6QjaWz ATnjbGM0fW7Co6bme9vW <a href="#">Hide</a>	<a href="#">Send email</a>

# Step Three: Download and Install the AWS CLI

- ◆ If you don't already have it installed on your local machine, download the appropriate installer from:
  - [cli-download]  
([https://https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html))

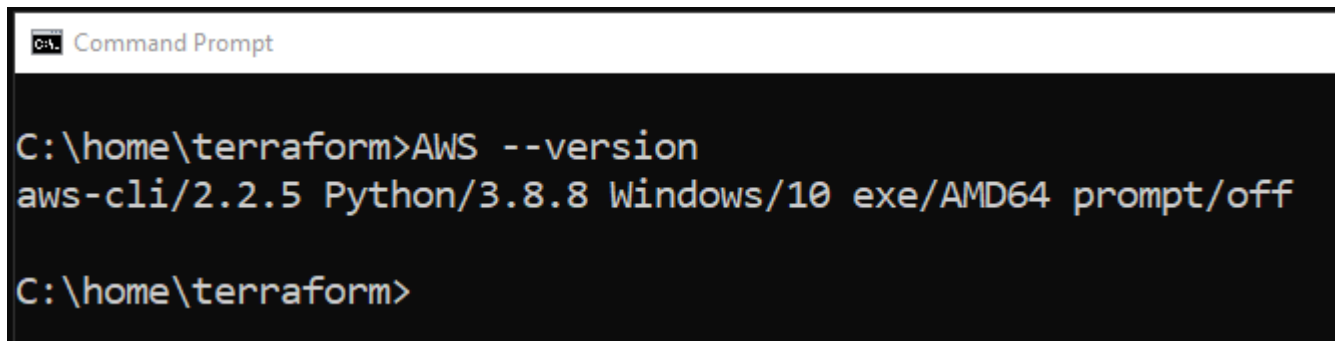


The screenshot shows the AWS Command Line Interface User Guide. The left sidebar contains a navigation menu with the following items: 'What Is the AWS CLI?', 'Installing the AWS CLI' (expanded), 'AWS CLI version 2' (selected), 'Docker', 'Linux', 'macOS', 'Windows', 'AWS CLI version 1', 'Configuring the AWS CLI', 'Using the AWS CLI', 'Using the AWS CLI with AWS Services', 'Security', 'Troubleshooting Errors', and 'Migrating/Breaking Changes'. The main content area features a blue information box at the top stating: 'Python 2.7, 3.4, and 3.5 is being deprecated for the AWS CLI version 1. For more information see the AWS CLI version 1 section of [About the AWS CLI versions](#).' Below this is the title 'Installing, updating, and uninstalling the AWS CLI version 2' with links for 'PDF', 'Kindle', and 'RSS'. The text explains that the topic provides links to information about how to install, update, and uninstall version 2 of the AWS Command Line Interface (AWS CLI) on supported operating systems. For the latest releases of AWS CLI version 2, it refers to the [AWS CLI version 2 change notes](#) on GitHub. Under the heading 'AWS CLI version 2 installation instructions:', there is a bulleted list of links: 'Docker', 'Linux', 'macOS', and 'Windows'.



## Step Three: Confirm the Installation

- ◆ Confirm the installation by using the command
  - “AWS --version”
- ◆ If your AWS CLI is installed correctly, then you should see something like this:



```
Command Prompt

C:\home\terraform>AWS --version
aws-cli/2.2.5 Python/3.8.8 Windows/10 exe/AMD64 prompt/off

C:\home\terraform>
```

## Step Four: Setting up the AWS CLI profile

- ◆ This step requires that you have the keys you downloaded in the .csv file
  - If you don't have them, or are using a supplied AWS ID, you will need to create new credentials
  - Creating credentials is covered in the next step
- ◆ Set up your profile using the command
  - “AWS configure --profile <name>”
  - You can leave the default region and output type set to [NONE]

	A	B	C	D
1	User name	Password	Access key ID	Secret access key
2	Dev		AKIA6KD2BMBQU5MYW3YK	NSQWeW51Rbch0W6QjaWzATnjvGM0fW7Co6bme9vW
3				
4				

dev\_credentials (+)

```
C:\home\terraform>aws configure --profile dev
AWS Access Key ID [None]: AKIA6KD2BMBQU5MYW3YK
AWS Secret Access Key [None]: NSQWeW51Rbch0W6QjaWzATnjvGM0fW7Co6bme9vW
Default region name [None]:
Default output format [None]:
```

## Step Four: Verify Credentials

- ◆ To ensure you set up your profile correctly, run a command to query your IAM profile
  - Use the command
    - *aws iam get-user --user-name <name> --profile <profile-name>*
  - Use the IAM name for the developer account you created for <name>
  - Use the profile name you created locally for <profile-name>

Command Prompt

```
C:\home\terraform>aws iam get-user --user-name dev --profile dev
{
  "User": {
    "Path": "/",
    "UserName": "Dev",
    "UserId": "AIDA6KD2BMBQ4PBXHLECD",
    "Arn": "arn:aws:iam::983803453537:user/Dev",
    "CreateDate": "2021-05-18T17:30:58+00:00",
    "PasswordLastUsed": "2021-05-18T17:37:19+00:00"
  }
}
```

# Step Five: OH NO! I LOST MY CREDENTIALS

- ◆ Your CLI credentials can be replaced at any time
  - This requires console access and IAM permissions
  - You can do this either as the developer user or the admin user
- ◆ You should change your credentials if you suspect they are no longer secret



# Step Five: Find the Credentials

- ◆ Login to the console
- ◆ Go to the IAM service and select the user whose credentials are to be changed
- ◆ Open up the "Security credentials" tab

Users > Dev

## Summary

[Delete user](#) [?](#)

**User ARN** arn:aws:iam::983803453537:user/Dev [🔗](#)

**Path** /

**Creation time** 2021-05-18 13:30 EDT

[Permissions](#) [Groups](#) [Tags](#) **[Security credentials](#)** [Access Advisor](#)

### Sign-in credentials

**Summary** • Console sign-in link: <https://roddavison.signin.aws.amazon.com/console> [🔗](#)

**Console password** Enabled (last signed in Today) | [Manage](#)

**Assigned MFA device** Not assigned | [Manage](#)

**Signing certificates** None [🔗](#)

### Access keys

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.

**If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.** [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status	
AKIA6KD2BMBQU5MYW3YK	2021-05-18 13:30 EDT	2021-05-18 19:09 EDT with iam in ...	Active   <a href="#">Make inactive</a>	<a href="#">✕</a>

# Step Five: Invalidate Credentials

- ◆ In the access key section, select the "Make inactive" option



✓ Access key AKIA6KD2BMBQU5MYW3YK deactivated ✕

Create access key

Access key ID	Created	Last used	Status	
AKIA6KD2BMBQU5MYW3YK	2021-05-18 13:30 EDT	2021-05-18 19:09 EDT with iam in ...	Inactive   <a href="#">Make active</a>	✕

## Step Five: Delete the Keys

- ◆ Click on the black x to delete the deactivated keys

 Access key AKIA6KD2BMBQU5MYW3YK deleted 

Create access key

Access key ID	Created	Last used	Status	
No results				

# Step Five: Generate New Keys

- ◆ Click on the "Create access key" button
- ◆ Download the \*.csv file and go back and use these to set your AWS CLI credentials

Create access key

✕

✓

Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Download .csv file

Access key ID	Secret access key
AKIA6KD2BMBQUHDI7BK3	***** <a href="#">Show</a>

Close



# Step Six: Install Terraform

- ◆ Go to the terraform download site
  - `https://www.terraform.io/downloads.html`
  - Download the correct binary archive for your local system
  - Extract the archive and place the binary on your local path

## Downloads

- [Download Terraform](#)
- [Debian/Ubuntu APT Packages](#)
- [RHEL/Fedora Yum Packages](#)
- [Upgrade Guides](#)

## Other Docs

- [Intro to Terraform](#)
- [Terraform Language](#)
- [Terraform CLI](#)
- [Terraform Cloud](#)
- [Terraform Enterprise](#)
- [Provider Documentation](#)
- [Terraform Glossary](#)
- [Publishing Providers and Modules](#)
- [Extending Terraform](#)

## Download Terraform

[JUMP TO SECTION](#) ▼

Below are the available downloads for the latest version of Terraform (0.15.3). Please download the proper package for your operating system and architecture.

Terraform is distributed as a single binary. Install Terraform by unzipping it and moving it to a directory included in your system's [PATH](#).

You can find the [SHA256 checksums](#) for Terraform 0.15.3 online and you can [verify the checksums signature file](#) which has been signed using HashiCorp's GPG key. You can also [download older versions of Terraform](#) from the releases service.

Check out the [v0.15.3 CHANGELOG](#) for information on the latest release.

**Note:** If you're upgrading from an older version of Terraform then there may be some extra notes or upgrade steps. Please refer to the [Upgrade Guides](#) to learn more.



**macOS**  
64-bit



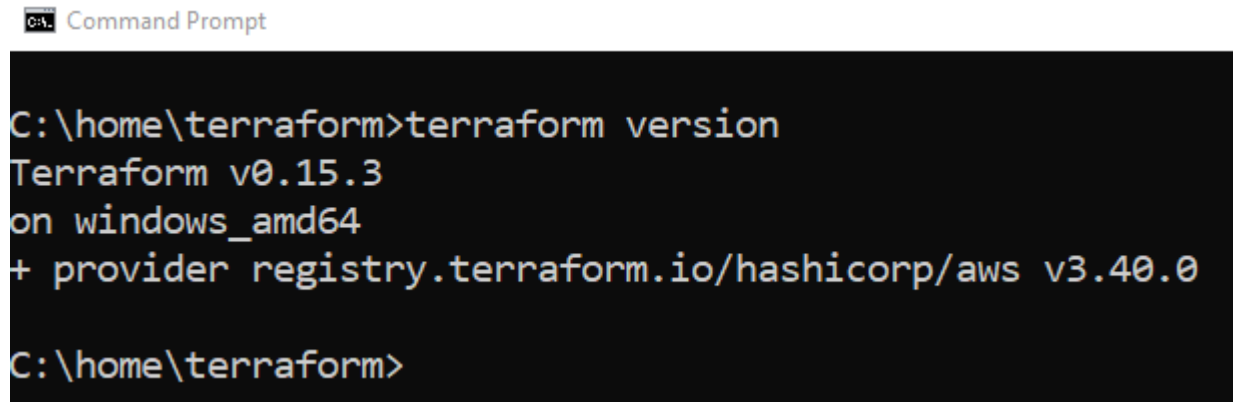
**FreeBSD**  
32-bit | 64-bit | Arm



**Linux**  
32-bit | 64-bit | Arm | Arm64

# Step Six: Test the Installation

- ◆ Run the command `terraform version` to ensure terraform is installed correctly



```
CA Command Prompt
C:\home\terraform>terraform version
Terraform v0.15.3
on windows_amd64
+ provider registry.terraform.io/hashicorp/aws v3.40.0
C:\home\terraform>
```

# Setup Complete

- ◆ You are now able to work with terraform and AWS