



ÅBO AKADEMI UNIVERSITY

CLOUD COMPUTING

Assignment 1



FILIPPE FELÍCIO (2004623)

APRIL 9, 2021

Launching an AWS VM

Login on the AWS console

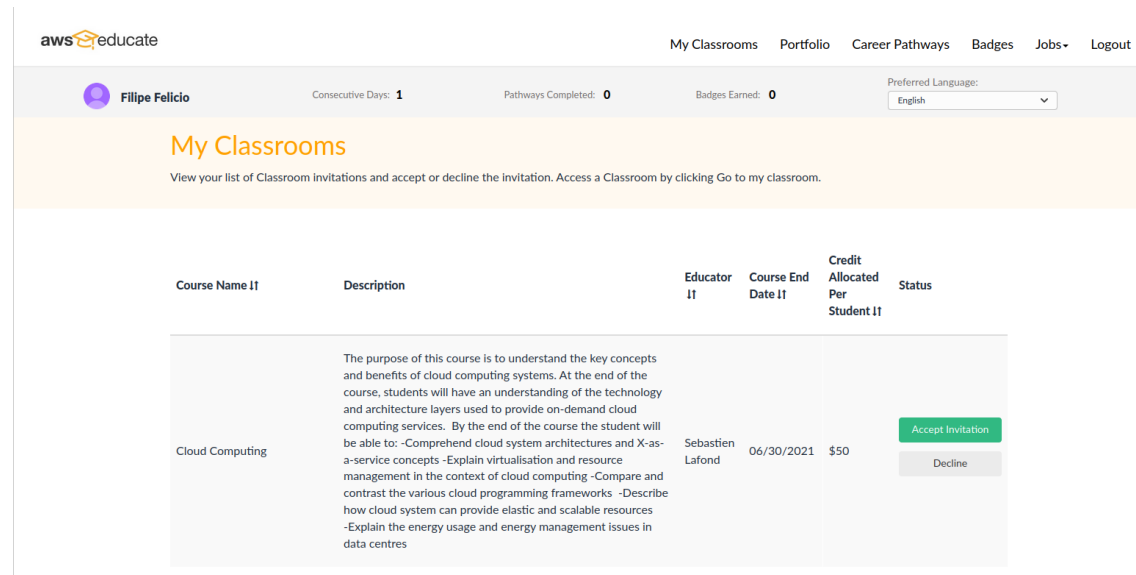


Figure 1: Login on the AWS console

Launch an instance (Virtual machine - VM) via the console

First I had to choose the Amazon Machine Image, in this case the Amazon Linux 2 AMI (HVM).

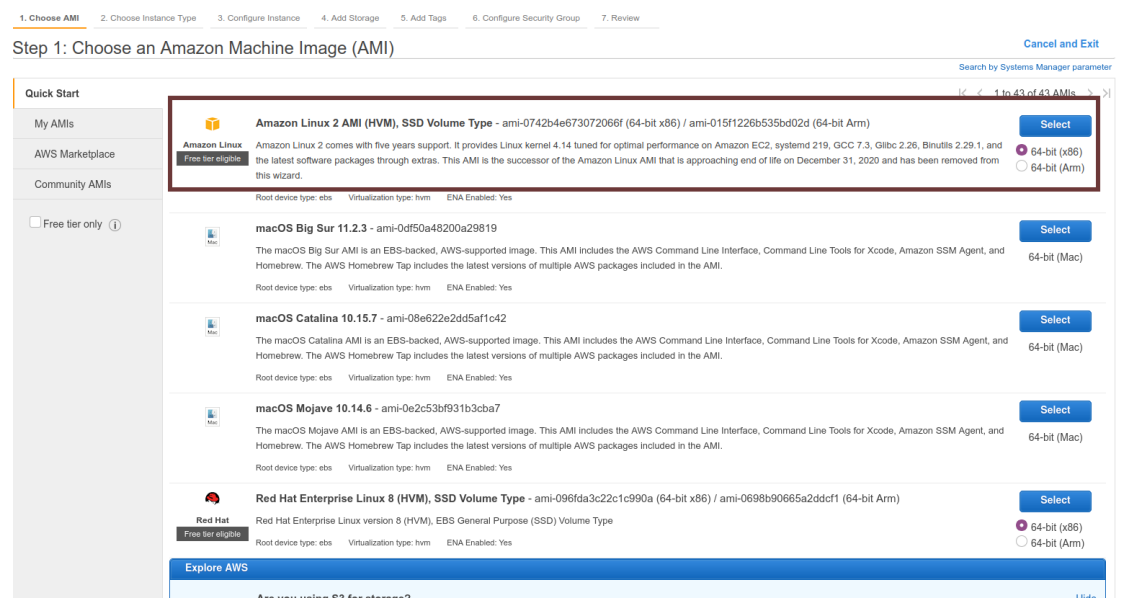


Figure 2: Choosing the AMI

Then I picked the t2.nano as my instance type.

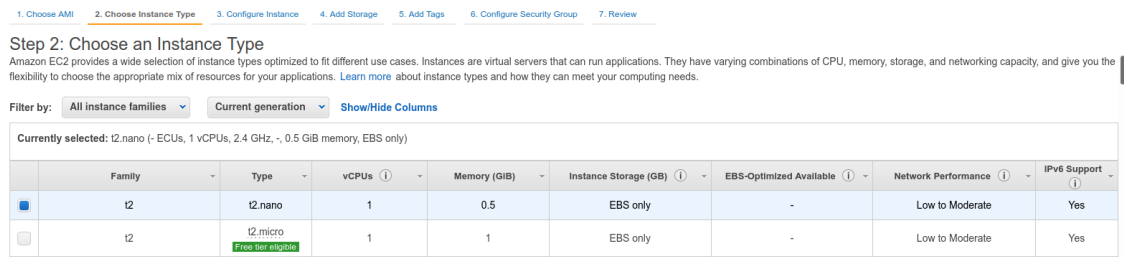


Figure 3: Selecting Type Instance

After I configured the instance details, in this case I left the default values.

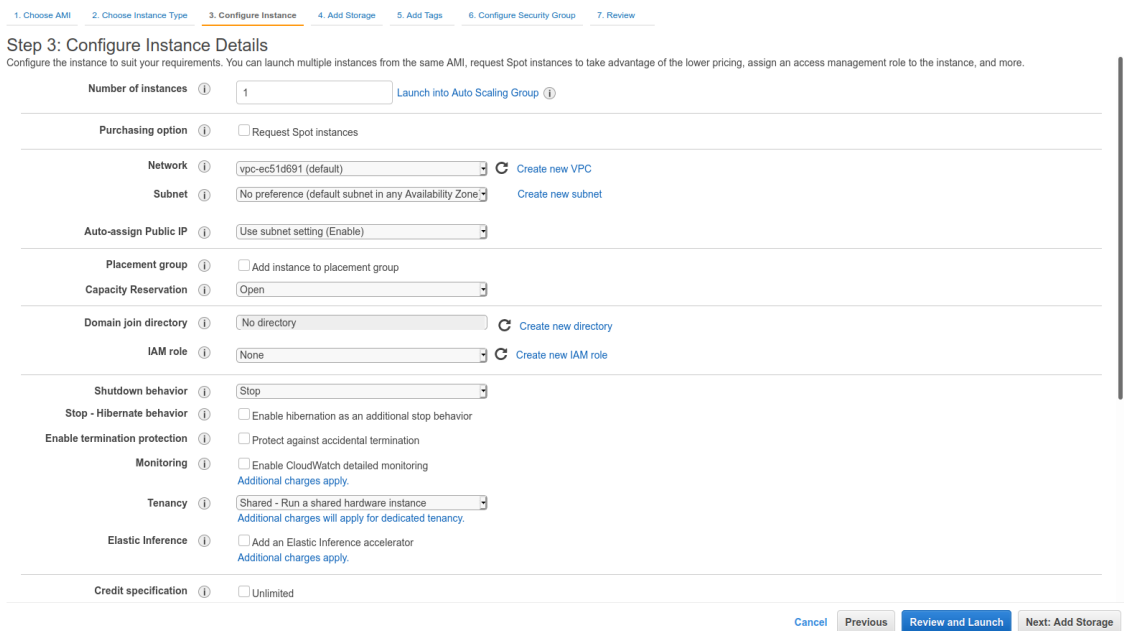


Figure 4: Configuring the Instance Details

Later I added Storage, here I also left the default values.

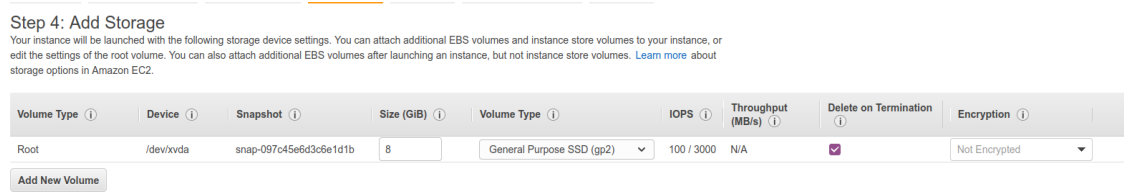


Figure 5: Adding Storage

Then it came the step of adding tags, here I did not add one.

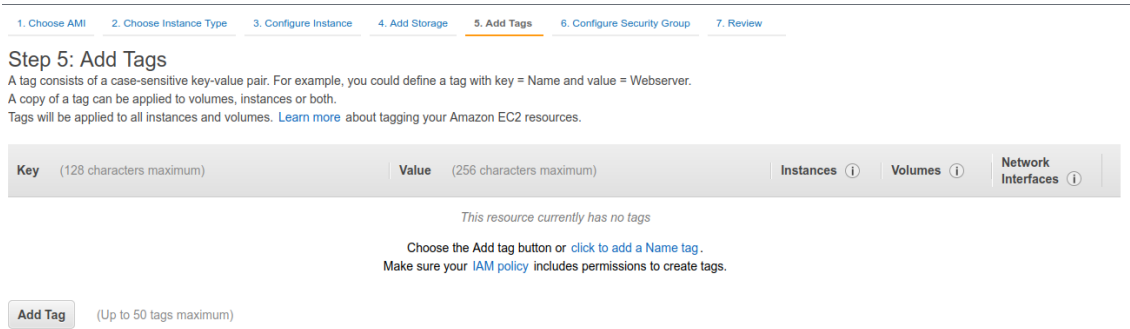


Figure 6: Adding Tags

After I configured the security group, here I changed the name of the security group to "Default" and added and HTTP Rule.

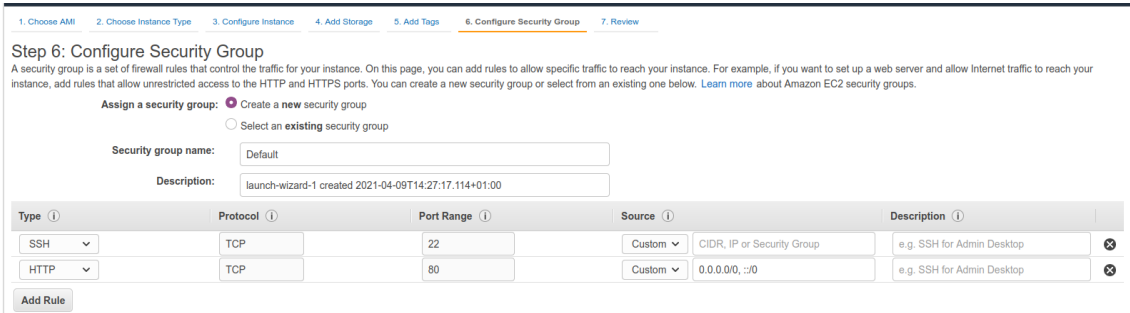


Figure 7: Configuring the security group

Finally, I Review my Instance Launch

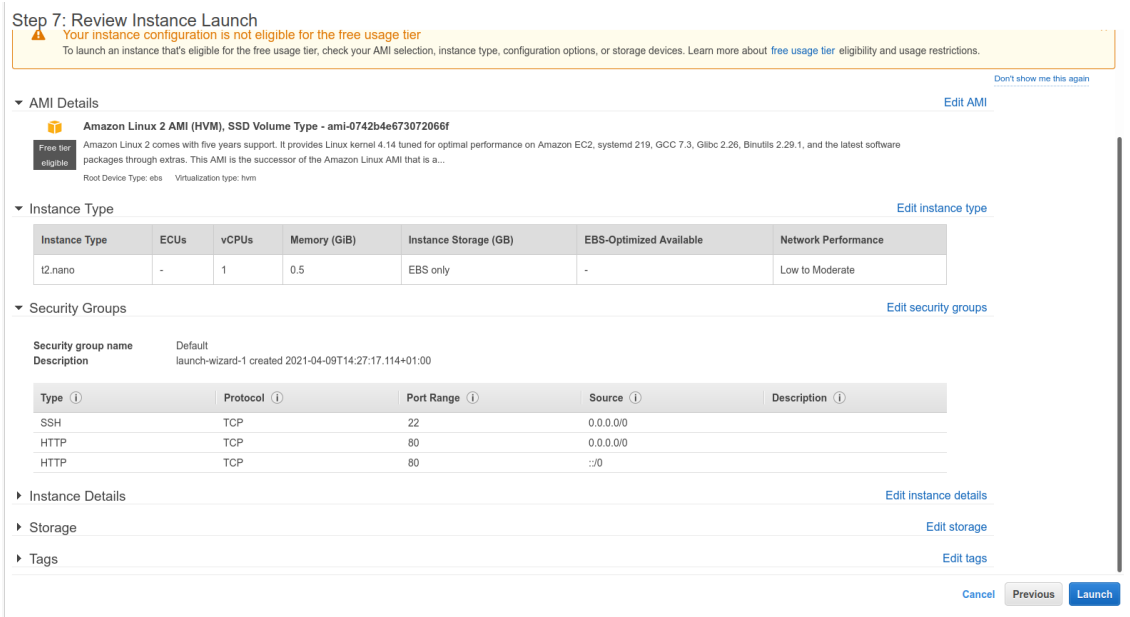


Figure 8: Reviewing the Instance Launch

Afterwards I had to create a new key pair.

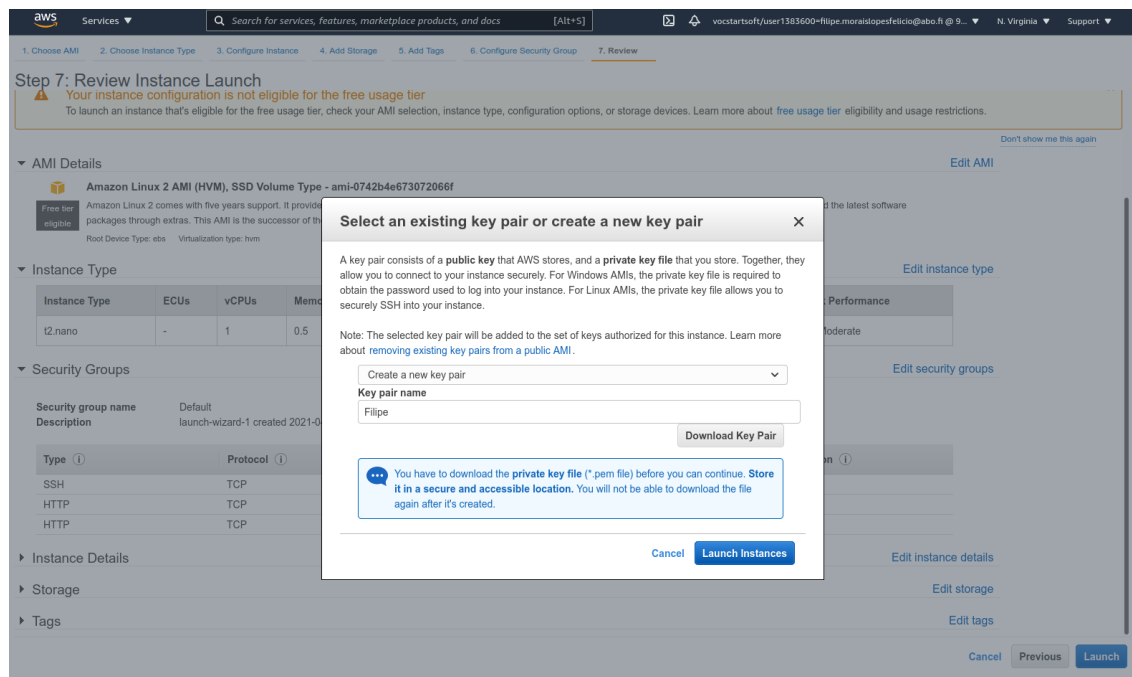


Figure 9: Creating a new key pair

Then I selected the newly created key pair.

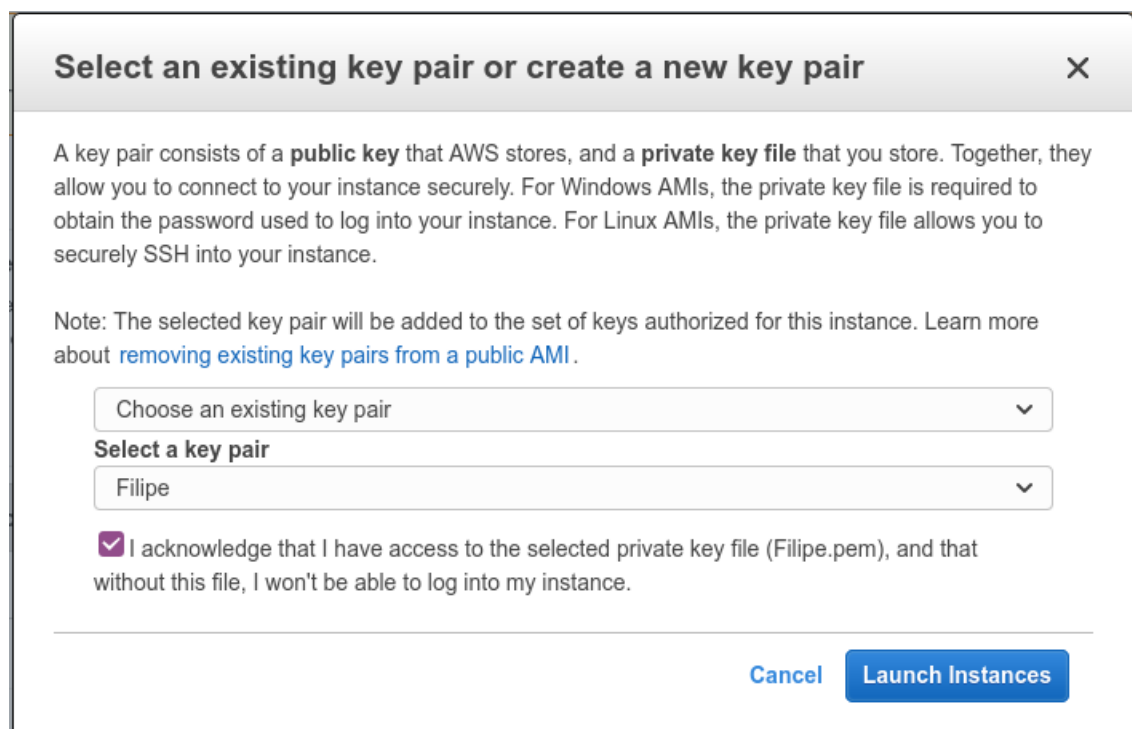


Figure 10: Selecting the key pair

Because the name "Default" was giving me errors, I decided to leave it as default.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

Figure 11: Re-configuring the Security Group

I was then able to launch the instance.

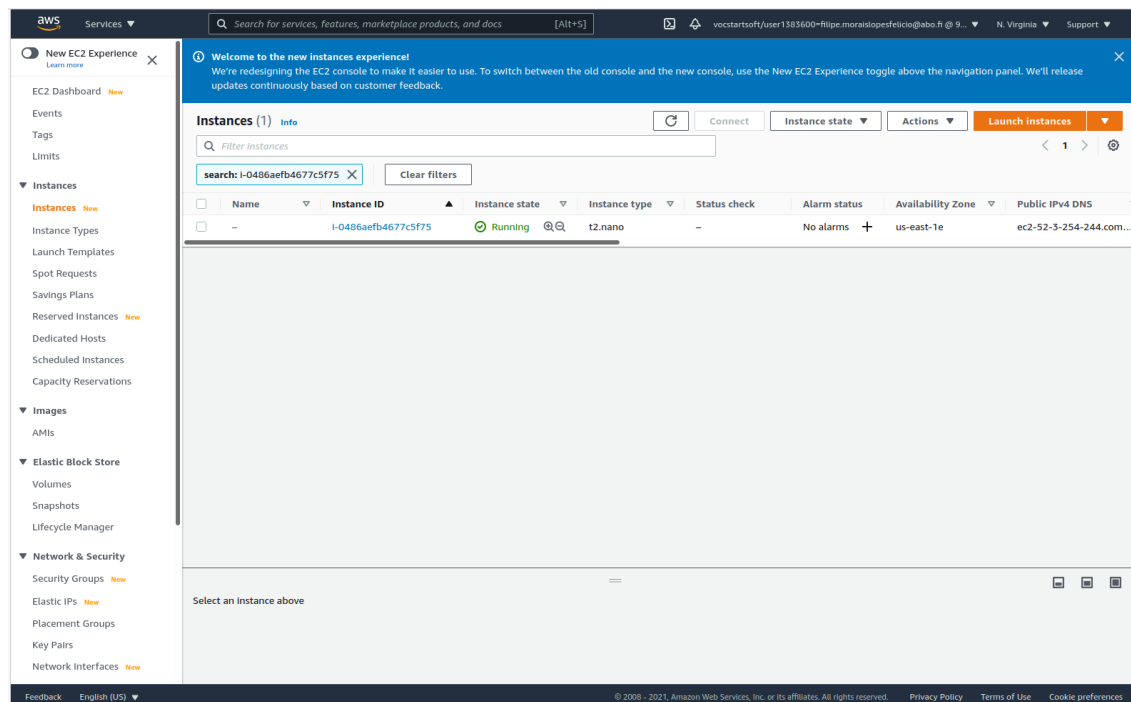


Figure 12: Launching the Instance

Using SSH and your key file, connect to your remote VM from your local machine.

First I check the information to connect to the instance.

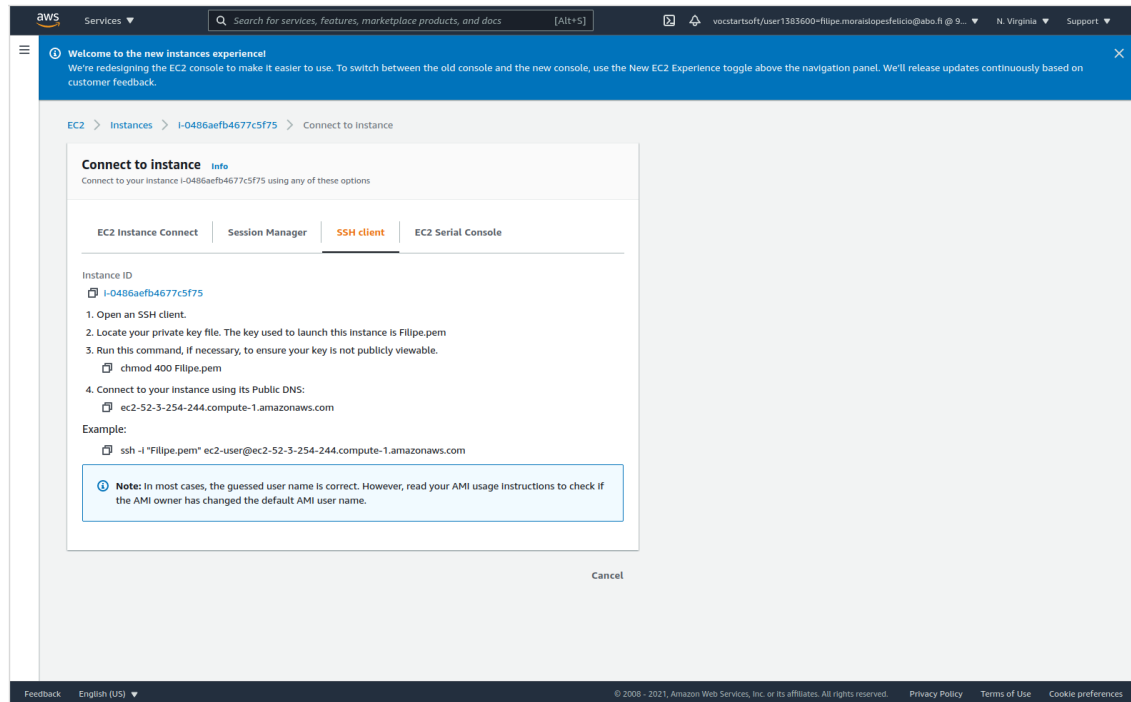


Figure 13: Checking the information to connect to the instance

Then using the following command I connected to the remote Virtual Machine from my local machine.

```
1 $ ssh -i "Filipe.pem" ec2-user@ec2-52-3-254-244.compute-1.amazonaws.com
```

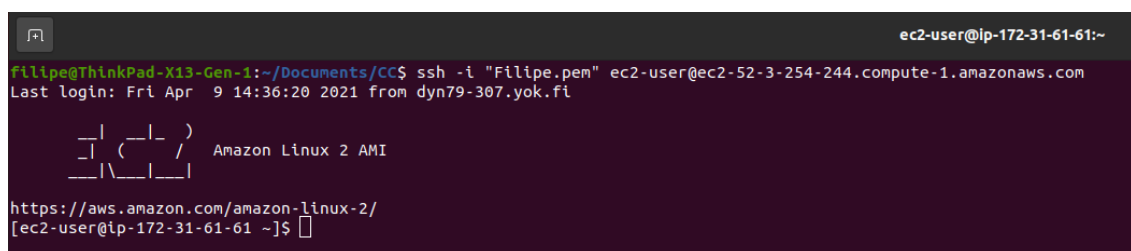


Figure 14: Connecting to the remote VM from my local machine

Check and report the following information

To answer the first four questions I used the following command:

```
1 $ cat /proc/cpuinfo
```

1. **What is the model name of your CPUs?**
Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz

2. What is the cache size?
30720 KB
3. What is the clock frequency of your CPU(s)?
2400.290 MHz
4. What is the CPU vendor?
GenuineIntel

```

[ec2-user@ip-172-31-61-61 ~]$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu_family     : 0
model          : 63
model name     : Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
stepping       : 2
microcode     : 0x44
cpu MHz        : 2400.290
cache size     : 30720 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep ntrr pge nca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology cpuid pni pclmulqdq ssse3
bugs           : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs itlb_multitit
bogomips       : 4800.14
clflush size   : 64
cache alignment : 64
address sizes   : 40 bits physical, 48 bits virtual
power management:

[ec2-user@ip-172-31-61-61 ~]$

```

Figure 15: Checking VM info

To answer the final question, I used this command:

```
1 $ sudo dmidecode
```

5. What is the name of the hypervisor vendor?
Xen

```

[ec2-user@ip-172-31-61-61 ~]$ sudo dmidecode
# dmidecode 3.0
Getting SMBIOS data from sysfs.
SMBIOS 2.7 present.
11 structures occupying 359 bytes.
Table at 0x000E801F:

Handle 0x0000, DMI type 0, 24 bytes
BIOS Information
    Vendor: Xen
    Version: 4.2.amazon
    Release Date: 08/24/2006
    Address: 0xE8000
    Runtime Size: 96 KB
    ROM Size: 64 KB
    Characteristics:
        PCI is supported
        ESD is supported
        Targeted content distribution is supported
    BIOS Revision: 4.2

```

Figure 16: Checking VM info

Still from your VM, execute the following command line:

```
1 $ curl "ec2-18-156-136-12.eu-central-1.compute.amazonaws.com/logs.php?name=filipe_felicio" > log.dat
```

```

[ec2-user@ip-172-31-61-61 ~]$ curl "ec2-18-156-136-12.eu-central-1.compute.amazonaws.com/logs.php?name=filipe_felicio" > log.dat
% Total    % Received % Xferd  Average Speed   Time    Time     Time    Current
           Dload  Upload   Total   Spent    Left     Speed
100 125    0 125    0    0    603    0 --:--:-- --:--:-- --:--:--    600
[ec2-user@ip-172-31-61-61 ~]$ ls
log.dat
[ec2-user@ip-172-31-61-61 ~]$ cat log.dat
Name: filipe_felicio -- 52.3.254.244 -- ec2-52-3-254-244.compute-1.amazonaws.com -- 172.31.36.77 -- 1617975577 -- curl/7.61.1[ec2-user@ip-172-31-61-61 ~]$

```

Figure 17: Running the given command

Download the created log.dat file on your computer

For this step I used the following command:

```
1 $ sudo scp -i "Filipe.pem" ec2-user@ec2-52-3-254-244.compute-1.amazonaws.com:log.dat .
```

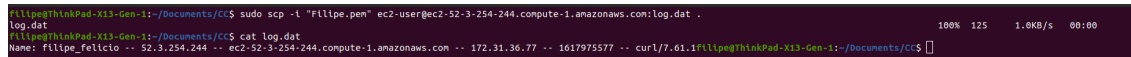


Figure 18: Downloading the created log.dat file on my computer

Content of my log.dat file

For this step I used the following command:

```
1 $ cat log.dat
```

That showed me the following output:

```
1 Name: filipe_felicio — 52.3.254.244 — ec2-52-3-254-244.compute-1.amazonaws.com  
— 172.31.36.77 — 1617975577 — curl/7.61.1
```

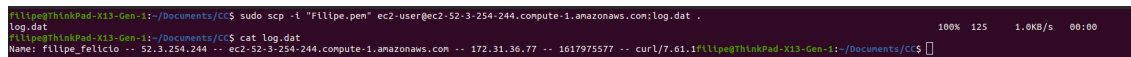


Figure 19: Viewing the content of the log.dat file on my computer

Terminate the Virtual Machine

Finally, I terminated the Virtual Machine.

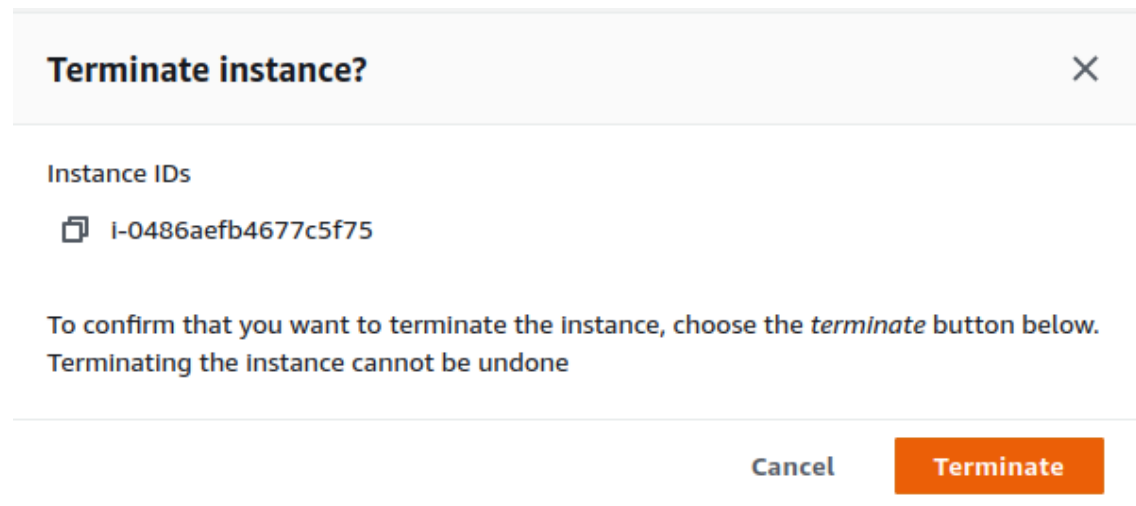
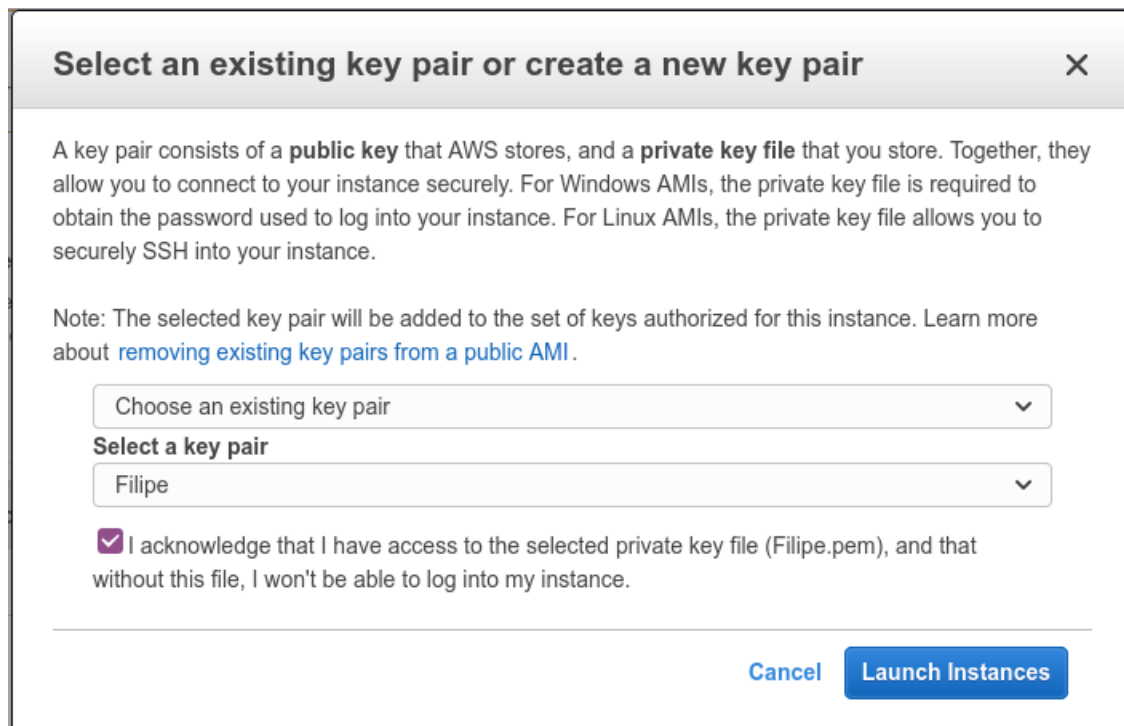


Figure 20: Terminating the Virtual Machine

Report

What would happen if you lose the private key provided when you instantiated your VM?

If I lost the selected private key file "Filipe.pem" I would not be able to log into the instantiated VM.



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair ▼

Select a key pair

Filipe ▼

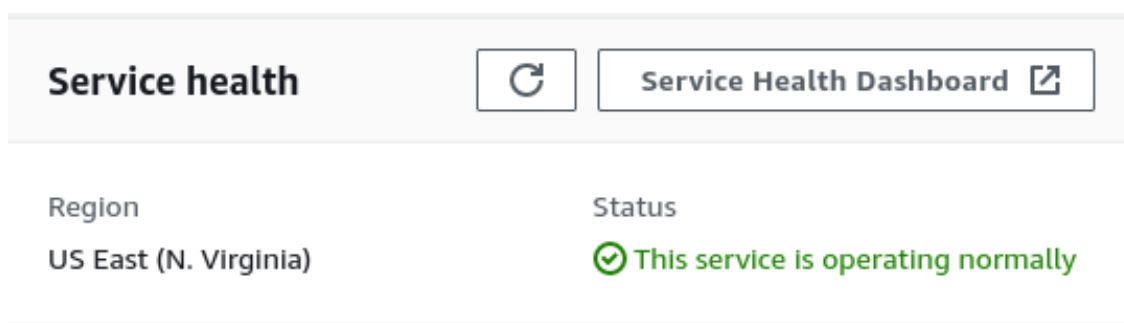
☒ I acknowledge that I have access to the selected private key file (Filipe.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

Figure 21: Selecting the key pair

Do you have any idea where was the physical server on which your VM was running?

The Location was US East (N.Virginia).



Region	Status
US East (N. Virginia)	✔ This service is operating normally

Figure 22: Physical Server Location

How long was the “waiting time” (approximately) between requesting a VM and having it up and running?

Approximately 20 seconds.

Conclusion

This assignment allow me to learn something complete new, as a Linux *aficionado*, I was familiar with the concept of Virtual Machines running in my local machine but never had the experience

of setting up a connection between my local machine and remote VM in the other side of the globe using AWS (I have done a connection with a university machine, back in my home institution in Portugal). Nothing surprised me and with the assignment guidelines and my academic background nothing was difficult. It was really satisfying, on my professional life I have been developing a web product where me and my team have the question on weather to choose a NAV Server in the company or a Cloud Service such as AWS to deploy my Databases and this first experience was very valuable.