

AWS Educate Setup

Dr Peadar Grant

September 15, 2021

Many of the labs in this course will use Amazon Web Services, or AWS. There are other cloud providers too - IBM, Google, Azure, others - this is not an endorsement of Amazon. Most of the ideas translate to the others. To do the labs in this course, you will need an AWS account.

Almost all AWS services are chargeable. Many services have a time-limited free tier.

AWS Educate has a number of resources including free credits so that labs exceeding the free tier don't incur charges. For more information on AWS Educate see:

<https://aws.amazon.com/education/awseducate/>

1 AWS Account types

AWS Educate used to support two types of accounts:

AWS starter account created by AWS, pre-loaded with credit and used only for education. It deletes automatically when the benefit ends. **This is the only type of account supported by AWS educate. Unfortunately it has some limitations!**

Regular AWS account where you create your own AWS account (which does require associating a valid credit card). *This account can't take advantage of AWS Educate credits. You can easily be charged for services you don't realise you've left running. I can't be liable if you go this route and get charged.*

Previously I used to recommend all students to get a regular AWS Account and apply AWS Educate credits to it, as it was the most flexible setup. Since December 2019, AWS Educate have removed the regular AWS account option for claiming credits. **An AWS Educate starter account is now the only option.**

2 AWS account setup

Sign up for AWS Educate at <https://www.awseducate.com/registration>.

Create a starter account using the instructions on the page.

3 Setup AWS command-line interface

The AWS Command Line Interface is a client program that runs on your local PC to allow you to manage AWS resources from the command-line (PowerShell, Bash). We will use the command-line extensively in this module.

3.1 PowerShell execution policy

PowerShell by default will not allow scripts to run that were downloaded from online. The following command will change this behaviour:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

3.2 AWS CLI installation

If you're on a lab computer OR if you already have the AWS CLI installed, then skip ahead to subsection 3.3.

Install the command-line tools from <https://aws.amazon.com/cli/>

3.3 Access key setup

In the folder:

```
C:\Users\yourusername
```

create a file config (no extension) with the following content:

```
[default]
region = us-east-1
```

(There is a script file setup-config-file.ps1 that can do this for you.)

Needs to be done EVERY time you log in on a student account! NOT the case for regular accounts!

Log in to AWS educate.

Go to <https://labs.vocareum.com/main/>.

Hit the *Account Details* button. Click *Show*. Copy and paste the contents into a file named EXACTLY

```
C:\Users\yourusername\.aws\credentials
```

(no .txt etc at the end).

To check that the AWS CLI is correctly configured, you can try running the command:

```
aws ec2 describe-instances
```

If it shows something similar to:

```
{
  "Reservations": []
}
```

then the AWS CLI is working OK.

4 Mac, Linux, UNIX users

Windows users can ignore this section.

Mac & *nix users will have no problems installing the AWS CLI, and the commands work identically to those on Windows.

The difference between Mac/Linux and Windows centres on the use of Bash/zsh by Mac vs PowerShell on Windows. The AWS CLI is perfectly scriptable using Bash, particularly in conjunction with jq to parse JSON. However, some of the scripts you will be provided with in this module will be PowerShell only due to time constraints.

The good news is that PowerShell 7 can be installed easily on a Mac with no issues. You *do not* need a Windows VM on your Mac to use any of the PowerShell or AWS commands in *this* course. Please go to the PowerShell page on GitHub for instructions.

When you have PowerShell installed, open the Terminal app and type pwsh and you'll be at a PowerShell prompt. Repeat the `aws ec2 describe-instances` command to confirm that the aws command is available in PowerShell.