



ÅBO AKADEMI UNIVERSITY

CLOUD COMPUTING

## Assignment 2



FILIBE FELÍCIO (2004623)

APRIL 14, 2021

## VM performance and cost-efficiency comparisons

### 1) Create an account on openbenchmarking.org

<https://openbenchmarking.org/user/feliciofilipe>

### 2) Have a look at all possible instance types provided by AWS

#### General Purpose

- **T2 family**

T2 instances are Burstable Performance Instances that provide a baseline level of CPU performance with the ability to burst above the baseline. T2 Unlimited instances can sustain high CPU performance for as long as a workload needs it. For most general-purpose workloads, T2 Unlimited instances will provide ample performance without any additional charges.

- **M4 family**

M4 instances provide a balance of compute, memory, and network resources, and it is a good choice for many applications.

- **M5 family**

M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175M processors. This family provides a balance of compute, memory, and network resources, and is a good choice for many applications.

#### Compute Optimized

- **C4 family**

M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175M processors. This family provides a balance of compute, memory, and network resources, and is a good choice for many applications.

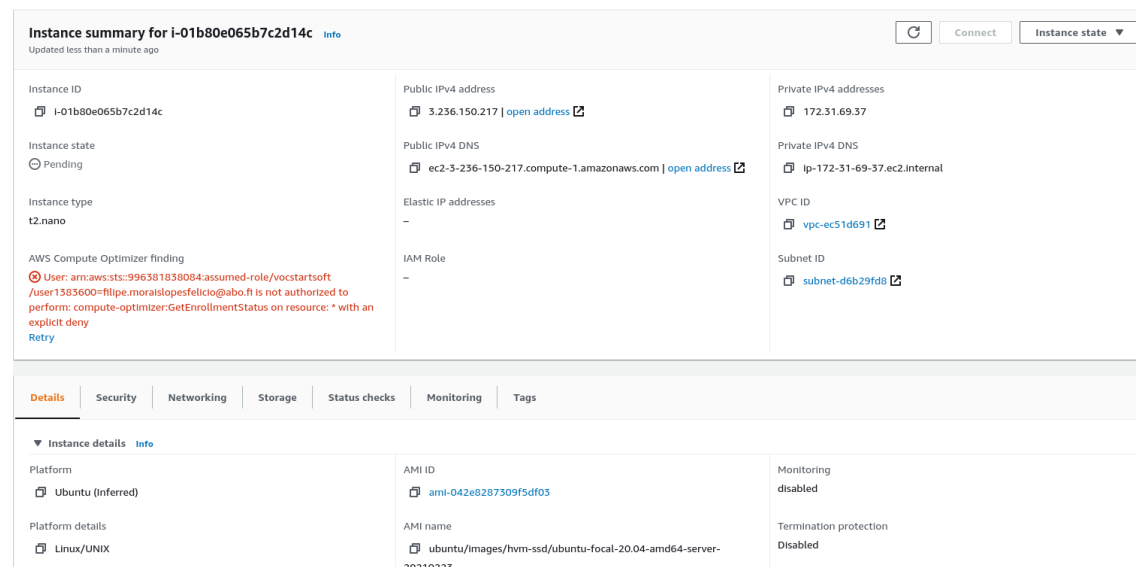
- **C5 family**

M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175M processors. This family provides a balance of compute, memory, and network resources, and is a good choice for many applications.

### 3) Select different instance types, with at least one from each of the following family: T, M and C.

I will select the '**t2.nano**', the '**t2.medium**', the '**m5.large**' and the '**c5.xlarge**' instance types. The goal was to have one from each family and different combinations of CPU and memory (from the nano to the xlarge).

#### 4) Launch the first VM - t2.nano



**Instance summary for i-01b80e065b7c2d14c** [Info](#)

Updated less than a minute ago

<b>Instance ID</b> i-01b80e065b7c2d14c	<b>Public IPv4 address</b> 3.236.150.217   <a href="#">open address</a>	<b>Private IPv4 addresses</b> 172.31.69.37
<b>Instance state</b> Pending	<b>Public IPv4 DNS</b> ec2-3-236-150-217.compute-1.amazonaws.com   <a href="#">open address</a>	<b>Private IPv4 DNS</b> ip-172-31-69-37.ec2.internal
<b>Instance type</b> t2.nano	<b>Elastic IP addresses</b> -	<b>VPC ID</b> vpc-ec51d691
<b>AWS Compute Optimizer finding</b> User: arn:aws:sts::996381838084:assumed-role/vocstartsoft /user1383600=filipe.moraislopesfelicio@abo.fi is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny <a href="#">Retry</a>	<b>IAM Role</b> -	<b>Subnet ID</b> subnet-d6b29fd6

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**▼ Instance details** [Info](#)

<b>Platform</b> Ubuntu (Inferred)	<b>AMI ID</b> ami-042e8287309f5df03	<b>Monitoring</b> disabled
<b>Platform details</b> Linux/UNIX	<b>AMI name</b> ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20210311	<b>Termination protection</b> Disabled

Figure 1: Launching the first VM with the instance type t2.nano

#### 5) Packages installation

Install the phoronix-test-suite

```
ubuntu@ip-172-31-69-37:~$ sudo dpkg -i phoronix-test-suite_10.2.2_all.deb
(Reading database ... 67263 files and directories currently installed.)
Preparing to unpack phoronix-test-suite_10.2.2_all.deb ...
Unpacking phoronix-test-suite (10.2.2) over (10.2.2) ...
Setting up phoronix-test-suite (10.2.2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Figure 2: Installing the phoronix-test-suite

Install the 'unzip' package

```
ubuntu@ip-172-31-69-37:~$ sudo apt-get install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-25ubuntu1).
```

Figure 3: Installing the 'unzip' package

## 6) Install the following Test Suites and Tests and the required dependencies

pts/openssl

```
pts/openssl-1.11.0:
  Test Installation 1 of 1
  1 File Needed [7.95 MB]
  Downloading: openssl-1.1.1.tar.gz [7.95MB]
  Downloading .....
  Approximate Install Size: 48 MB
  Estimated Install Time: 1 Minute, 9 Seconds
  Installing Test @ 14:05:58
```

Figure 4: Installing the pts/openssl

pts/stream

```
pts/stream-1.3.1:
  Test Installation 1 of 1
  1 File Needed [0.01 MB / 1 Minute]
  Downloading: stream-2013-01-17.tar.bz2 [0.01MB]
  Estimated Download Time: 1m .....
  Approximate Install Size: 0.1 MB
  Estimated Install Time: 2 Seconds
  Installing Test @ 14:09:58
```

Figure 5: Installing the pts/stream

pts/encode-mp3

```
To Install: pts/timed-audio-encode-1.0.1
To Install: pts/encode-mp3-1.7.4

Determining File Requirements .....
Searching Download Caches .....

2 Tests To Install
  2 Files To Download [76.45MB]
  5MB Of Disk Space Is Needed
  42 Seconds Estimated Install Time

pts/timed-audio-encode-1.0.1:
  Test Installation 1 of 2
  1 File Needed [75 MB / 2 Minutes]
  Downloading: pts-trondheim-wav-3.tar.gz [75.00MB]
  Estimated Download Time: 2m .....
  Estimated Install Time: 3 Seconds
  Installing Test @ 14:19:57

pts/encode-mp3-1.7.4:
  Test Installation 2 of 2
  1 File Needed [1.45 MB / 1 Minute]
  Downloading: lame-3.100.tar.gz [1.45MB]
  Download Failed: http://sourceforge.mirror-service.org/l/la/lame/lame/3.100/lame-3.100.tar.gz
  Attempting to download from alternate mirror.
  Downloading: lame-3.100.tar.gz [1.45MB]
  Estimated Download Time: 1m .....
  Approximate Install Size: 5 MB
  Estimated Install Time: 39 Seconds
  Installing Test @ 14:20:01
```

Figure 6: Installing the pts/encode-mp3

## pts/apache

```

To Install: pts/apache-1.7.2

Determining File Requirements .....
Searching Download Caches .....

1 Test To Install
  4 Files To Download [7.49MB]
  208MB Of Disk Space Is Needed
  2 Minutes, 45 Seconds Estimated Install Time

pts/apache-1.7.2:
  Test Installation 1 of 1
  4 Files Needed [7.49 MB / 1 Minute]
  Downloading: httpd-2.4.29.tar.bz2 [6.26MB]
  Estimated Download Time: 1m .....
  Downloading: apache-ab-test-files-1.tar.gz [0.01MB]
  Estimated Download Time: 1m .....
  Downloading: apr-1.6.3.tar.bz2 [0.81MB]
  Download Failed: http://apache.melbournnetmirror.net/apr/apr-1.6.3.tar.bz2
  Attempting to download from alternate mirror.
  Downloading: apr-1.6.3.tar.bz2 [0.81MB]
  Estimated Download Time: 1m .....
  Downloading: apr-util-1.6.1.tar.bz2 [0.41MB]
  Estimated Download Time: 1m .....
  Approximate Install Size: 208 MB
  Estimated Install Time: 2 Minutes, 45 Seconds
  Installing Test @ 14:21:52

```

Figure 7: Installing the pts/apache

## pts/network-loopback

```

pts/network-loopback-1.0.3:
  Test Installation 1 of 1
  Approximate Install Size: 1 MB
  Estimated Install Time: 2 Seconds
  Installing Test @ 14:26:58

```

Figure 8: Installing the pts/network-loopback

## pts/john-the-ripper

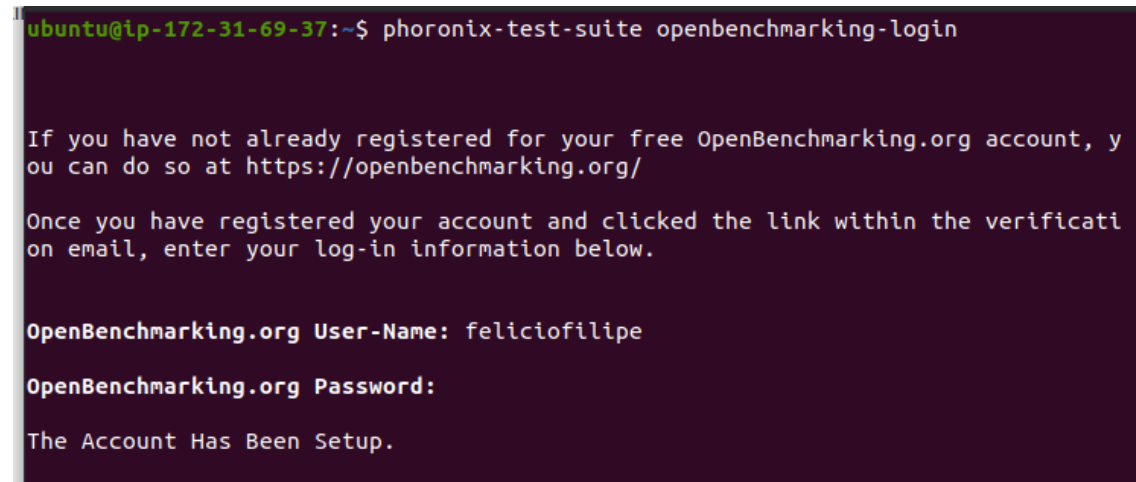
```

pts/john-the-ripper-1.7.2:
  Test Installation 1 of 1
  1 File Needed [42.25 MB]
  Downloading: john-1.9.0-jumbo-1.tar.gz [42.25MB]
  Downloading .....
  Approximate Install Size: 102 MB
  Estimated Install Time: 1 Minute, 32 Seconds
  Installing Test @ 14:27:29

```

Figure 9: Installing the pts/john-the-ripper

## 7) Login into the openbenchmarking.org account from the phoronix-test-suite command line



```
ubuntu@ip-172-31-69-37:~$ phoronix-test-suite openbenchmarking-login

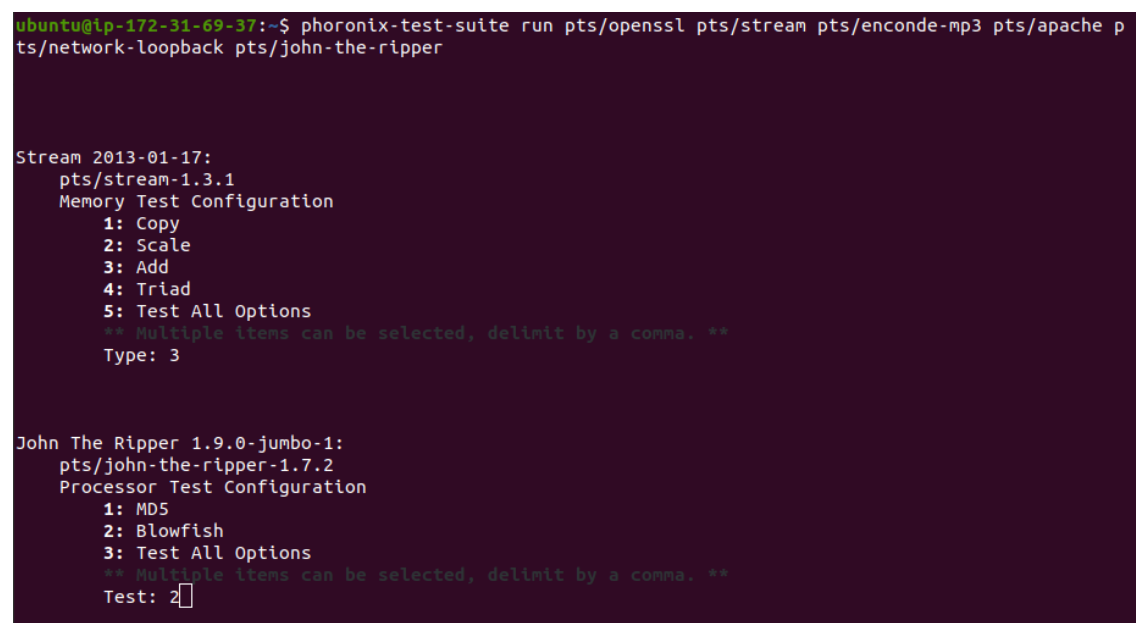
If you have not already registered for your free OpenBenchmarking.org account, you can do so at https://openbenchmarking.org/

Once you have registered your account and clicked the link within the verification email, enter your log-in information below.

OpenBenchmarking.org User-Name: feliciofilipe
OpenBenchmarking.org Password:
The Account Has Been Setup.
```

Figure 10: Logging into the openbenchmarking.org account from the phoronix-test-suite command line

## 8) Execute the 6 benchmarks



```
ubuntu@ip-172-31-69-37:~$ phoronix-test-suite run pts/openssl pts/stream pts/enconde-mp3 pts/apache pts/network-loopback pts/john-the-ripper

Stream 2013-01-17:
pts/stream-1.3.1
Memory Test Configuration
  1: Copy
  2: Scale
  3: Add
  4: Triad
  5: Test All Options
  ** Multiple items can be selected, delimit by a comma. **
Type: 3

John The Ripper 1.9.0-jumbo-1:
pts/john-the-ripper-1.7.2
Processor Test Configuration
  1: MD5
  2: Blowfish
  3: Test All Options
  ** Multiple items can be selected, delimit by a comma. **
Test: 2
```

Figure 11: Executing the 6 benchmarks

```
Would you like to save these test results (Y/n):
Enter a name for the result file: CC21Assignment2
Enter a unique name to describe this test run / configuration: t2.nano
```

Figure 12: Executing the 6 benchmarks

```
Would you like to upload the results to OpenBenchmarking.org (y/n):
Would you like to upload the results to OpenBenchmarking.org (y/n):
Would you like to upload the results to OpenBenchmarking.org (y/n): y
Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (y/n): y
Results Uploaded To: https://openbenchmarking.org/result/2104161-FELI-CC21ASS71
```

Figure 13: Executing the 6 benchmarks

## 9) Create an image

Instance ID  
i-01b80e065b7c2d14c

Image name  
Ubuntu\_With\_Phoronix

Maximum 127 characters. Can't be modified after creation.

Image description - optional  
Image description

Maximum 255 characters

No reboot  
☐ Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/s...	Create new snapshot fr...	8	EBS General Purpose SS...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately  
Tag the image and the snapshots with different tags.

No tags associated with the resource.

Add tag

You can add 50 more tags.

Cancel Create image

Figure 14: Creating an image

10) Launch the second VM - m5.large

AMI Details

Ubuntu\_With\_Phoronix - ami-07bb8c3c2ad600654

Root Device Type: ebs    Virtualization type: hvm

Edit AMI

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m5.large	-	2	8	EBS only	Yes	Up to 10 Gigabit

Edit instance type

Security Groups

Security group name

launch-wizard-3

Description

launch-wizard-3 created 2021-04-16T16:24:31.164+01:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Edit security groups

Instance Details

Edit instance details

Storage

Edit storage

Tags

Edit tags

Cancel

Previous

Launch

Figure 15: Launching the second VM - m5.large with an AMI

11) Execute the 6 benchmarks

```
ubuntu@ip-172-31-73-155:~$ phoronix-test-suite benchmark 2104161-FELI-CC21ASS71
```

Figure 16: Execute the 6 benchmarks with a single command: **phoronix-test-suite benchmark 2104161-FELI-CC21ASS71**

12) Launch the third VM - c5.xlarge

AMI Details

Ubuntu\_With\_Phoronix - ami-07bb8c3c2ad600654

Root Device Type: ebs    Virtualization type: hvm

Edit AMI

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
c5.xlarge	-	4	8	EBS only	Yes	Up to 10 Gigabit

Edit instance type

Security Groups

Security group name

launch-wizard-4

Description

launch-wizard-4 created 2021-04-16T16:51:34.111+01:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Edit security groups

Instance Details

Edit instance details

Cancel

Previous

Launch

Figure 17: Launching the second VM - c5.xlarge with an AMI

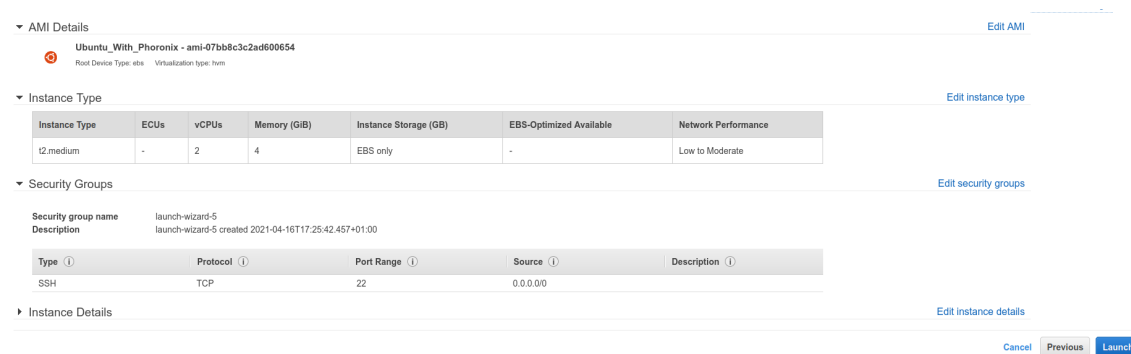
13) Execute the 6 benchmarks

```
ubuntu@ip-172-31-18-31:~$ phoronix-test-suite benchmark 2104161-FELI-CC21ASS71
```

Figure 18: Execute the 6 benchmarks with a single command: **phoronix-test-suite benchmark 2104161-FELI-CC21ASS71**



## 14) Launch the fourth VM - t2.medium



AMI Details: Ubuntu\_With\_Phoronix - ami-07bb8c3c2ad600654

Instance Type: t2.medium

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.medium	-	2	4	EBS only	-	Low to Moderate

Security Groups: launch-wizard-5

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Buttons: Cancel, Previous, Launch

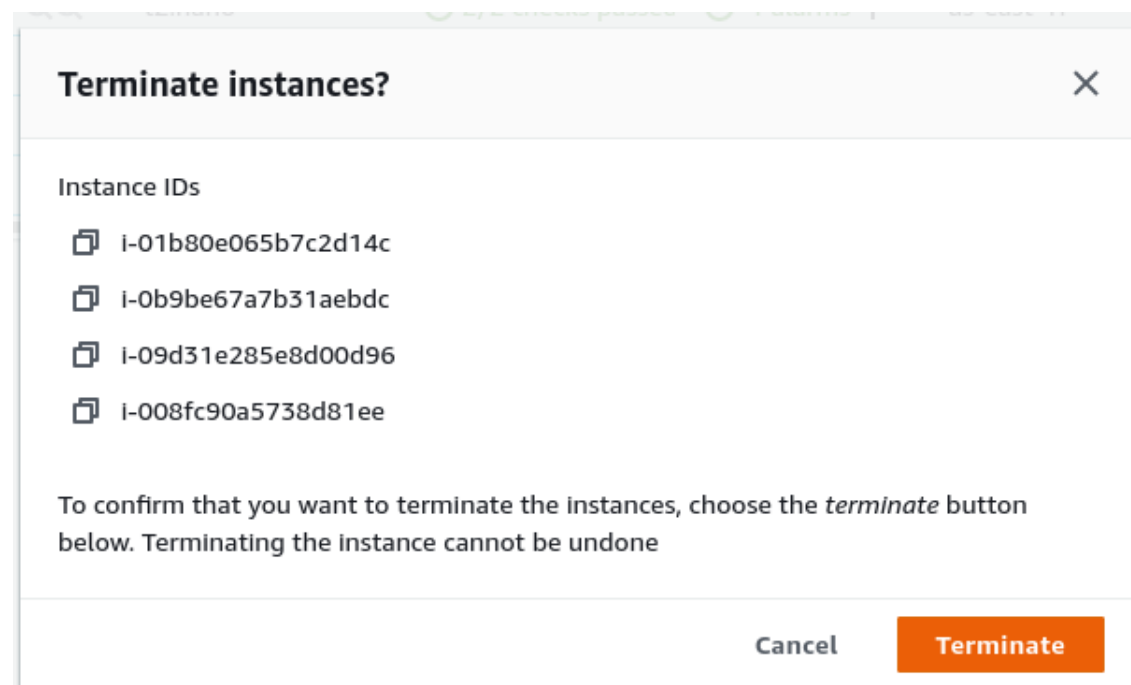
Figure 19: Launching the second VM - t2.medium with an AMI

## 15) Execute the 6 benchmarks

```
ubuntu@ip-172-31-71-180:~$ phoronix-test-suite benchmark 2104161-FELI-CC21ASS71
```

Figure 20: Execute the 6 benchmarks with a single command: **phoronix-test-suite benchmark 2104161-FELI-CC21ASS71**

## 16) Terminate all VMs



Terminate instances?

Instance IDs

- i-01b80e065b7c2d14c
- i-0b9be67a7b31aebdc
- i-09d31e285e8d00d96
- i-008fc90a5738d81ee

To confirm that you want to terminate the instances, choose the *terminate* button below. Terminating the instance cannot be undone

Buttons: Cancel, Terminate

Figure 21: Terminating all VMs

17) Deregister the AMI and delete its corresponding Snapshot

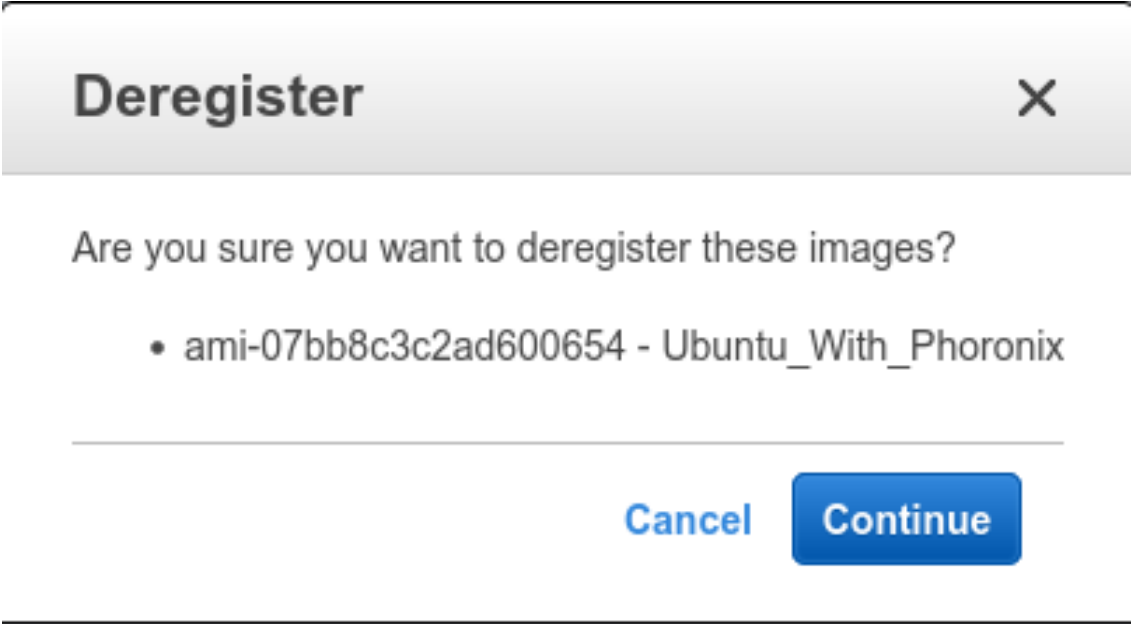


Figure 22: Deregistering the AMI and delete its corresponding Snapshot

Report

Results From feliciofilipe	
<div><div></div><div>CC21Assignment2</div><div>2 Systems - 4 Benchmark Results</div></div>	Intel Xeon E5-2686 v4 - Xen HVM domU - Intel 440FX 82441FX PMC Ubuntu 20.04 - 5.4.0-1038-aws - GCC 9.3.0
<div><div></div><div>CC21Assignment2</div><div>2 Systems - 4 Benchmark Results</div></div>	Intel Xeon E5-2676 v3 - Xen HVM domU - Intel 440FX 82441FX PMC Ubuntu 20.04 - 5.4.0-1038-aws - GCC 9.3.0
<div><div></div><div>CC21Assignment2</div><div>2 Systems - 4 Benchmark Results</div></div>	Intel Xeon E5-2676 v3 - Xen HVM domU - Intel 440FX 82441FX PMC Ubuntu 20.04 - 5.4.0-1038-aws - GCC 9.3.0
<div><div></div><div>CC21Assignment2</div><div>1 System - 4 Benchmark Results</div></div>	Intel Xeon E5-2676 v3 - Xen HVM domU - Intel 440FX 82441FX PMC Ubuntu 20.04 - 5.4.0-1038-aws - GCC 9.3.0

Figure 23

The link to your profile

<https://openbenchmarking.org/user/feliciofilipe>

Report on the selected instance types, and briefly explain why you choose them

The instance types that I selected are the ones in the following table, I have made this choices to accomplish a few key features to have good tests. First, I have choose one from each family has required. Second I choose different sizes (CPU and Memory) so I could compare the improvement

in performance to the rise in cost so I could determine which is the most efficient. I decided to use the largest on the C family so I could better visualise the impact of Compute Optimized.

Instance name ▼	On-Demand hourly rate ▼	vCPU ▼	Memory ▼	Storage ▼	Network performance ▼
c5.xlarge	\$0.17	4	8 GiB	EBS Only	Up to 10 Gigabit
m5.large	\$0.096	2	8 GiB	EBS Only	Up to 10 Gigabit
t2.medium	\$0.0464	2	4 GiB	EBS Only	Low to Moderate
t2.nano	\$0.0058	1	0.5 GiB	EBS Only	Low

Figure 24

## Report and comments the obtained results

First and foremost we need to get familiar with tests:

### Loopback TCP Network Performance

This test measures the loopback network adapter performance using a micro-benchmark to measure the TCP performance.

### John The Ripper

This is a benchmark of John The Ripper, which is a password cracker.

### OpenSSL

OpenSSL is an open-source toolkit that implements SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols. This test measures the RSA 4096-bit performance of OpenSSL.

### Apache Benchmark

This is a test of ab, which is the Apache benchmark program. This test profile measures how many requests per second a given system can sustain when carrying out 1,000,000 requests with 100 requests being carried out concurrently.

## Results


CC21Assignment2				
	t2.nano	t2.medium	m5.large	c5.xlarge
network-loopback: Time To Transfer 10GB Via Loopback	22.115	17.103	15.340	16.007
john-the-ripper: Blowfish	1002	2106	1446	3129
openssl: RSA 4096-bit Performance	121.4	246.9	235.7	512.5
apache: Static Web Page Serving	4928.24	8475.80	11589.55	19744.72

Figure 25

The only results that surprise me its in the performance of the t2.medium with the m5.large, considering that the second is more expensive and its reported has having better Network performance,

with the t2.medium having better performance in john-the-ripper (more 45,64%) and OpenSSL (more 4,75%).

Despite this, the m5.large has the best TPC performance (more 4,35% than the c5.xlarge).

### **Provide your own evaluation and analysis on the obtained relative cost-efficiency of the tested VMs**

Considering that:

- t2.medium is 80% more expensive than the t2.nano
- m5.large is 107% more expensive than the t2.medium
- c5.large is 77% more expensive than the c5.large
- t2.medium is 78,38% better performer overall on the tests than the t2.nano
- m5.large is 0,54% worst performer overall on the tests than the t2.medium
- c5.large is 74,96% better performer overall on the tests than the c5.large

If we compare the increases in price and in performance we can gather that the t2.medium has the better relative cost-efficiency.

## **Conclusion**

With this assignment I learned something completely new in this capability of evaluating the performance of VM's through phoronix-test-suite and openbenchmarking.org. This was particularly satisfying because I can imagine the utility of knowing these tools in my professional career.

Considering that we are going towards a world where product it's being replace by service, allowing a more efficient live adaptation of the resources, knowing how to test the different VM services and how to apply the correct benchmarks to evaluate their performance contrasting to the additional financial investment is a great tool to have as an engineer.