Linux in AWS

Table of Contents

Background

Linux utilises a free, open-source operating system kernel. There are a wide variety of Linux distributions (operating systems using the Linux kernel) such as RedHat, Ubuntu and CentOS. Many Linux distributions are free.

Linux is very versatile. With heavy use of open-source code, Linux distributions can be modified to suit different use cases.

Many different vendors have been using Linux for many years for a variety of different products from things like your modem/router at home, to IoT devices.

These days we are used to using a keyboard and mouse (or similar) to use a Graphical User Interface (GUI) where we have icons, buttons, menus etc.

Before the GUI, there used to be a text interface where you only saw text on the screen. This text interface is what we call the Command Line Interface (CLI).

Network Attached Storage (NAS)

Network Attached Storage (NAS) devices are headless computers (no display monitor, keyboard or mouse) that connect to a network. You put one or more drives in it, and access the files stored on the NAS over the network.

NAS devices tend to run Linux. As there is no need for a normal Linux desktop GUI without a screen, it has been removed from the operating system. This frees up valuable resources such as CPU, memory, GPU etc. so that it can be optimised to provide fast performance for transferring files to/from the storage device across the network. You don't need as powerful a CPU etc. as you would have needed if the GUI was still there albeit not being accessed.

Instead of the Linux desktop, a Network Attached Storage device is accessing using an admin page in the web browser, much like when you would manage your modem/router at home.

You can also remotely access the CLI using SSH (Secure Shell Protocol).

Amazon Web Services

Elastic Compute 2 (EC2)

We use the Compute service EC2 to run VMs (Virtual Machines) in AWS.

With this you can run a variety of operating systems including Windows, Ubuntu and even Mac OS.

Windows licences cost money and the Windows desktop uses CPU, RAM etc. even when we are not accessing it via the Remote Desktop.

So, where there's not a need to run Windows software in the Cloud, almost invariably Linux is going to be chosen.

Amazon has its own Linux distribution Amazon Linux 2 available on free tier, and it is what we will be using when we use EC2 in this course.

You can remotely access the CLI on Amazon Linux 2 using SSH. To avoid confusion with the AWS CLI alternative to the Management Console, we will refer to the CLI on Amazon Linux 2 as the Linux shell from here on in this document.

Amazon Linux 2 is based on the CentOS distribution, so the commands will mostly be pretty similar to what you would use with CentOS.

Why Learn Linux commands?

When running a Virtual Machine in the Cloud you'll most likely be using Linux unless there's a very good reason not to, and you will need to enter commands to install software, configure the operating system etc. We don't want to spend money on more CPU, RAM etc. to run a Linux desktop that we won't need. Learning how things are done in the industry will prepare you better for working in it.

In this course we will use the AWS Management Console, the web page admin console to access Amazon Web Services. This is good for an introductory course, when learning new things for the first time, but an experienced Cloud worker will use the AWS CLI instead to create EC2 instances, stop EC2 instances etc.

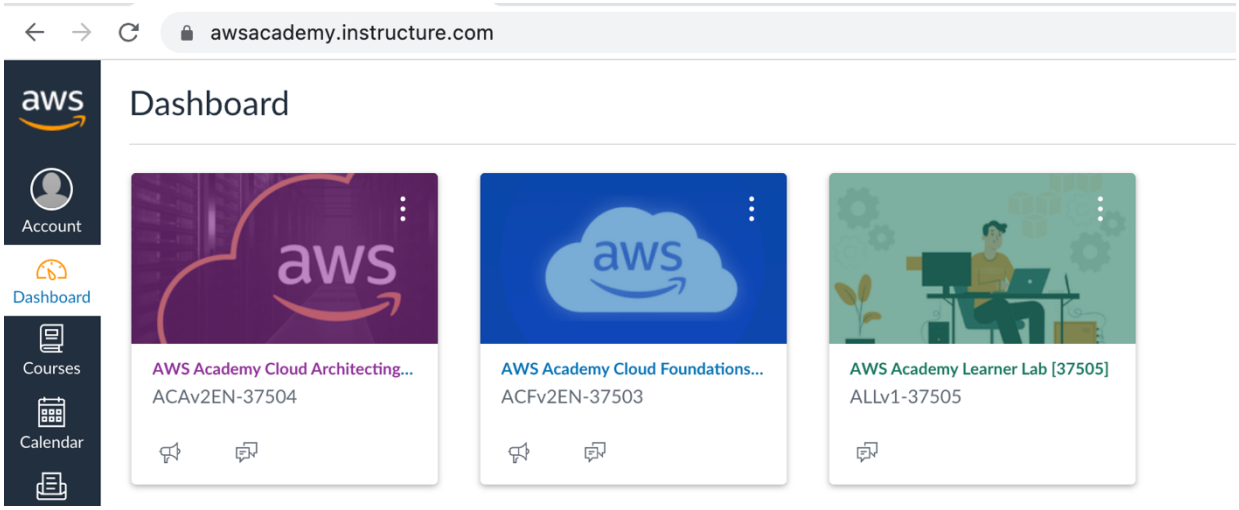
AWS has a well-documented CLI. AWS regularly changes the Management Console, and you may find that some lab instructions are out of date. Whereas the CLI will change much more slowly, and if it does change, you can quickly

identify what to do by looking at the documentation.

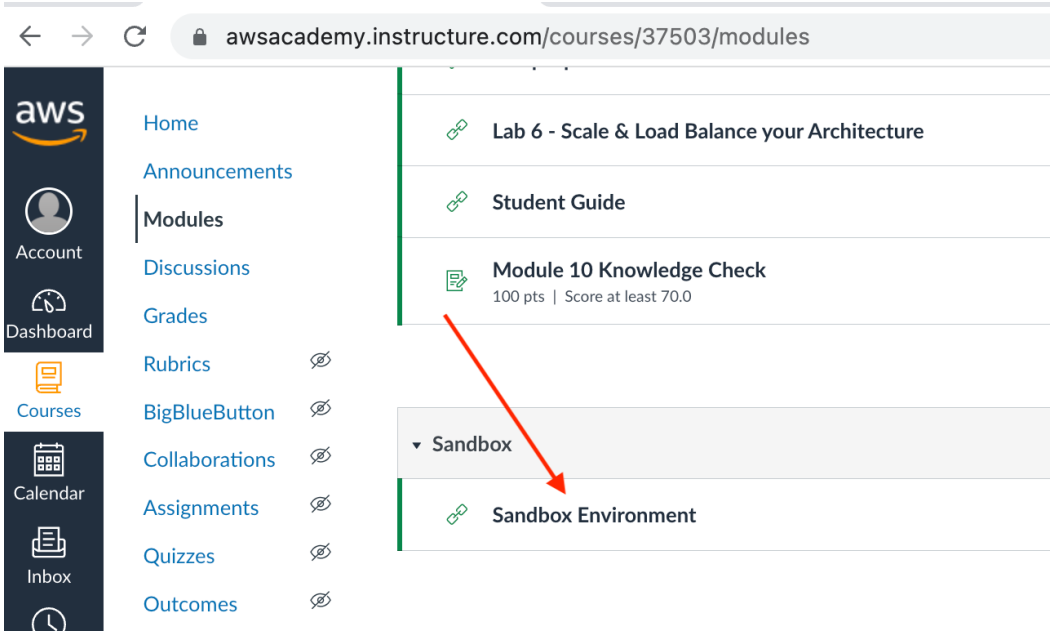
Amazon Web Services Academy Sandbox

When you open AWS Canvas you will see some Academy Courses and once activated a Learner Lab environment as well.

Note that the AWS Academy Course numbers change each semester, so if yours are different don't worry about that.



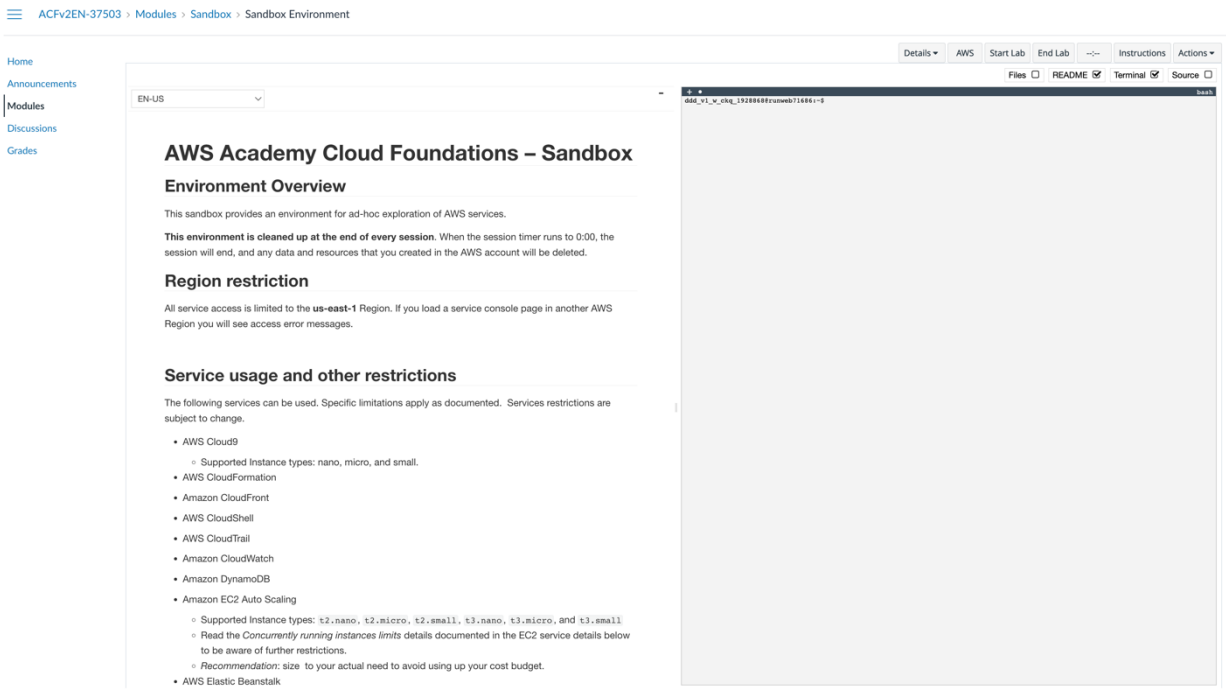
If you choose an Academy Course (except for the Learner Lab which is only used for Assignments), go to Modules and scroll down you will see a link to the Sandbox environment near the bottom of the page



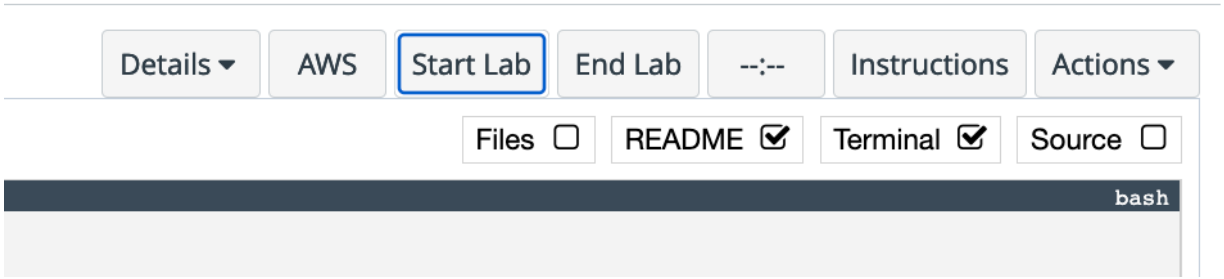
When you go to this link you will see the Sandbox environment.

The first time you will be prompted to accept Terms and Conditions. Read through the T&Cs and at the bottom of the page click on the Button I agree.

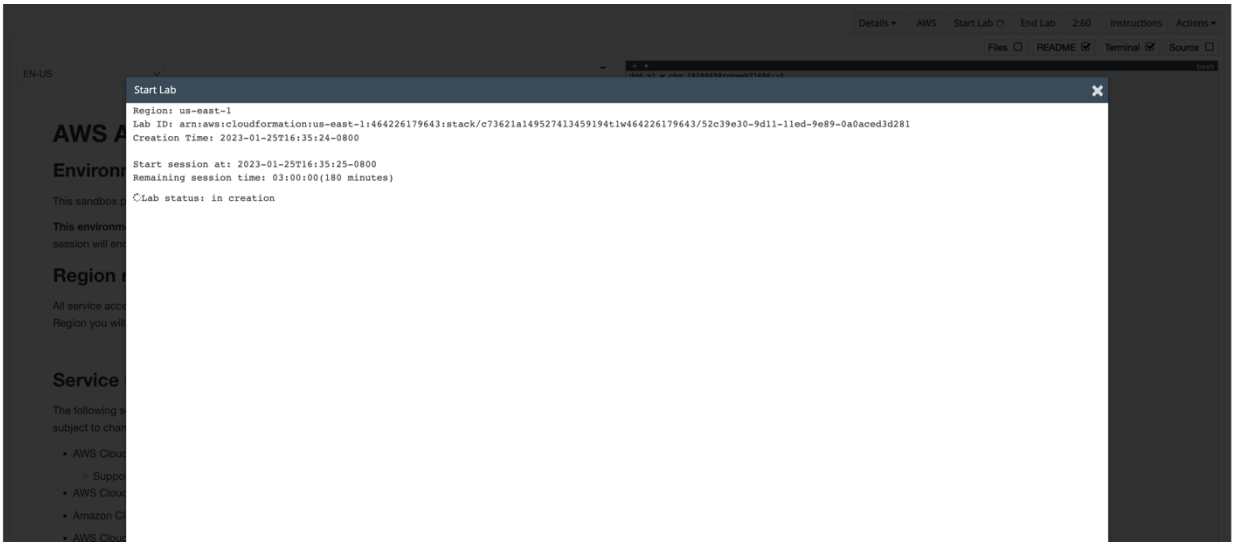
You should then see this screen. Have a read of the instructions on the left side of the screen. You will need to scroll down the page to read them all.



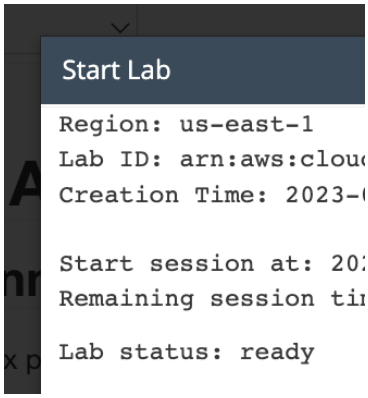
Click on Start Lab at the top right of the screen



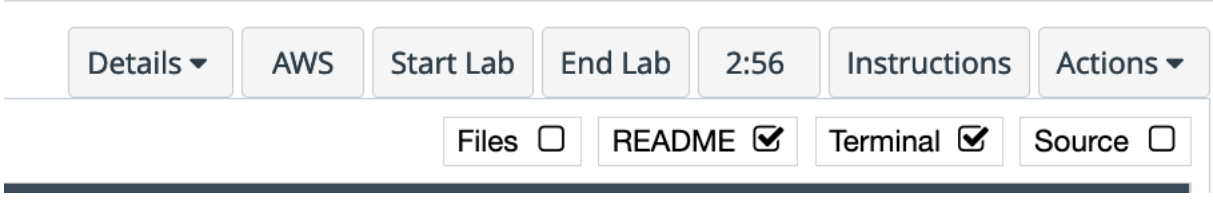
You will then see a screen as shown below indicating “lab status: in creation”. As the Sandbox creates minimal resources it will start much faster than some of the labs you will do as the semester progresses, but it may still take a few minutes



When it is done the Lab Status will change to ready



You can see that the timer shows I have 2 hours and 56 minutes left

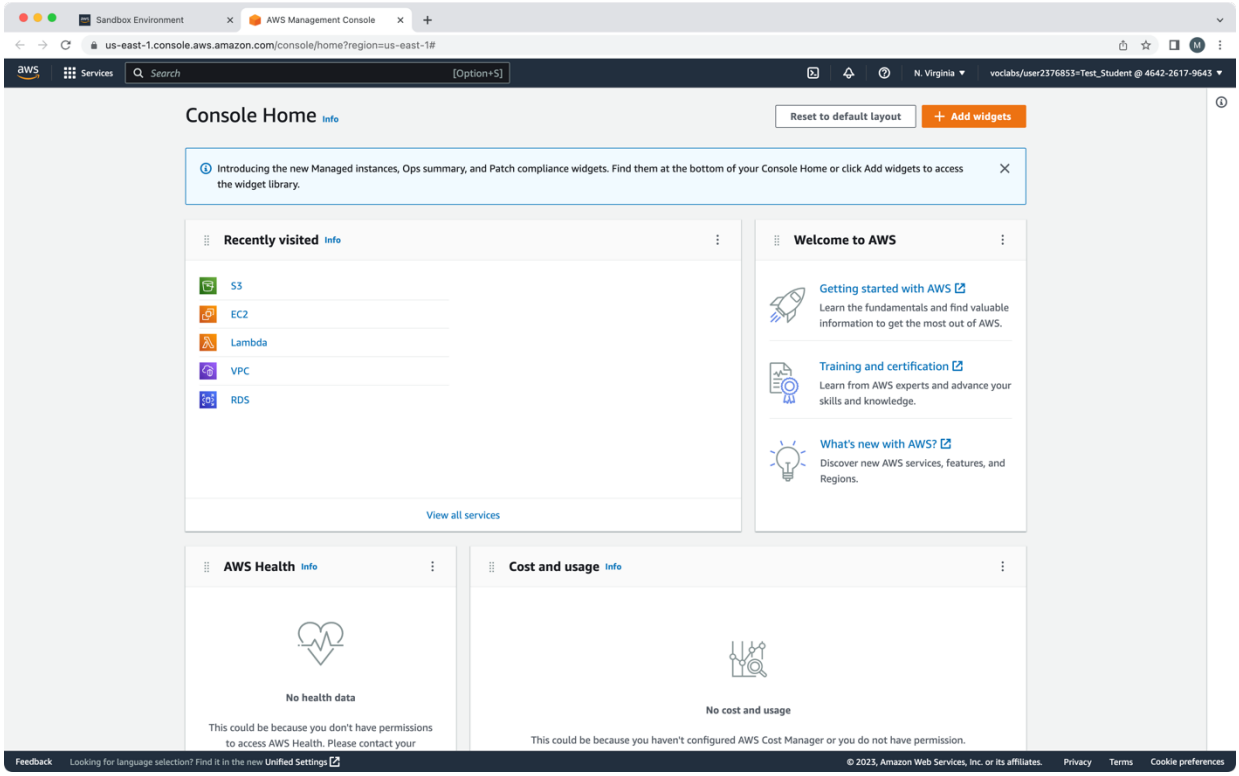


When this timer gets to 0 or if I hit End Lab, all the resources I have created in the Sandbox will be terminated and it will be cleaned up. The timer can be extended by clicking the Start Lab button again. You can also do this if you

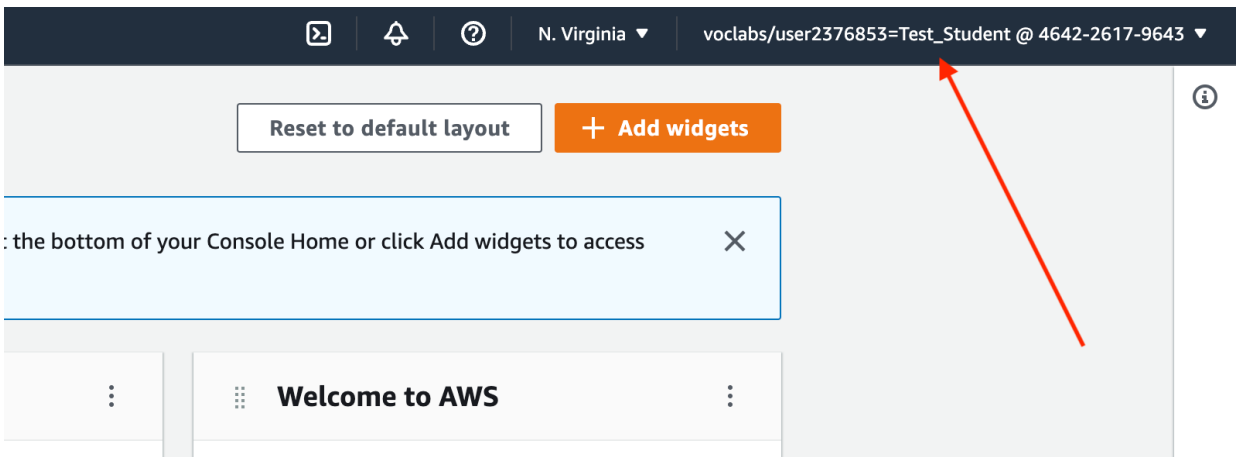
need more time to complete one of the labs or to demonstrate it to your tutor.

Open the AWS Management Console

Just to the left of the Start Lab button, there is an AWS button. Click on this to bring up the AWS Management Console in a new tab

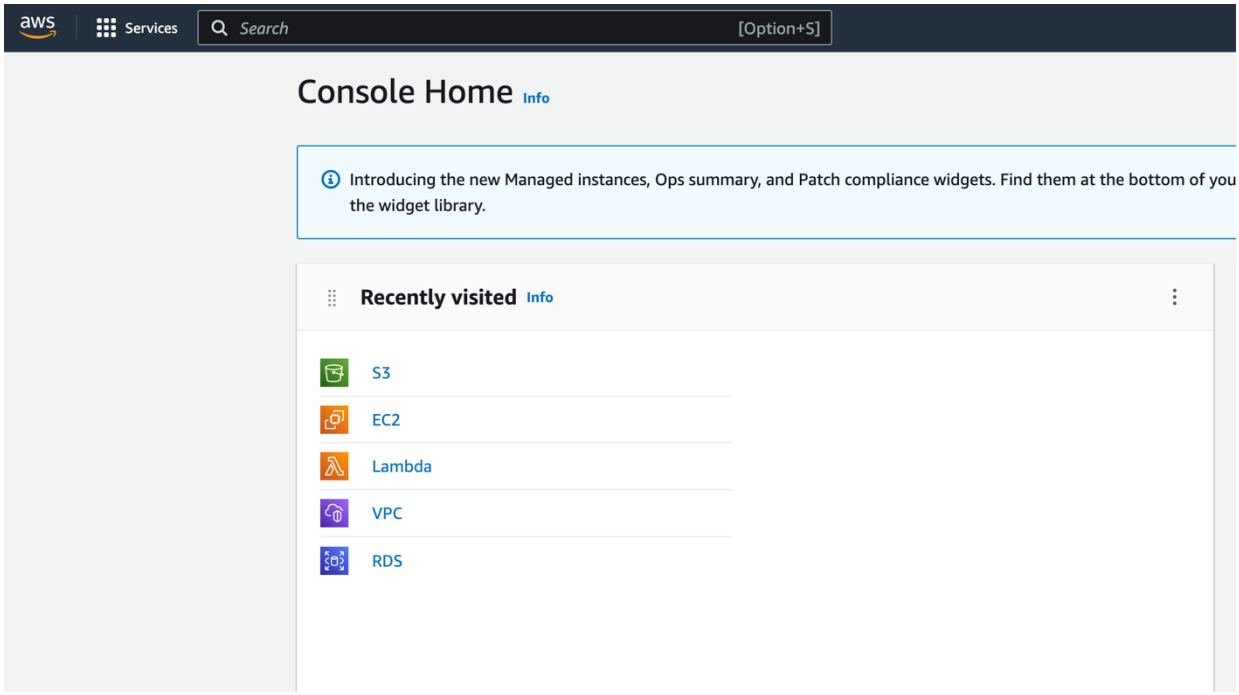


At the top right of the AWS Management Console is your account name. This will typically be your student email address, or it could be your name. This must be visible in all Lab Reports and Assignment Reports.

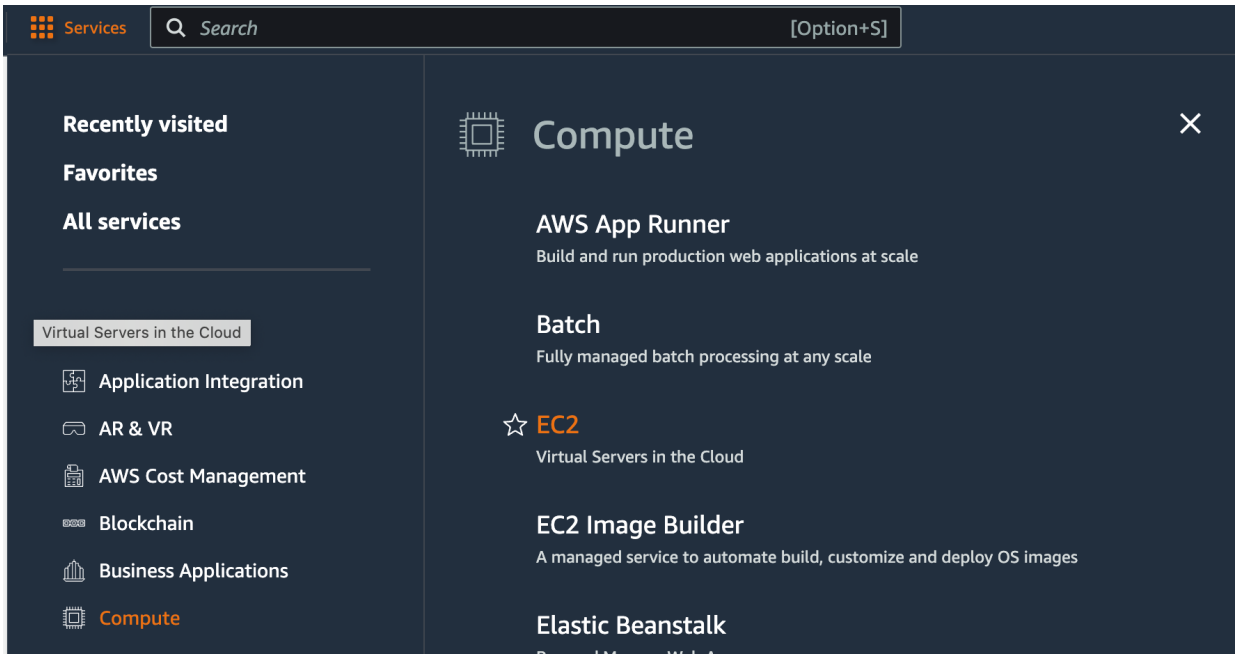


You can open the EC2 console in one of a number of ways

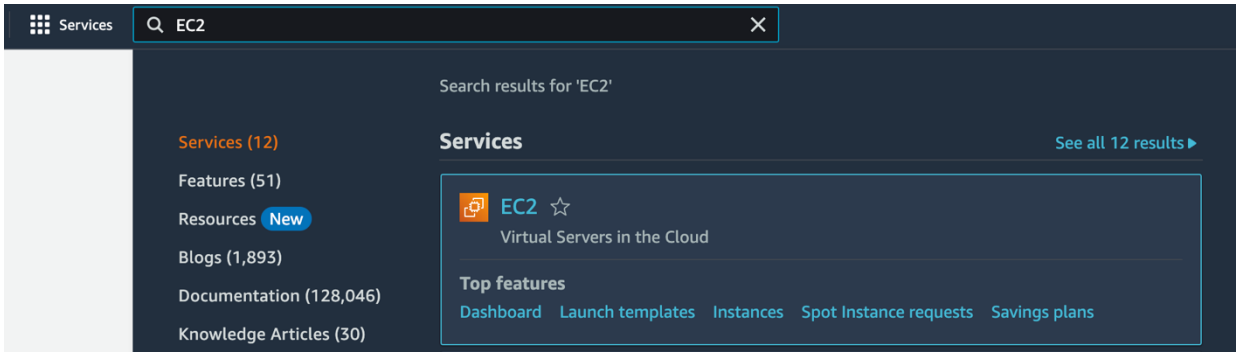
It may show under Recently visited



Or you can click Services at the top left, Recall that EC2 is a Compute Service, so select Compute and EC2



Another way is to type EC2 in the Search box and then select EC2.

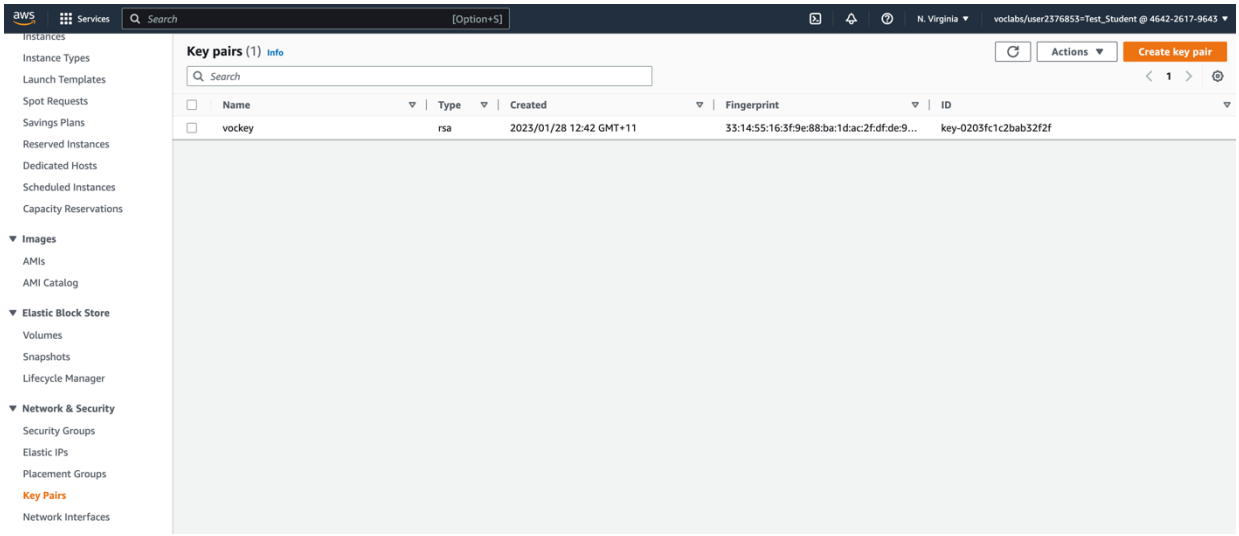


Use whichever method works best for you.

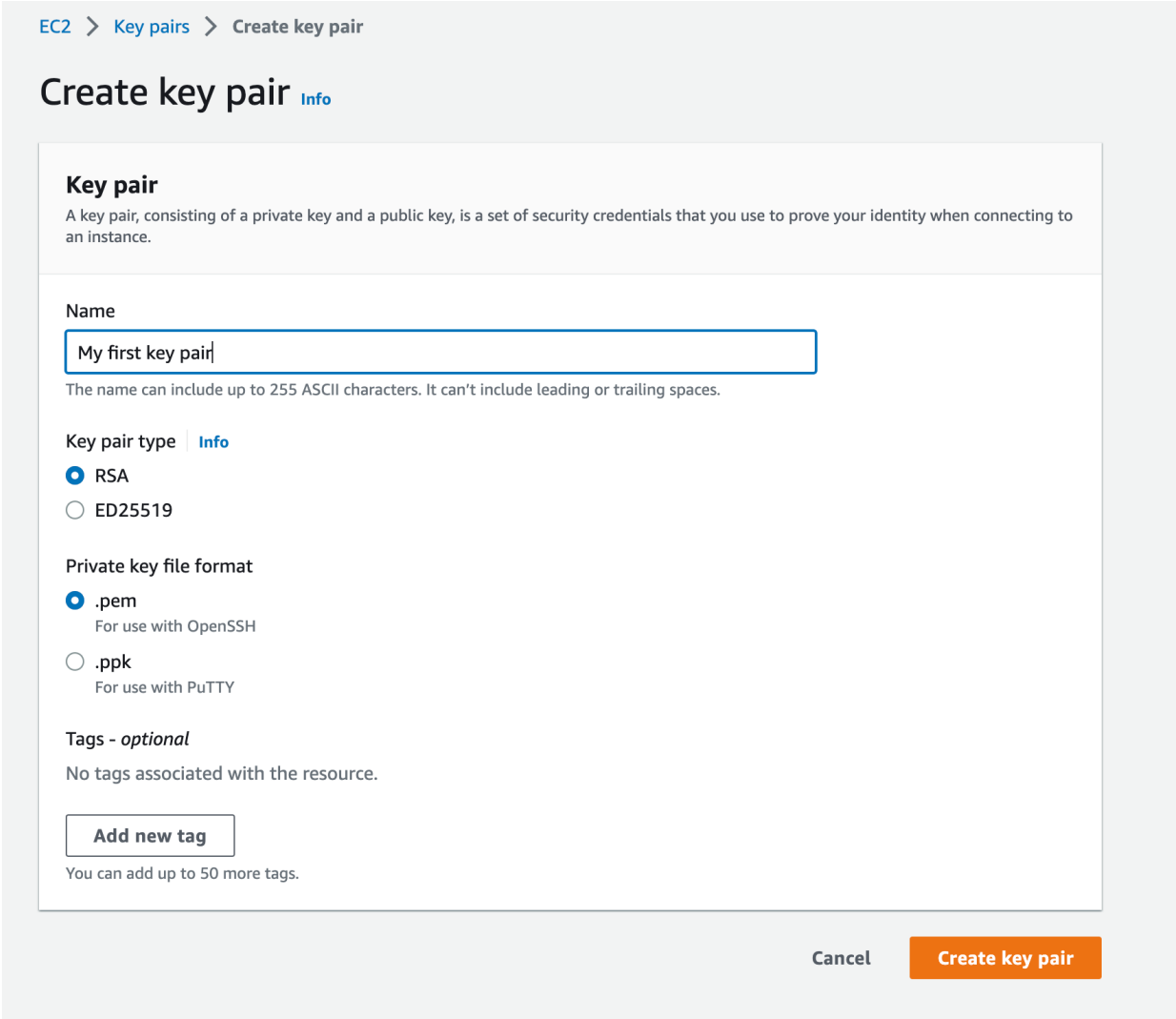
Creating a Key Pair

It's best practice to create your own Key Pair. In the EC2 console, scroll down on the panel at the left hand side and select Key Pairs

You can see the provided vockey. You can click Create Key Pair at the top right to create your own key pair



Enter a name for your key pair, and select a pem key pair if you are on Mac/Linux, or ppk if you are on Windows and you are going to use PuTTY.



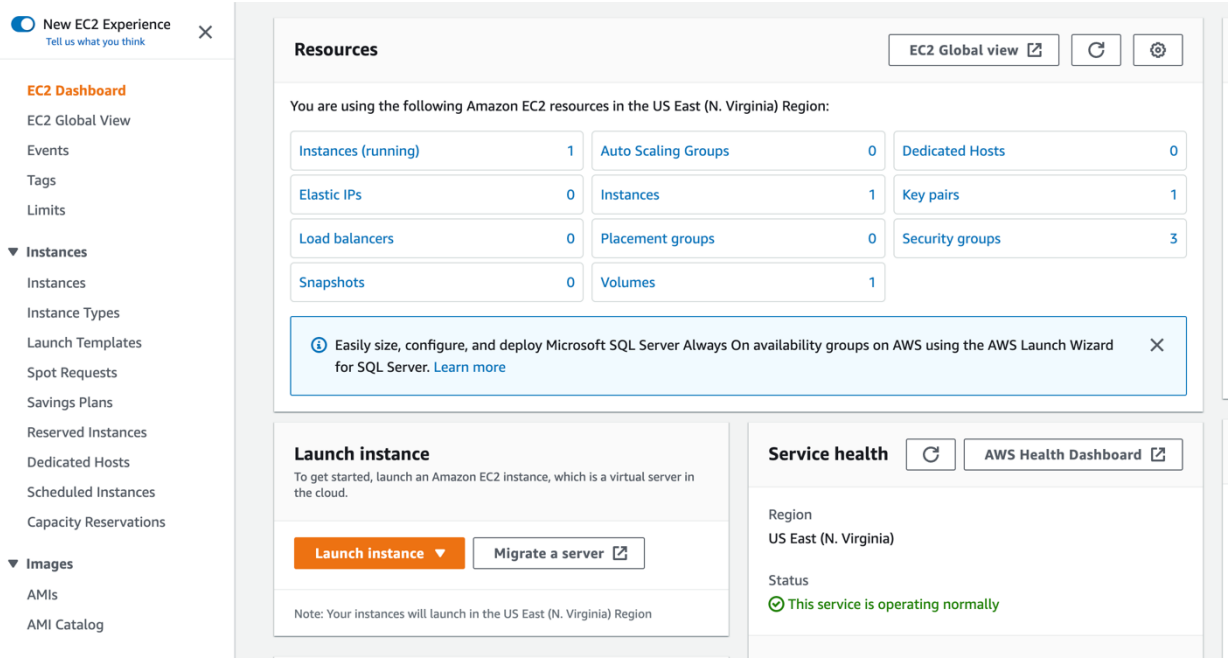
Click Create Key Pair and the key pair will be downloaded.

Note that as you are using the Sandbox the key pair will be destroyed when the Sandbox session ends so you won't need to keep the key pair after you finish this lab.

However, when using the Learner Lab for your assignments the key pair will survive the end of a lab session.

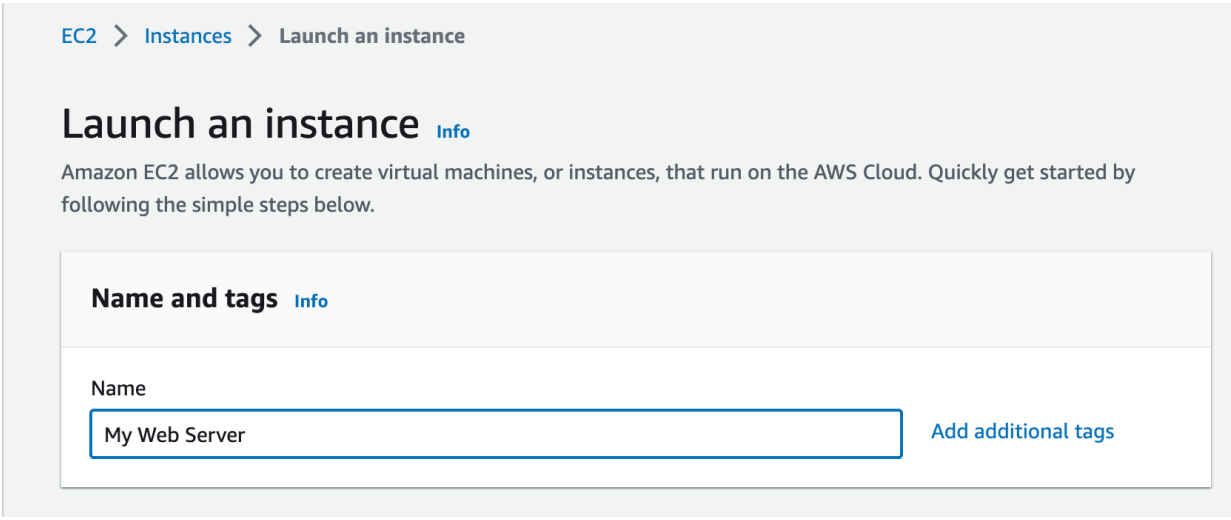
Creating your first EC2 instance

Go back to the main screen for the EC2 console. You'll see a Launch Instance button. Click on it

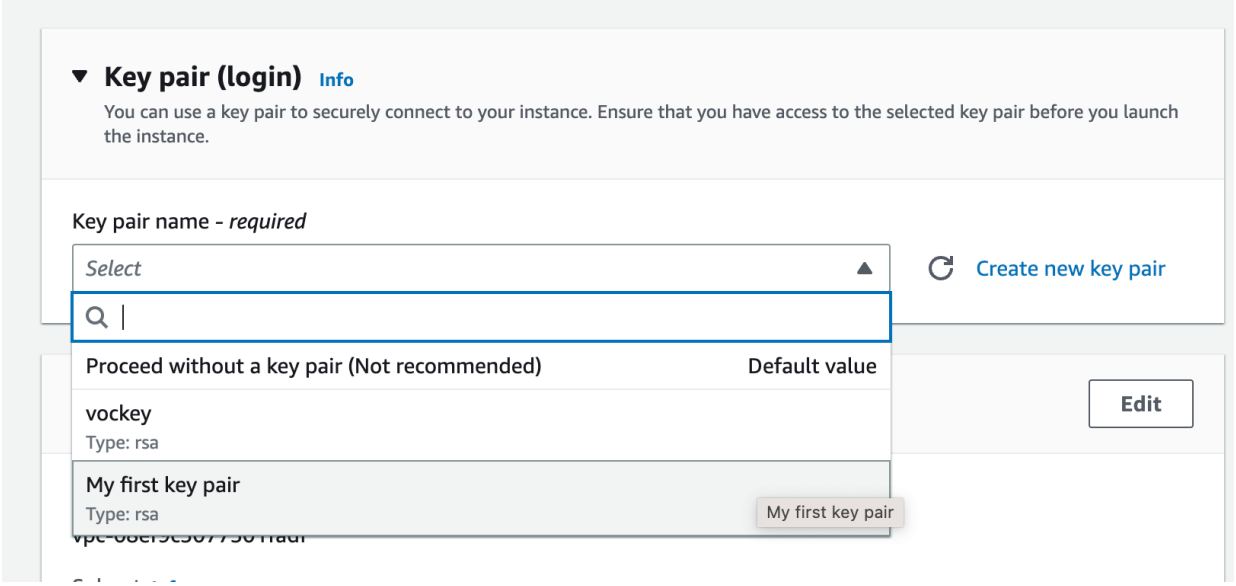


We are going to mostly useful default settings for this lab.

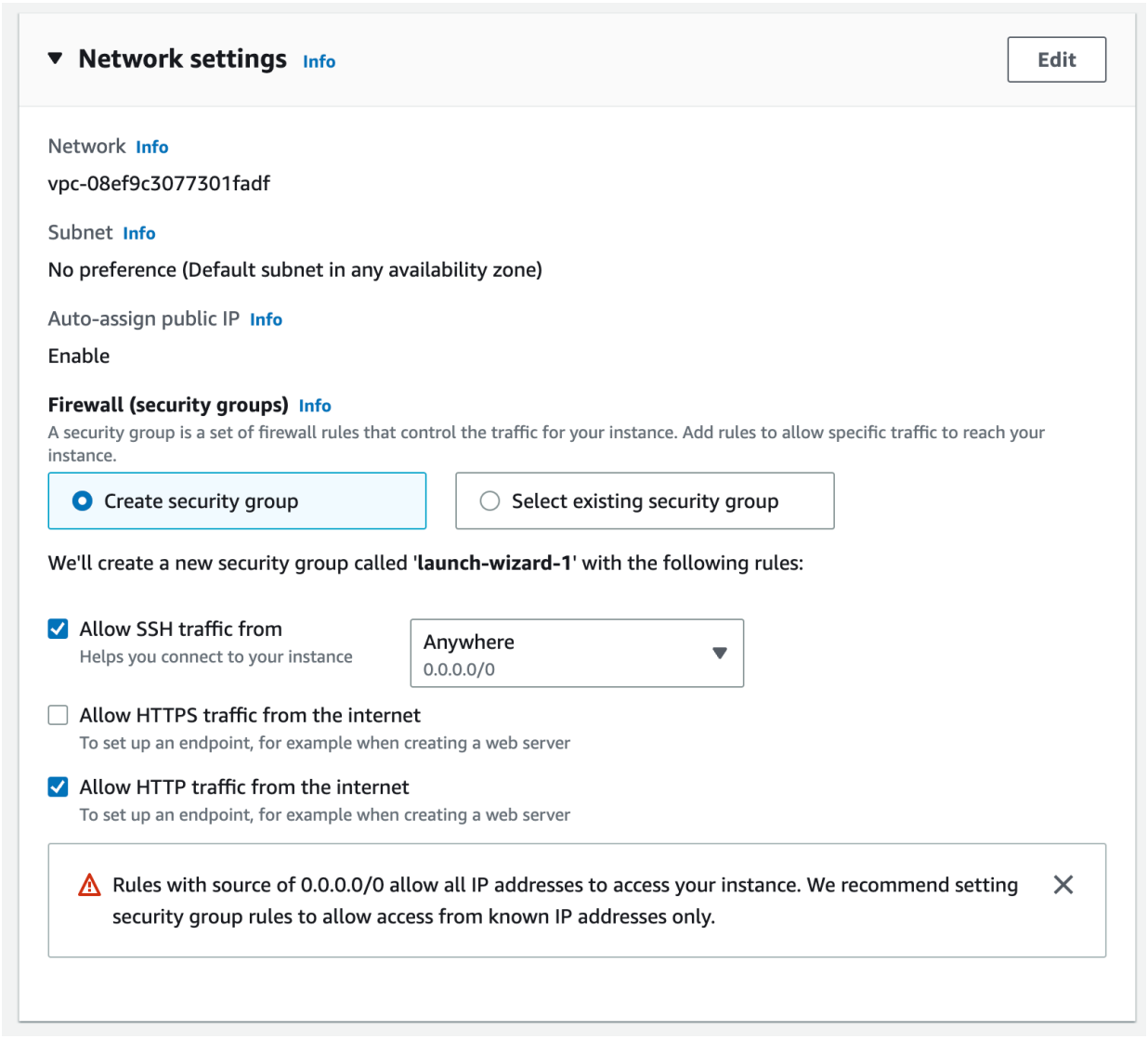
Enter a Name for the Instance e.g. My Web Server



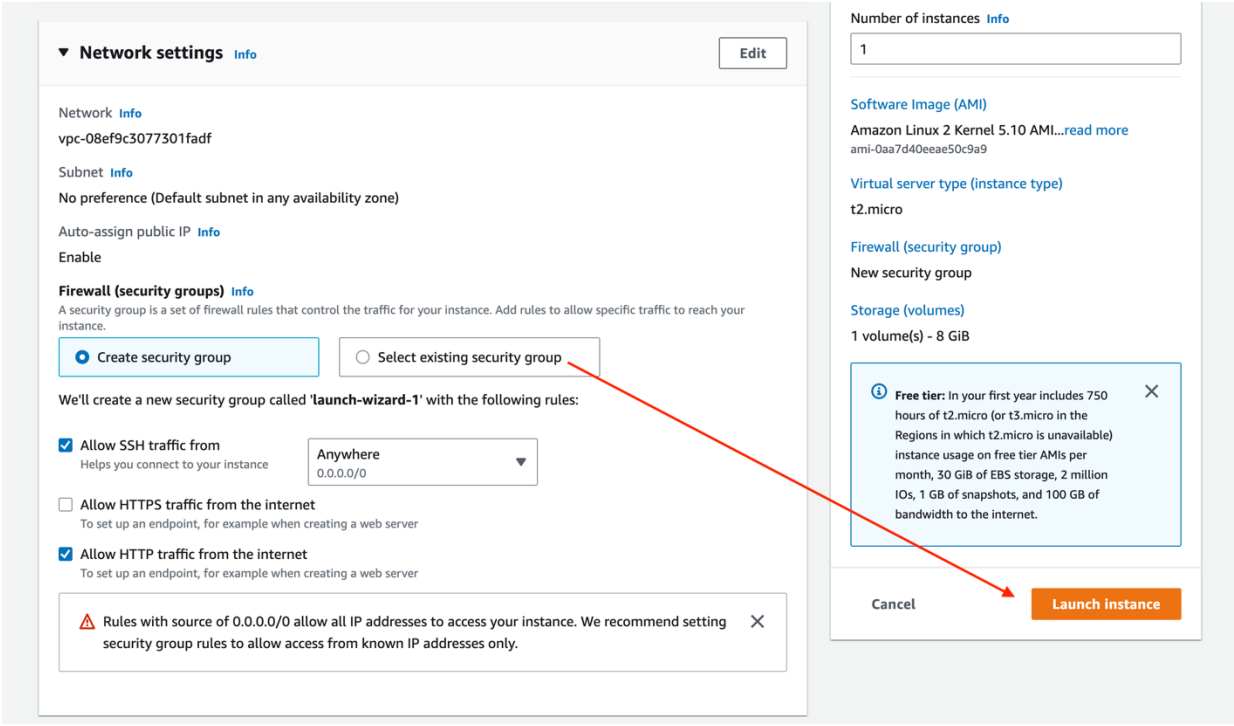
Scroll down until you see keypair and select a keypair, either the one you created if you created one, or the vockey.



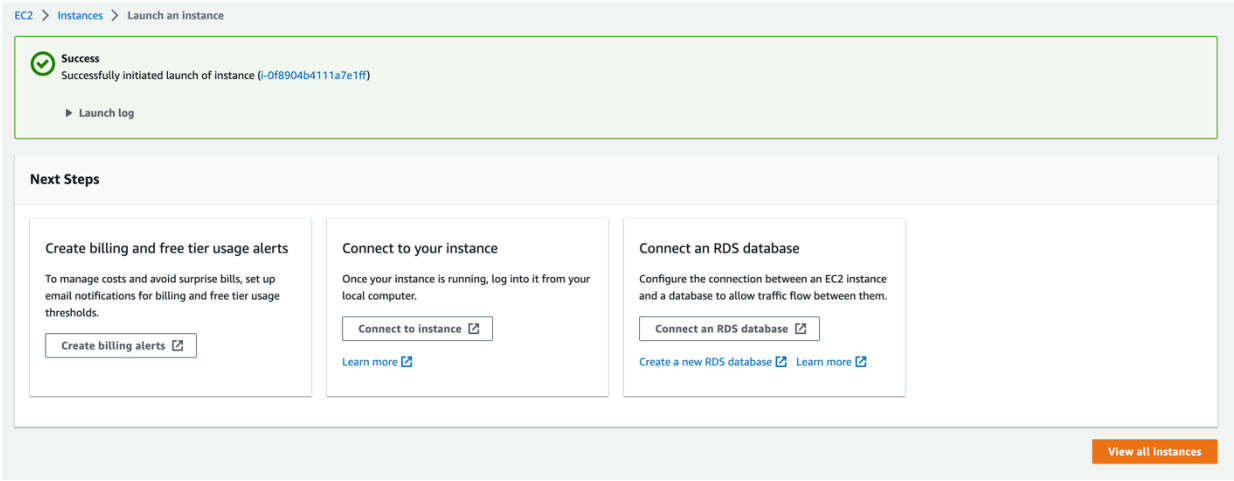
We are going to use the default VPC (not recommended but fine for the quick and dirty experimenting in this lab) and we are going to allow SSH traffic from anywhere and HTTP traffic from the internet



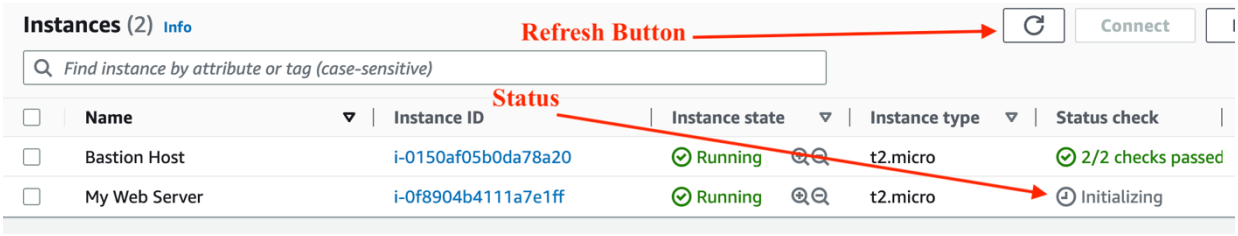
Leave all the other settings on the default and click Launch Instance



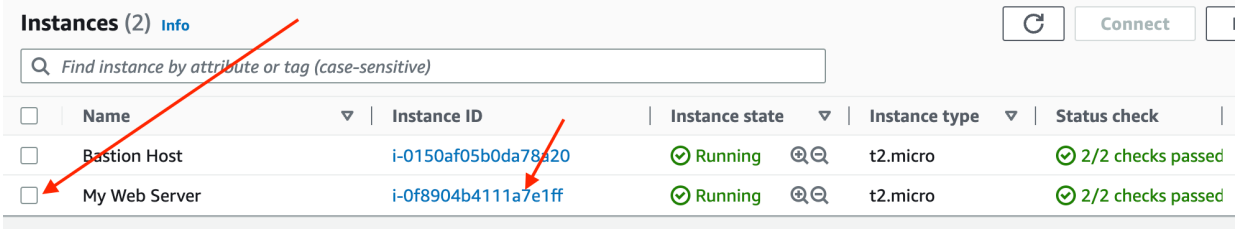
Click View All instances at the bottom right



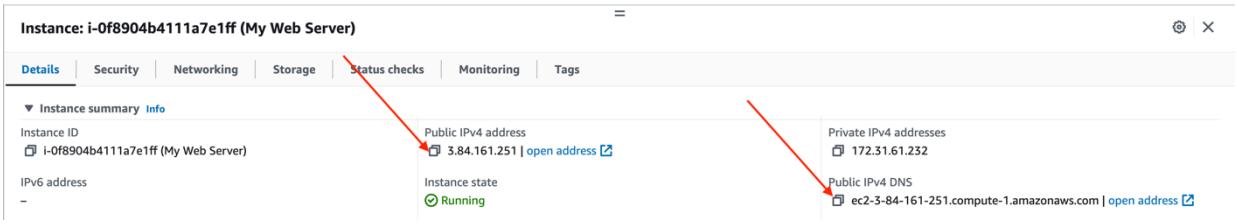
Your new instance may take a few minutes to boot up. It will show the status of initialising. Click the refresh button every 30 seconds or so until the status check changes to 2/2 passed



If you click on the checkbox next to the instance or the instance ID you will then see more information about the instance including the Public IP address and Public DNS



You can click the copy icon next to the public IP address or public DNS to copy it



SSH into EC2 instance using the Sandbox

The Sandbox instructions include the steps to connect to an EC2 instance using SSH. Your tutor will likely demonstrate this in class. Please ask your tutor for assistance if you need help with this as it is important to be able to use SSH to complete the labs and your assignments.

There are instructions for both Windows users and Mac/Linux users.

You can use the vockey or a key pair that you have created.

Linux Commands

When you have connected to Amazon Linux 2 using SSH, practice entering some common Linux Commands

Paths

Let's change the directory to the top (/) root directory (note there is a space between "cd" and /

```
cd /
```

Directories and absolute paths (i.e. exact position in the system) are always prefixed with a /. For example

```
cd ./etc
```

~ means home directory so the following command takes you to home directory. Each user has a home directory and by default this is the directory you will start in when logging in.

```
cd ~
```

dot (.) represents your current directory

The following command changes the current directory to your current directory (which is a bit pointless)

```
cd .
```

NOTE: there is a space between cd and the dot

(..) represents the parent directory.

Read and Write files

You can use echo to write a small file

```
echo "This is a test" > test.txt
```

You can use an editor such as "vim" or "vi" which are command based editors.

However, if your file is a small one you can issue a command called "cat" to see the content in a terminal.

```
cat test.txt
```

Remove a file

```
rm test.txt
```

You can also use "nano" to have a more powerful and user friendly editor. It is installed on most Linux distributions by default. Here is some basic operations in nano:

CTL+x=exit and save

CTL+o=save

Input/Output Redirection

To direct the output of a command and write it to a file you can use > sign,

e.g. `$ ls >> file_list.txt` would write the directory listing to a file.

```
ls > file_list.txt
```

>> : Redirect the output and append it to the file

```
ls >> file_list.txt
```

Pipe operator(|): Connect Standard output of a command with Standard input of the next command

```
cat filename|sort
```

Useful Commands

To see the current user

```
whoami
```

To see current directory (print working directory)

```
pwd
```

To see the content of the current directory

```
ls
```

To see all files including hidden one

```
ls -a
```

To list all details of the content of a directory

```
ls -l
```

To make a directory in your current location

```
mkdir directoryname
```

To see the Amazon Linux 2 distribution and version

```
cat /proc/version
```

To get help

```
man command          OR
info command
```

Less Command

It lets you to browse page by page the content of a file or output of a command.

```
ls –l |less
```

q : quit less

> : Go to end of file

To change the access rights and issue commands as the root (superuser) user

On Linux the user with full admin rights is the root (superuser) user. For security reasons, it is always best not to login using admin credentials but whenever you need to do something like

to install software that requires admin rights switch to the administrative role. To do so you issue the following command.

```
sudo su
```

Alternatively, you can mix both in one line and write a command like the following one that updates the yum package (yum is the primary tools to install and update programs in some Linux distributions such as RedHat and Ubuntu)

```
sudo yum update -y
```

The user data script run on the first launch of an EC2 instance is run by the superuser. That is why you don't need to use sudo in user data scripts.

Installing Programs

To install, update and remove software Amazon Linux 2 uses yum command.

```
sudo yum install -y httpd
```

The above command installs Apache (httpd), a Linux webserver.

We can install custom amazon packages using the amazon-linux-extras command.

```
sudo yum install -y php
```

The above command installs PHP

Starting Programs

To start the httpd service (just until we stop it or the EC2 instance is stopped/rebooted)

```
sudo systemctl start httpd
```

To enable the httpd service (start it each time the system is booted from the next reboot)

```
sudo systemctl enable httpd
```

Decompress gz files

With Linux we often use tar files with gz compression. To decompress gz files in the current directory

```
tar –zxvf yourfile.tar.gz
```

Troubleshooting

Sometimes after installing a program you still not sure if it up and running.

To check that httpd is active (running) issue systemctl status command

```
systemctl status httpd
```

To check all running processes such as Http and open ports, issue ss command

```
ss -tl
```

If you do not see the process and would like to check the issue we can check the logs. For the most recent 100 log entries we can do

```
journalctl -a | tail -n 100
```

Or for a specific service e.g.

```
journalctl -a | grep httpd | tail -n 100
```

ADVANCED: Creating a php info file

Let's use vim to create a php info file

First let's open the file we wish to create e.g.

```
sudo vim /var/www/html/phpinfo.php
```

Then press the i key to enable insert

You should see insert at the bottom left of the Terminal or PuTTY window

Now type in or paste in the following

```
<?php
phpinfo();
?>
```

Once you have done this press the ESC (escape) key and type ":wq" (no quotes, w means write and q means quit). You should see :wq at the bottom left of the PuTTY/Terminal Window.

Press Enter/Return and the changes you made to the file will be saved.

If you made a mistake and want to recreate the file from scratch you can remove the php info file using the rm command

```
sudo rm /var/www/html/phpinfo.php
```

Then try creating it again.

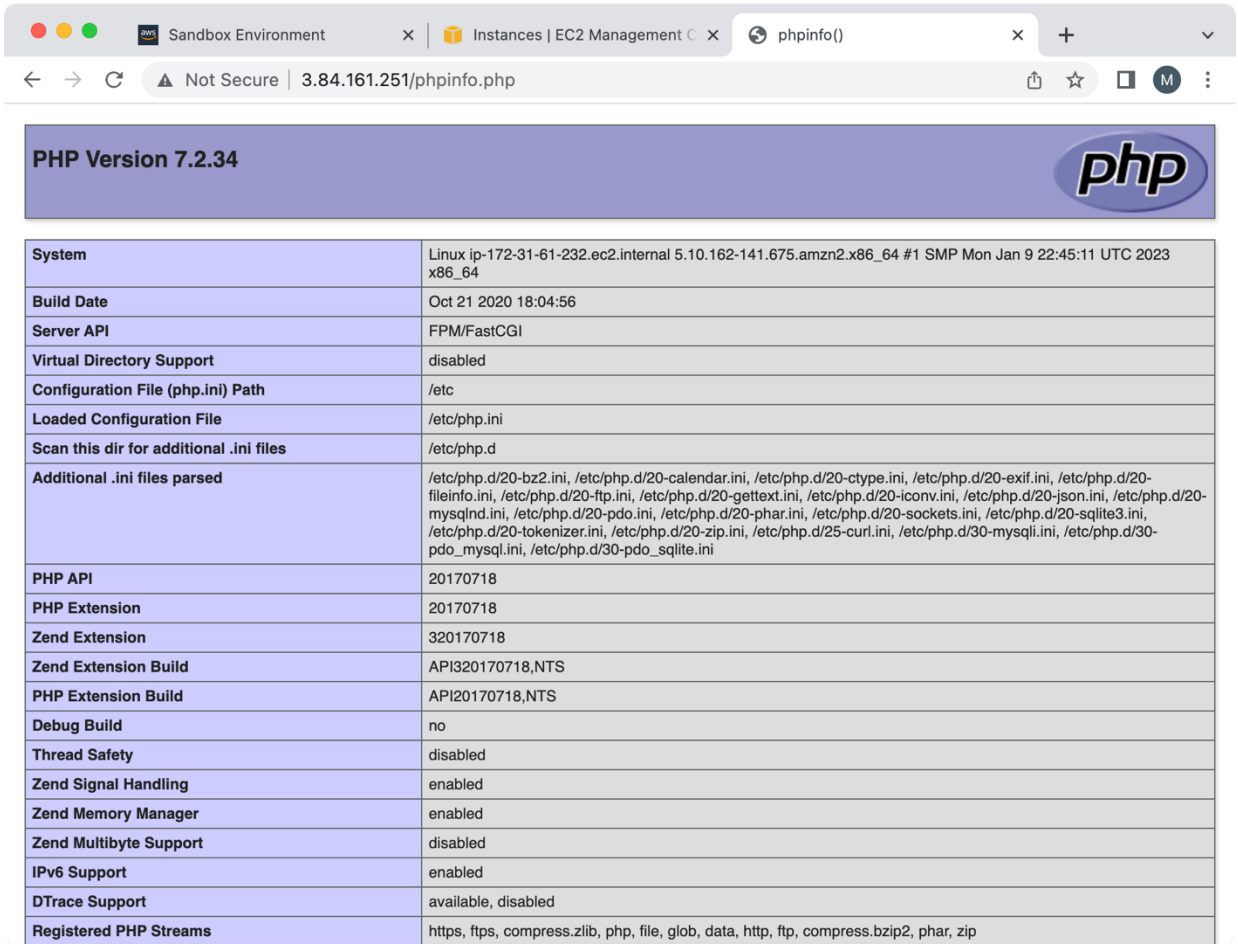
To test the below you will need the phpinfo file to still be there

Open the web page

Open the web page. In my example it would be http://3.84.161.251/phpinfo.php

Note your IP will be different and be sure that the URL starts with http not https or it won't work.

You should see a PHP info page similar to below

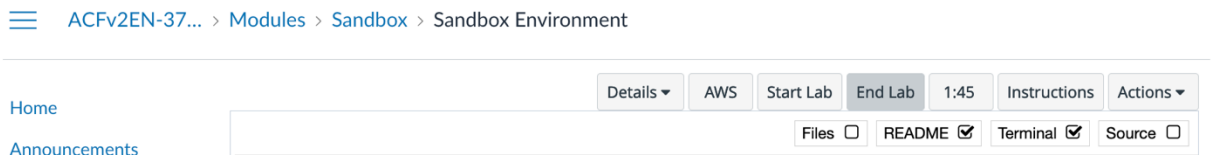


System	Linux ip-172-31-61-232.ec2.internal 5.10.162-141.675.amzn2.x86_64 #1 SMP Mon Jan 9 22:45:11 UTC 2023 x86_64
Build Date	Oct 21 2020 18:04:56
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-zip.ini, /etc/php.d/25-curl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip

Ending the Lab…

There is no demonstration this week. But from next week you MUST demonstrate the lab or take all needed screenshots for a lab report prior to ending the lab. Otherwise, you will need to redo the lab again.

When you are done you can click the End Lab button



Learning more…

There are many good Linux resources and tutorials on the Web. Here are some

links:

https://www.w3schools.in/operating-system-tutorial/linux-operating-system/

http://linuxcommand.org/index.php

http://www.theunixschool.com/

http://www.tldp.org/LDP/abs/html/textproc.html