Terraform Setup

The Setup Process

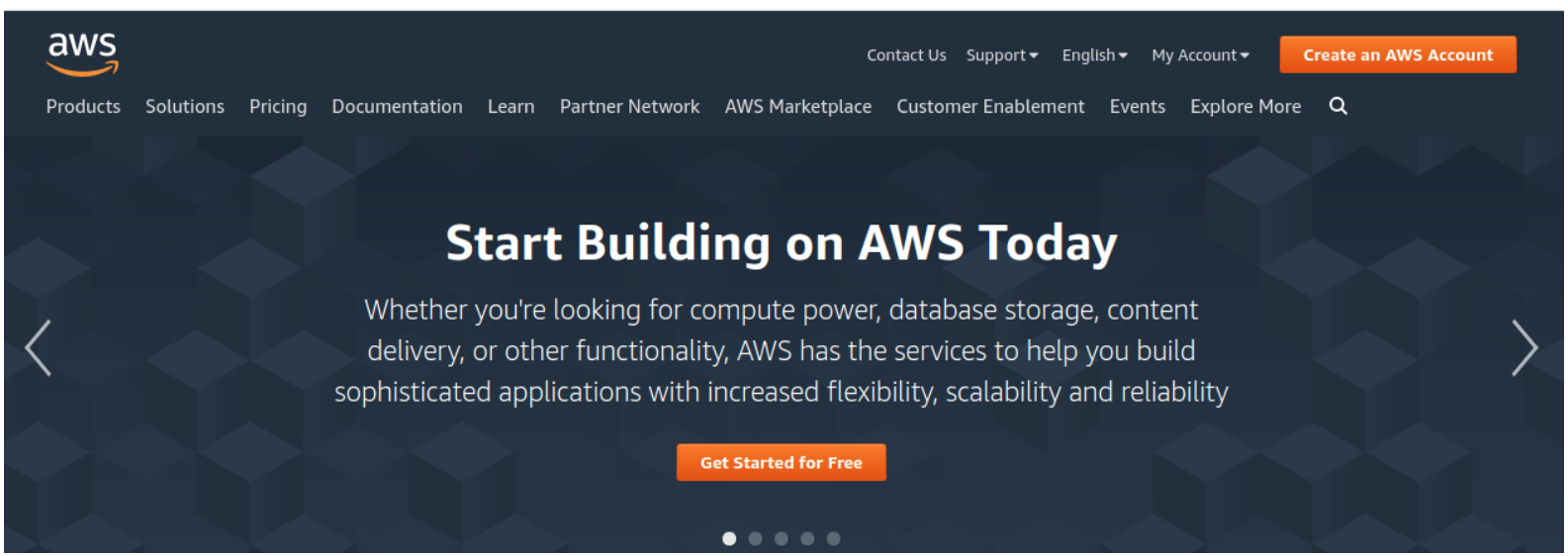
- This slide deck demonstrates the process in lab01-01
- Not all of the setup steps will apply to your configuration
- The steps of the setup process are:
  - Getting your AWS account or ID created or configured
  - Setting up an AWS IAM user for the class work
  - Installing the AWS CLI (command line 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

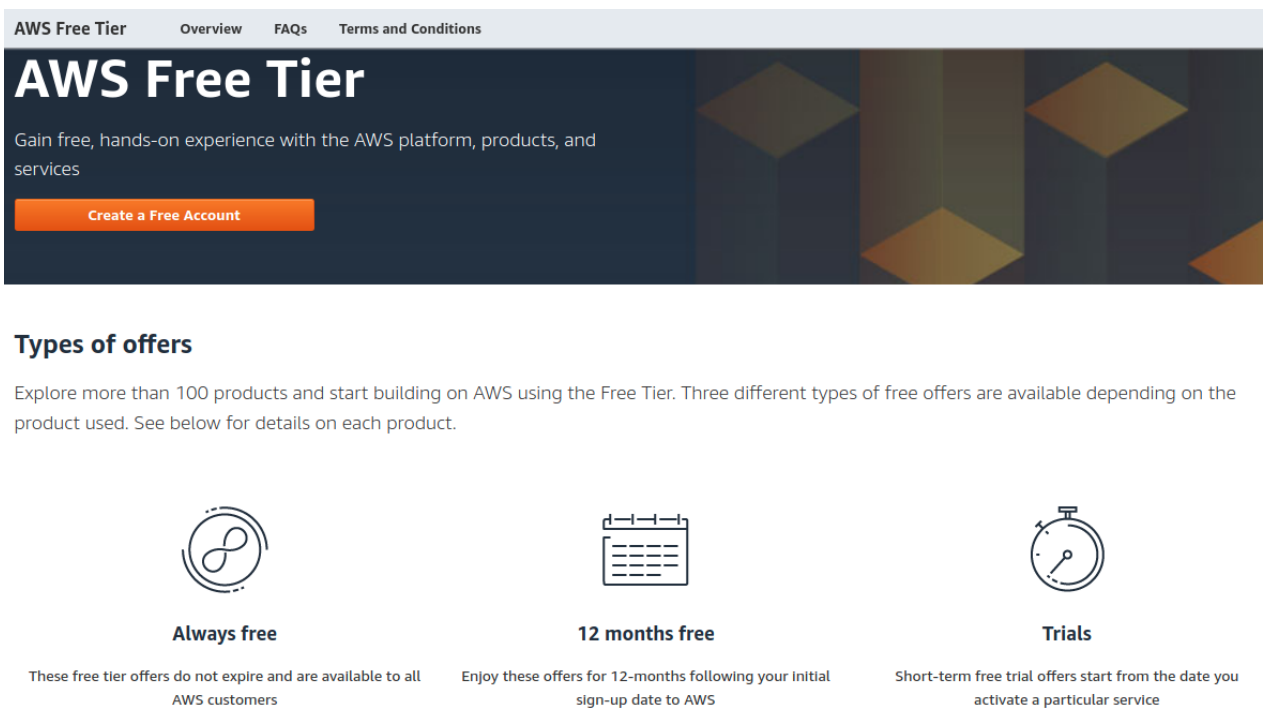
Step One: New Account

- 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



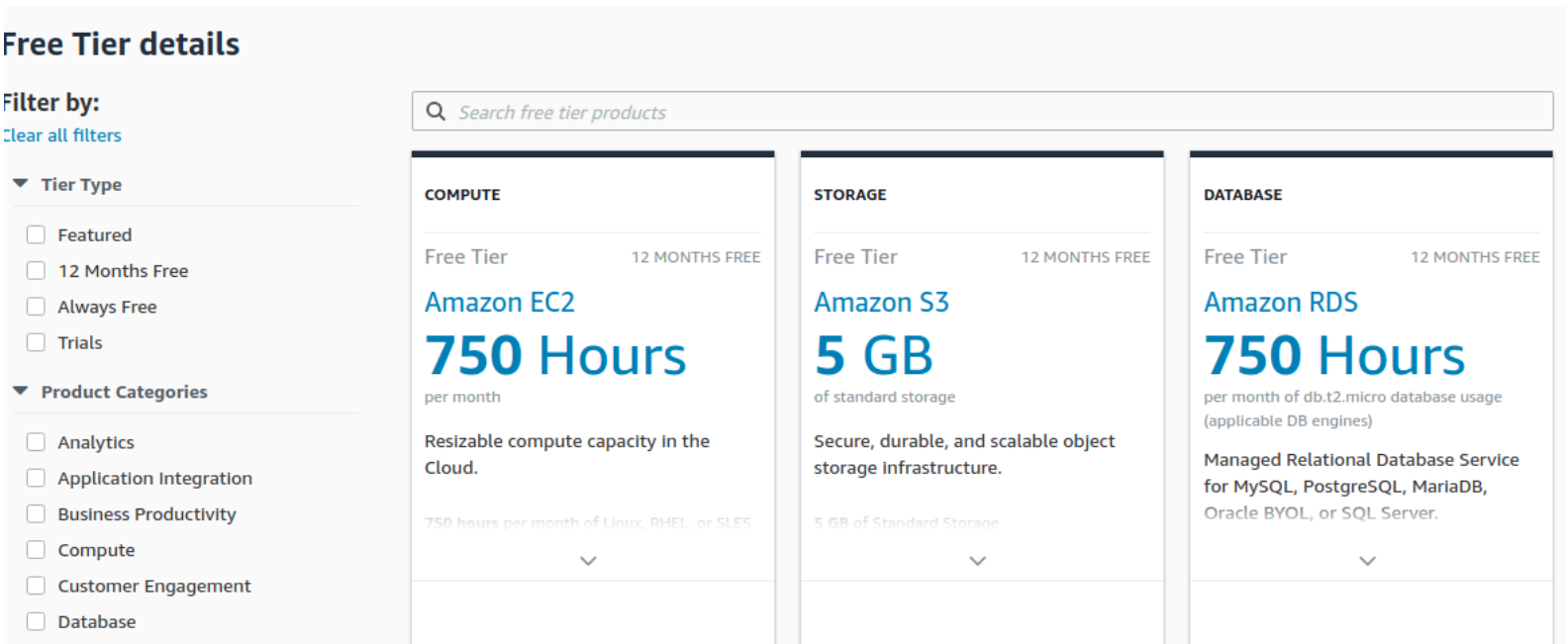
Step One: Free Tier Accounts

- 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**



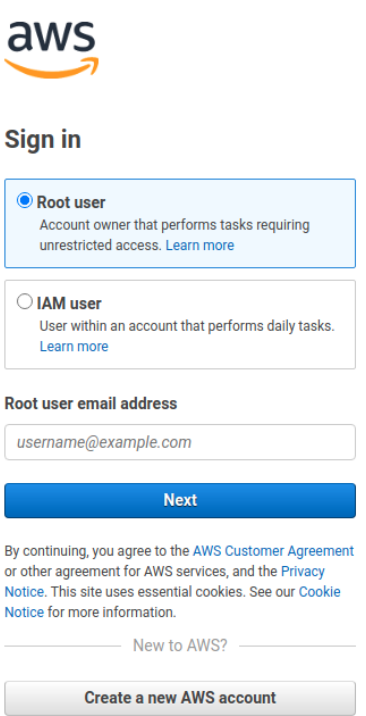
Step One: Free Tier Usage Rates

- 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



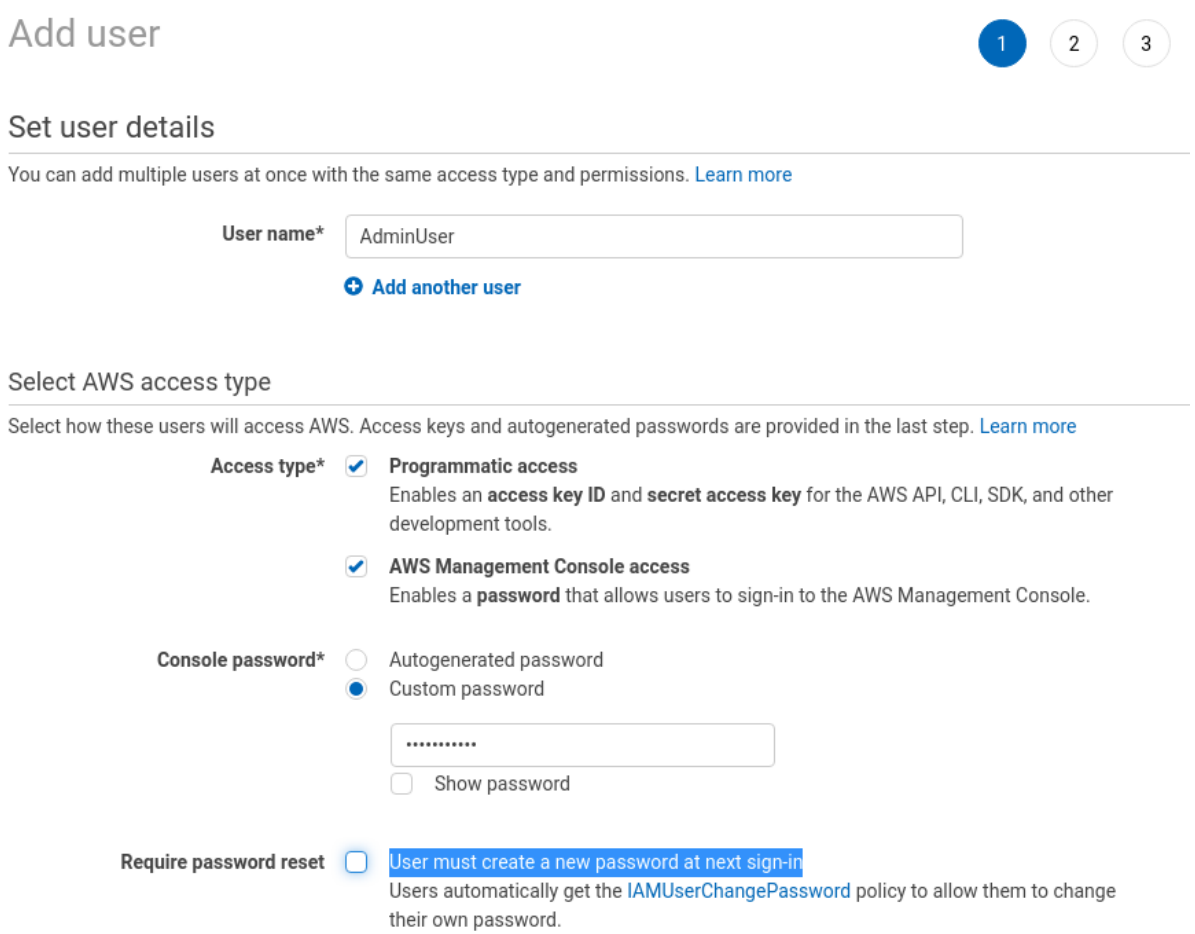
Step Two: Administrative User

- 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



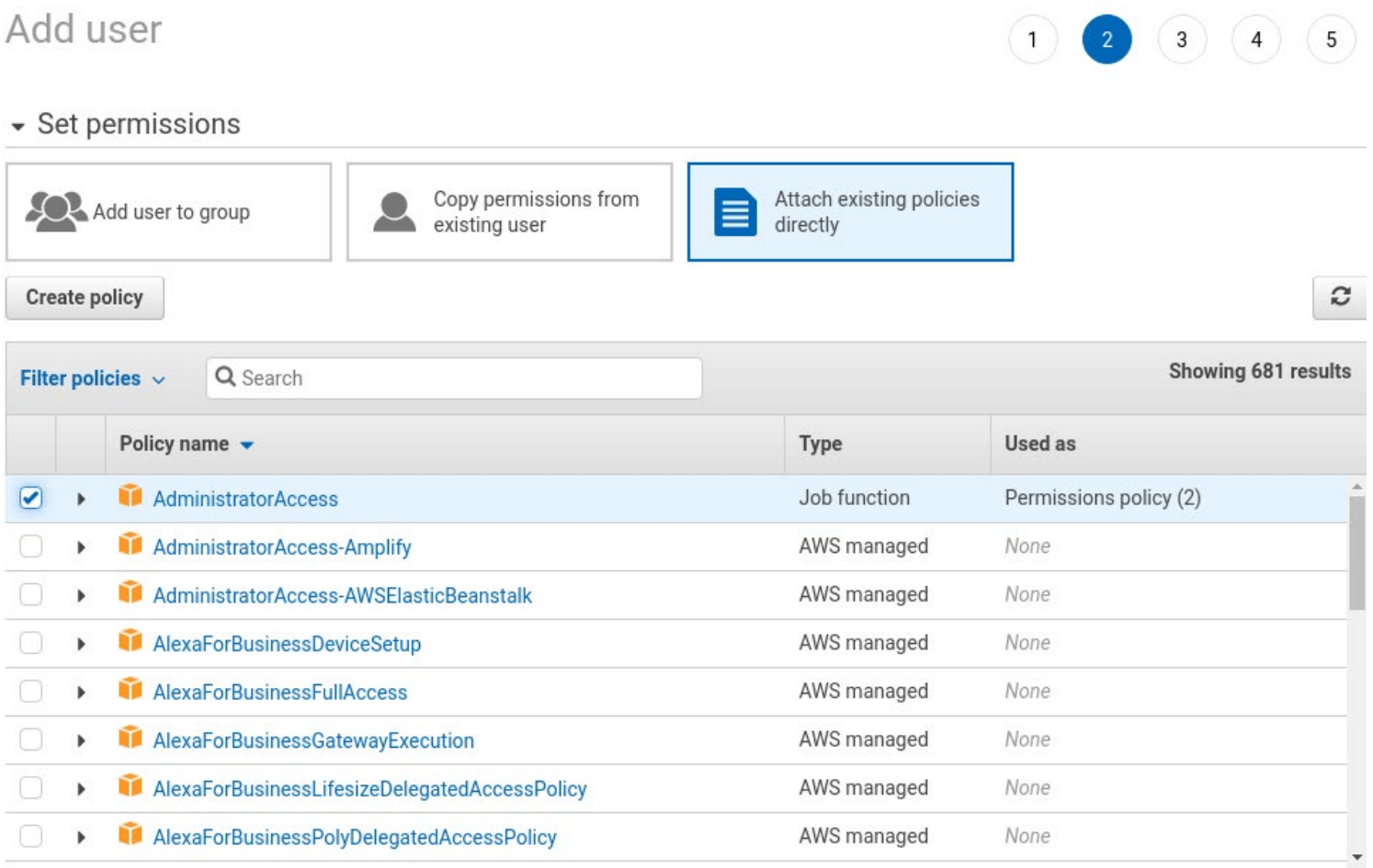
Step Two: Creating the User

- 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"



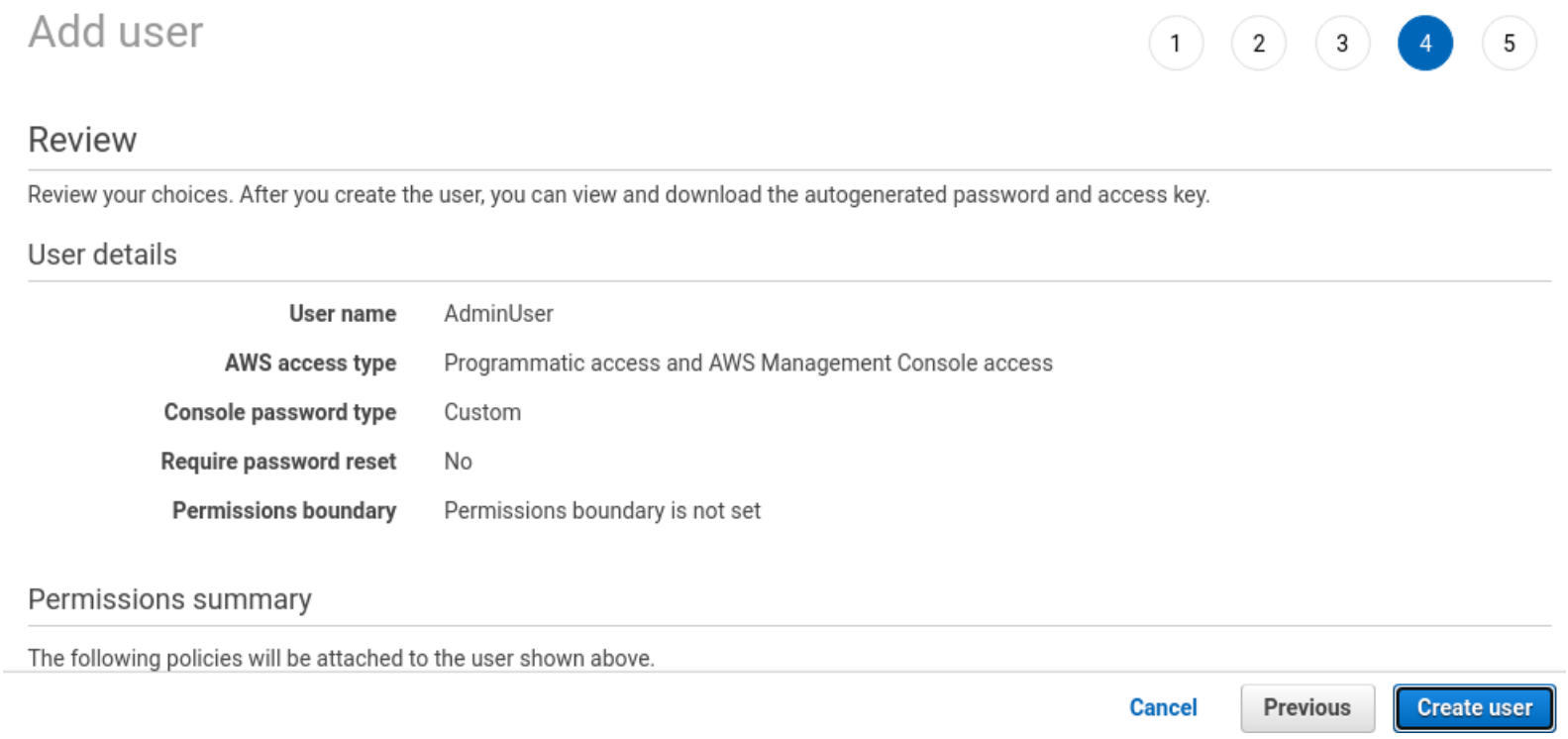
Step Two: Adding Permissions

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



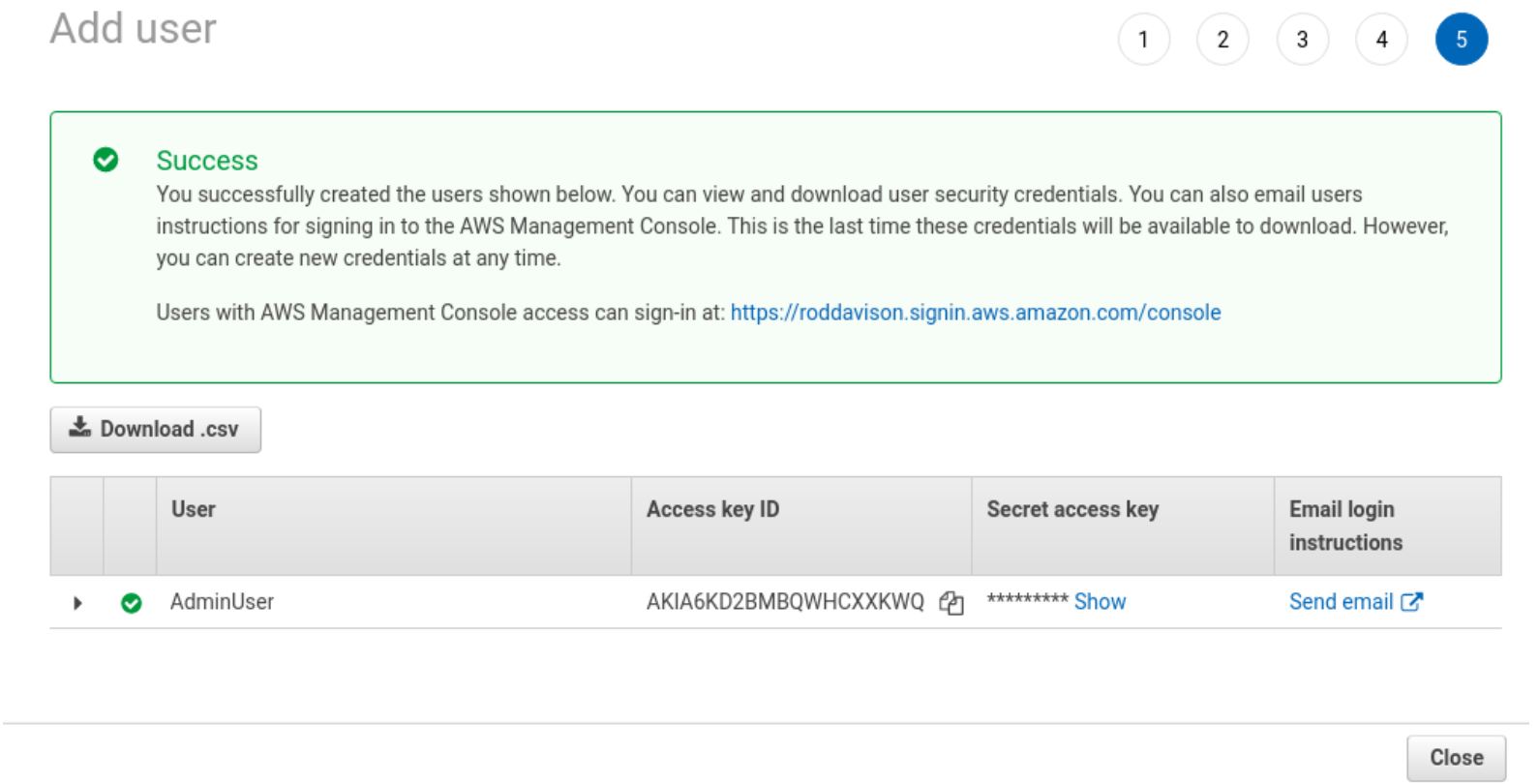
Step Two: Review the User

- 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"



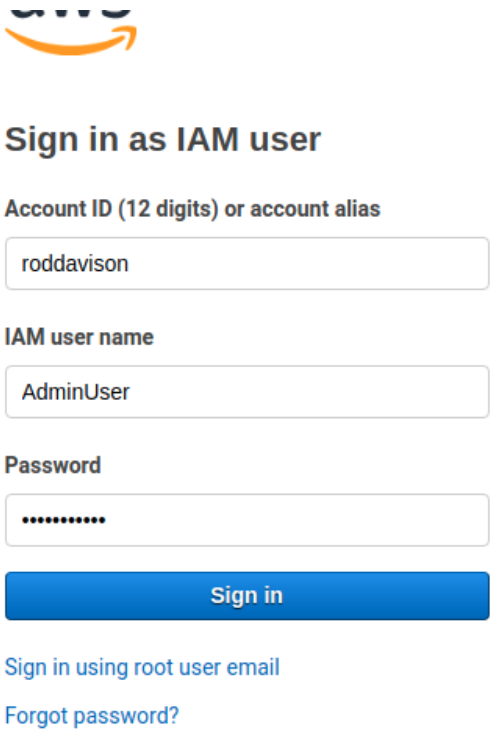
Step Two: Success Screen

- 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"



Step Two: Login as the Adminstrative User

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



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

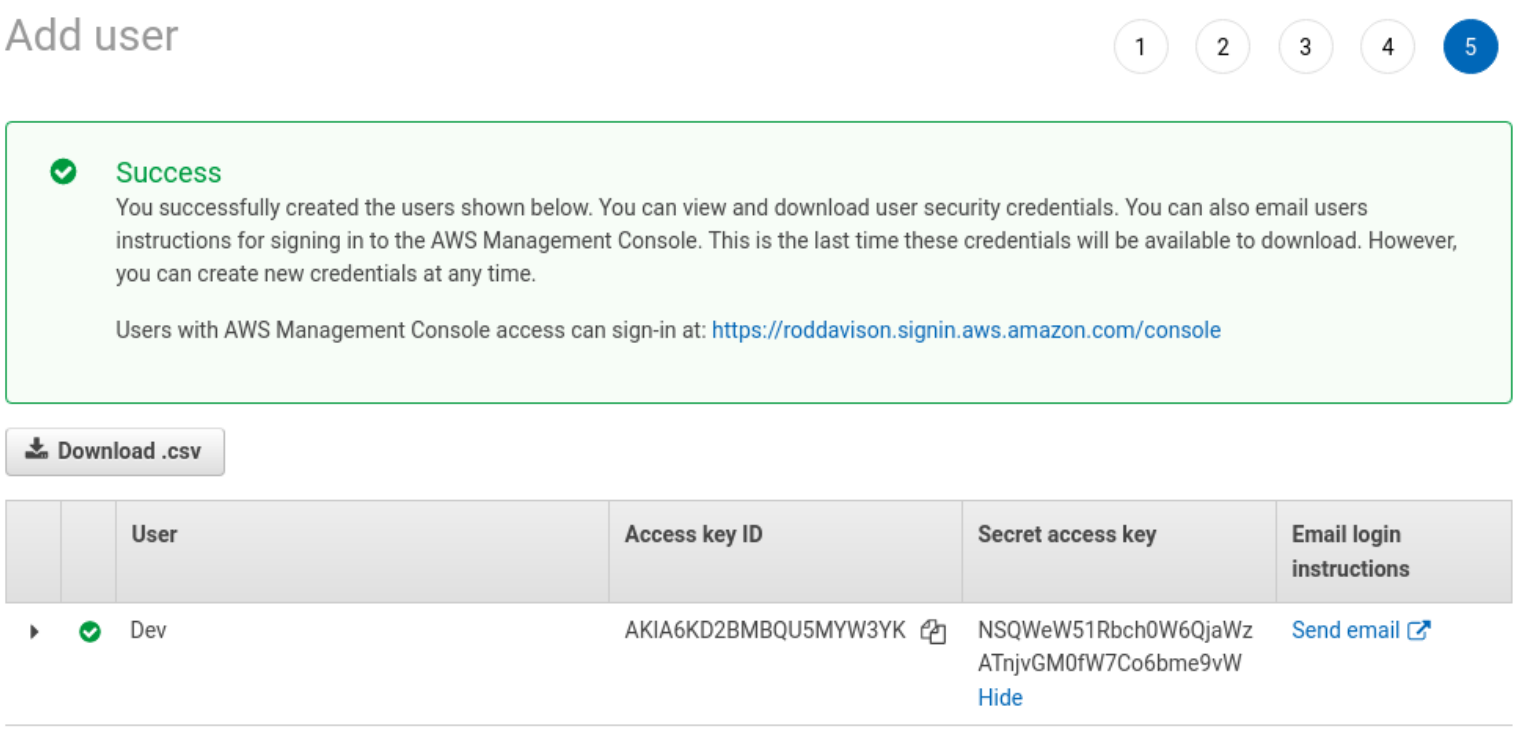
Permissions summary

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

Type	Name
Managed policy	AmazonS3FullAccess
Managed policy	AmazonEC2FullAccess
Managed policy	AmazonDynamoDBFullAccess
Managed policy	IAMFullAccess
Managed policy	CloudWatchFullAccess
Managed policy	AmazonRDSDataFullAccess

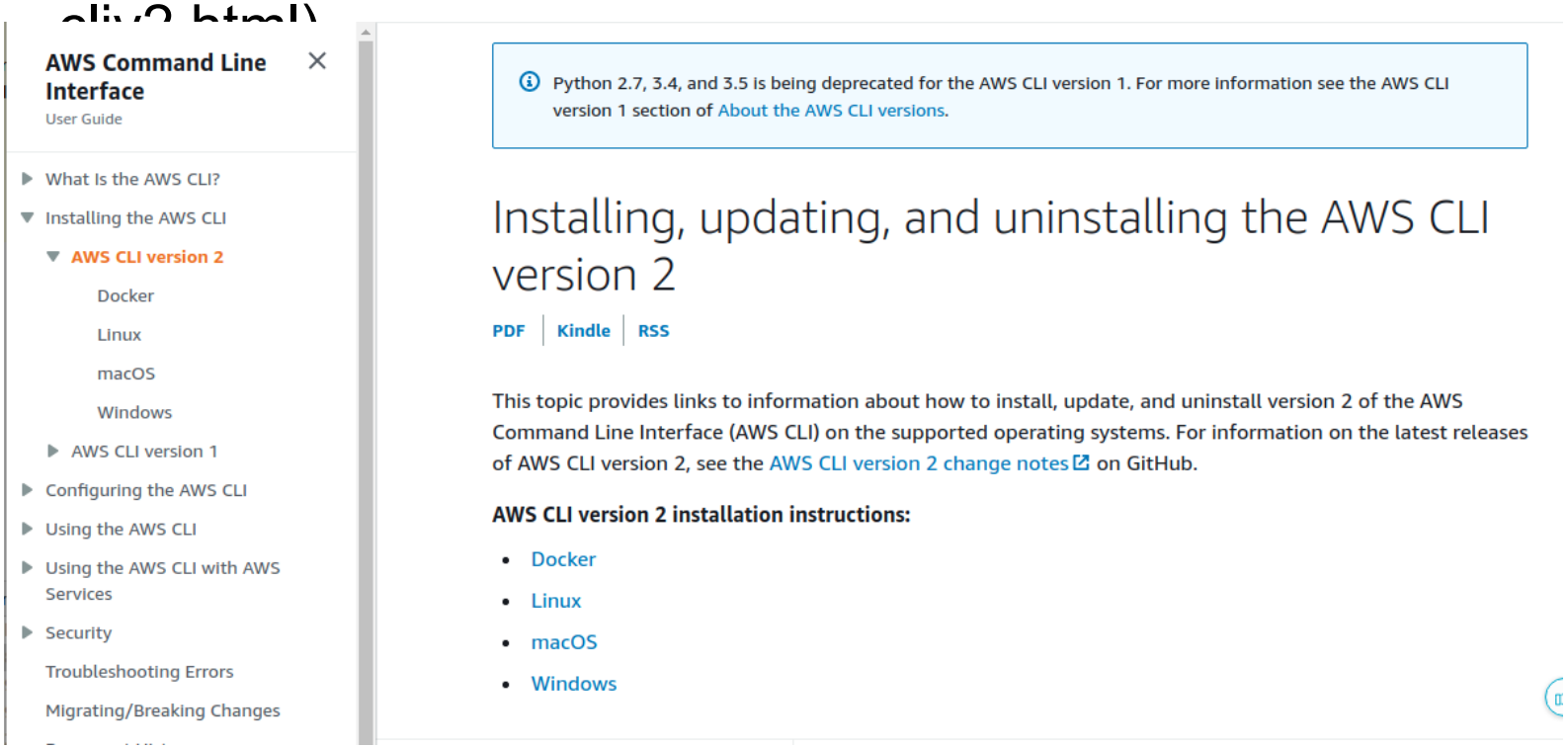
Step Two: Download Access Keys

- 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



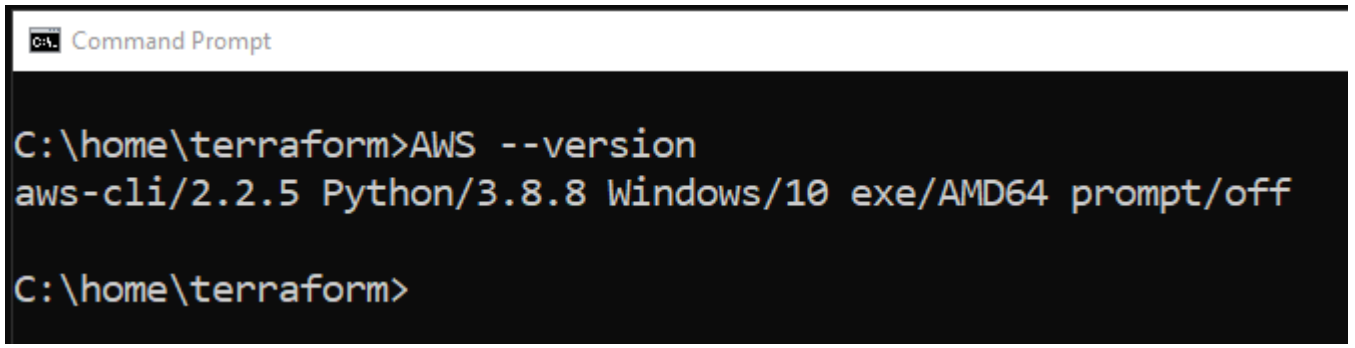
Step Three: Download and Install the AWS CLI

- 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)



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:



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]

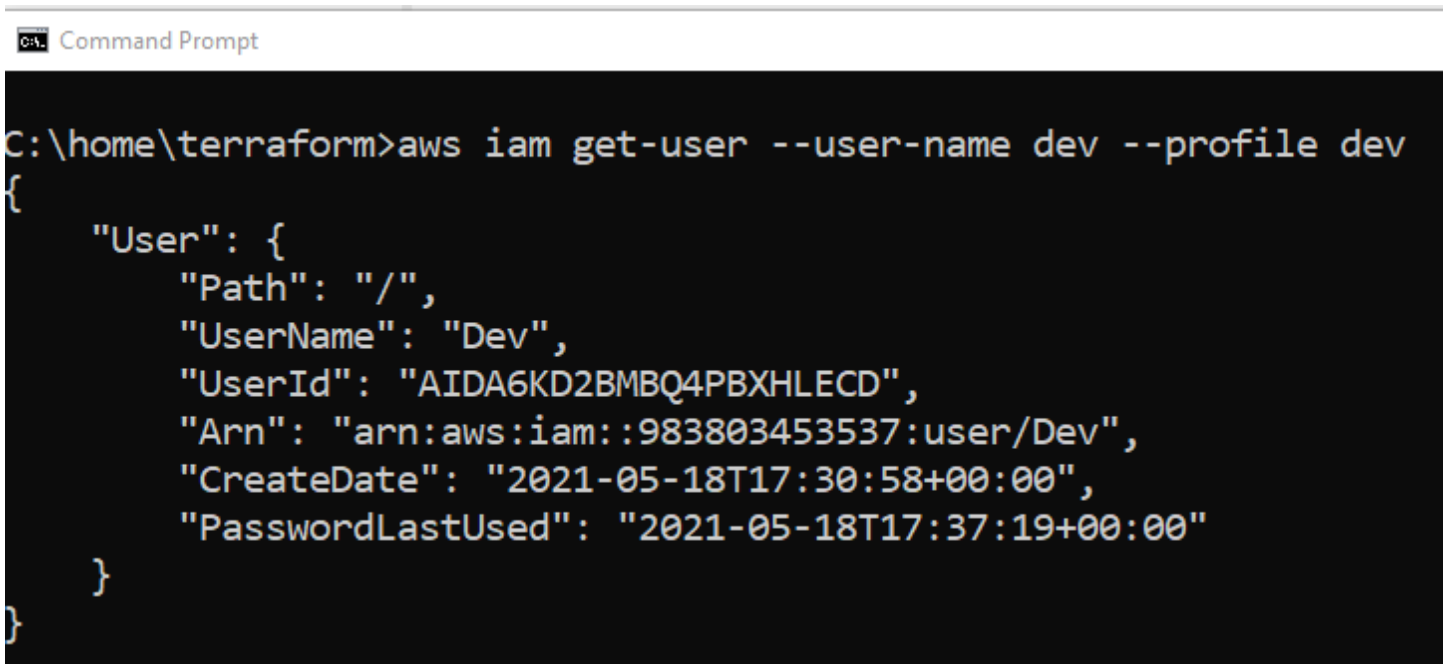
	A	B	C	D
1	User nam	Password	Access key ID	Secret access key
2	Dev		AKIA6KD2BMBQU5MYW3YK	NSQWeW51Rbch0W6QjaWzATnjvGM0fW7Co6bme9vW
3				
4				

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>*

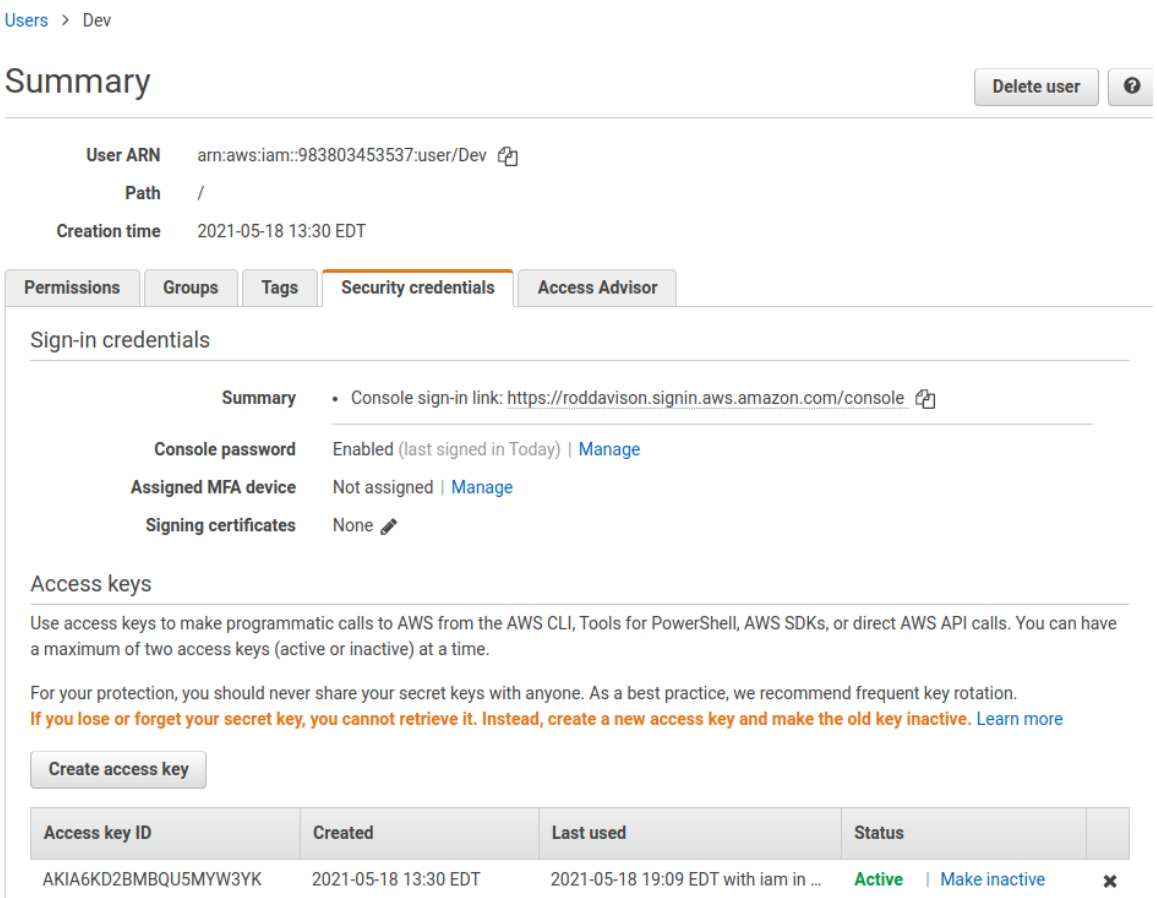


Step Five: OH NO! I LOST MY CREDENTIALS

- 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

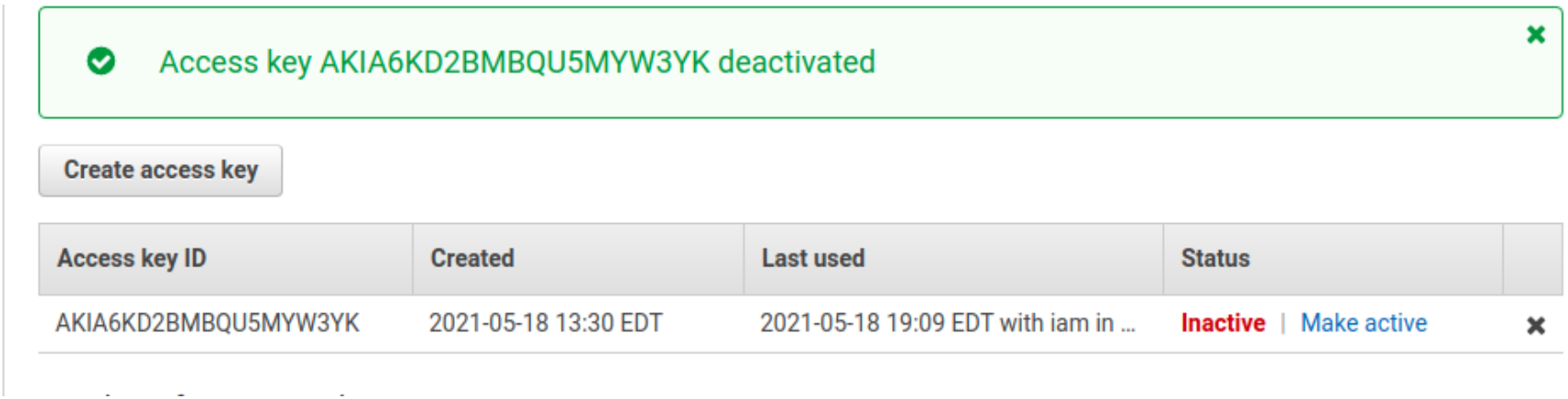
Step Five: Find the Credentials

- 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



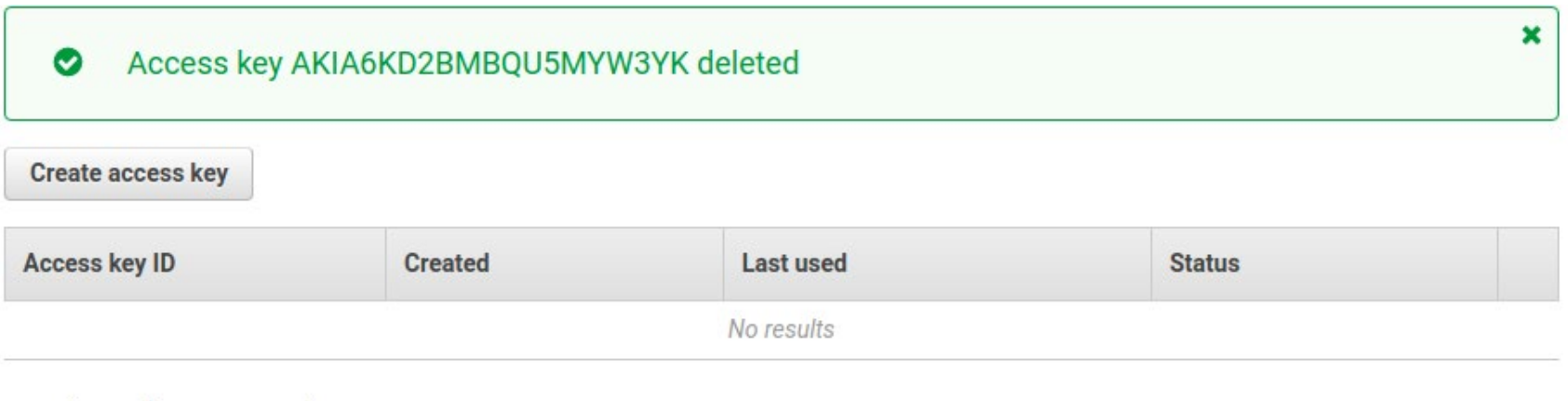
Step Five: Invalidate Credentials

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



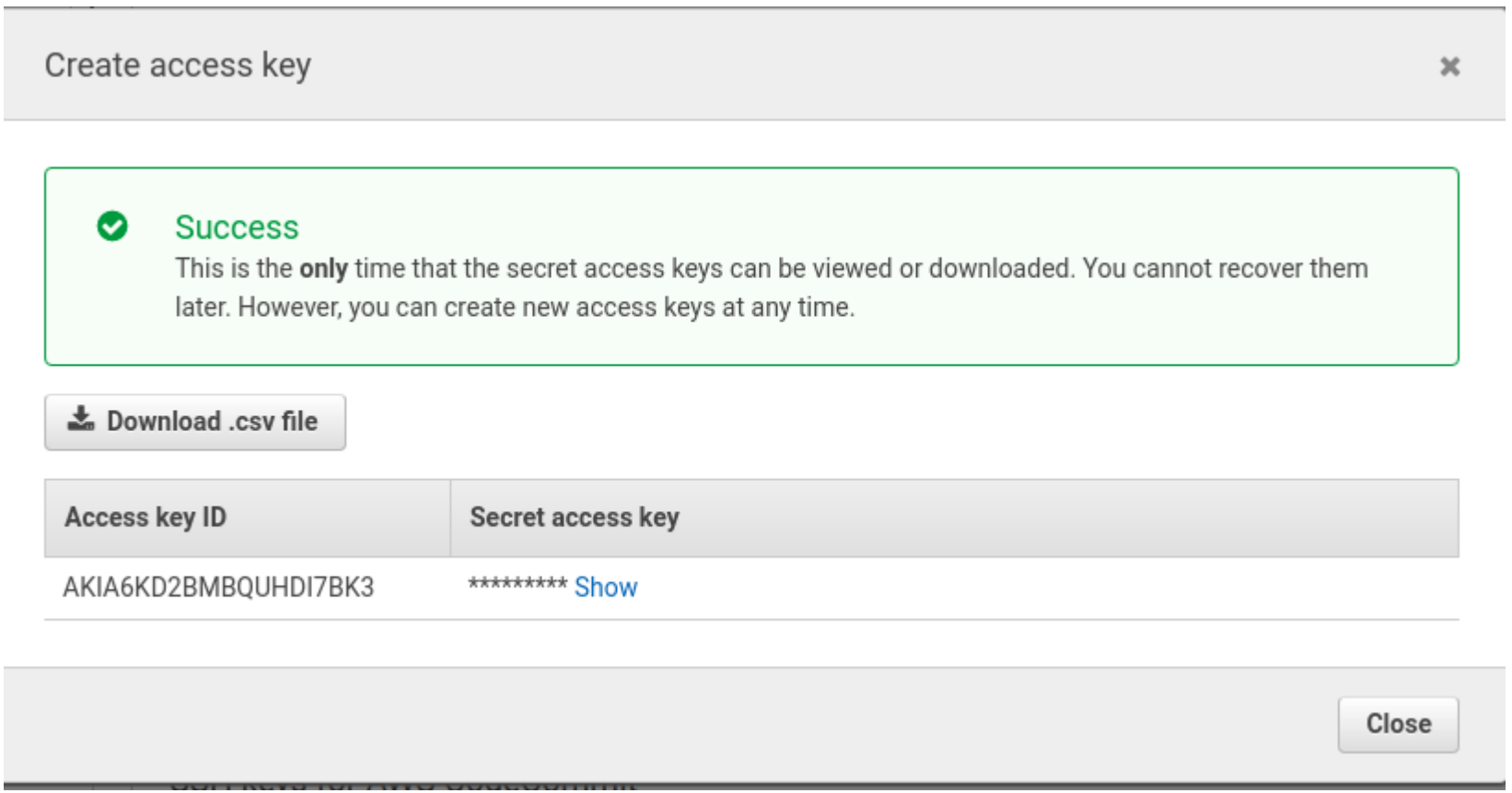
Step Five: Delete the Keys

- Click on the black x to delete the deactivated keys



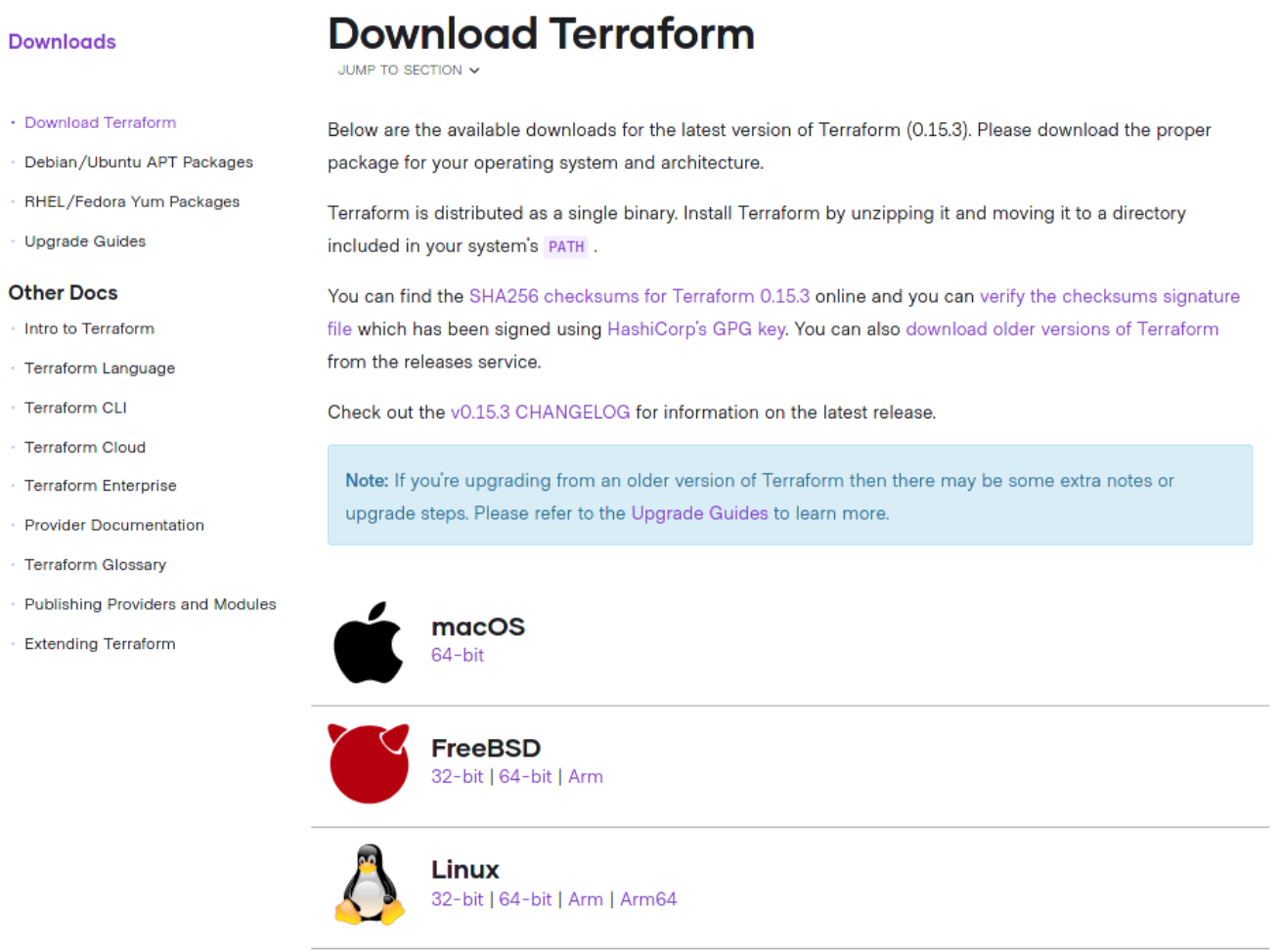
Step Five: Generate New Keys

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



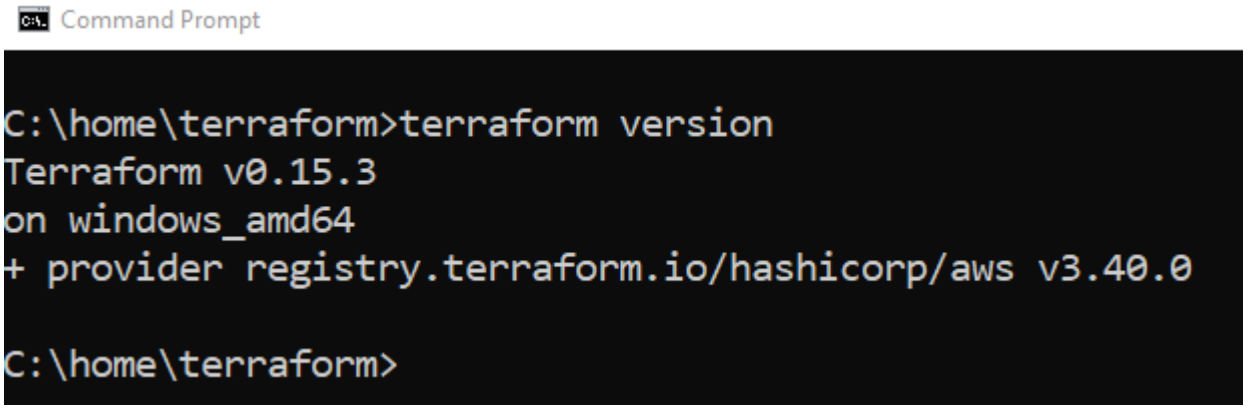
Step Six: Install Terraform

- 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



Step Six: Test the Installation

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



Setup Complete

- You are now able to work with terraform and AWS