

# Report - Lab 2

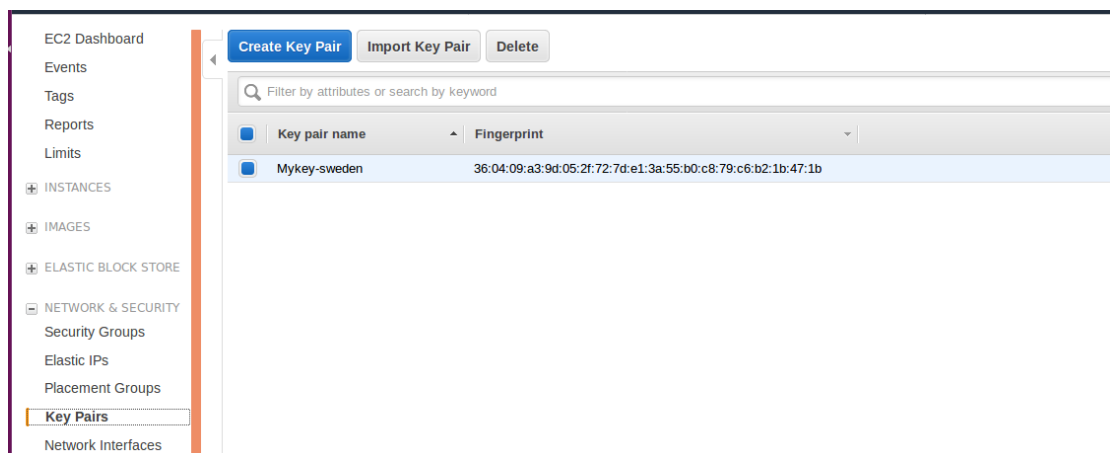
Group – 5

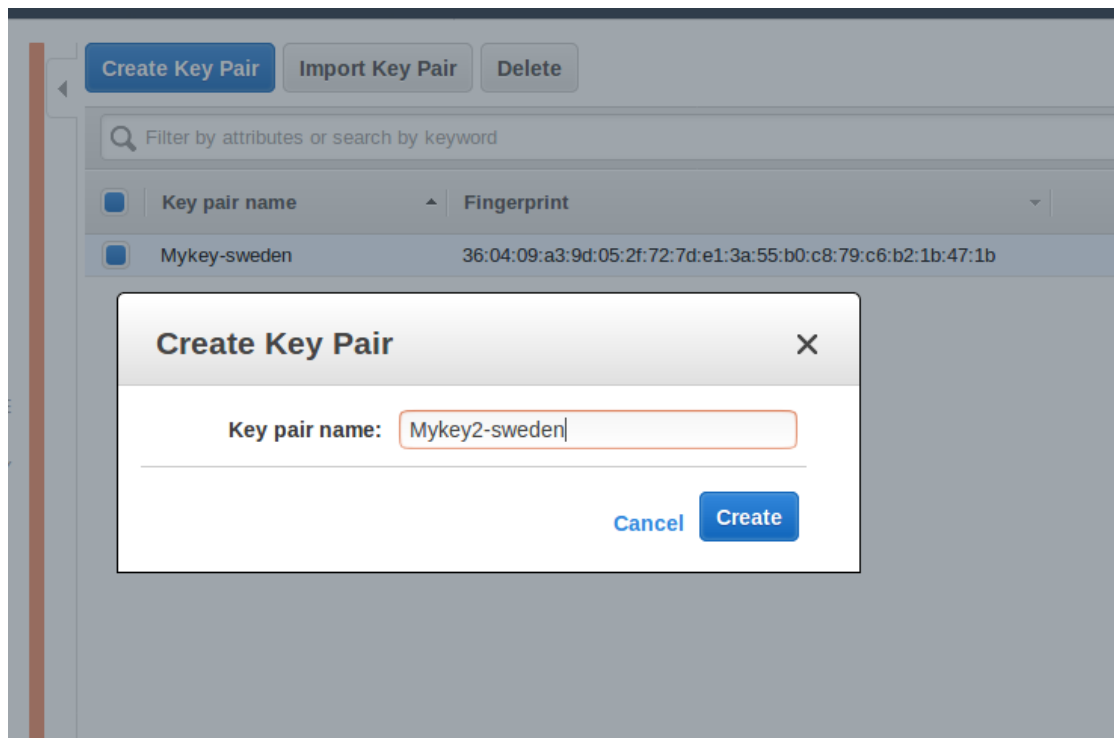
Name	E-mail	PNumber
Sri Harsha Arakatavemula	<a href="mailto:sraa16@student.bth.se">sraa16@student.bth.se</a>	9501036991
Suman Barua	subr@student.bth.se	8210133370
Narasimha Krishna Nannuri	nana18@student.bth.se	9208309535
Bing Li	bili18@student.bth.se	8911151242
Guojun Lai	gula18@student.bth.se	9706102572

Task 1 Setup and run an amazon EC2 instance

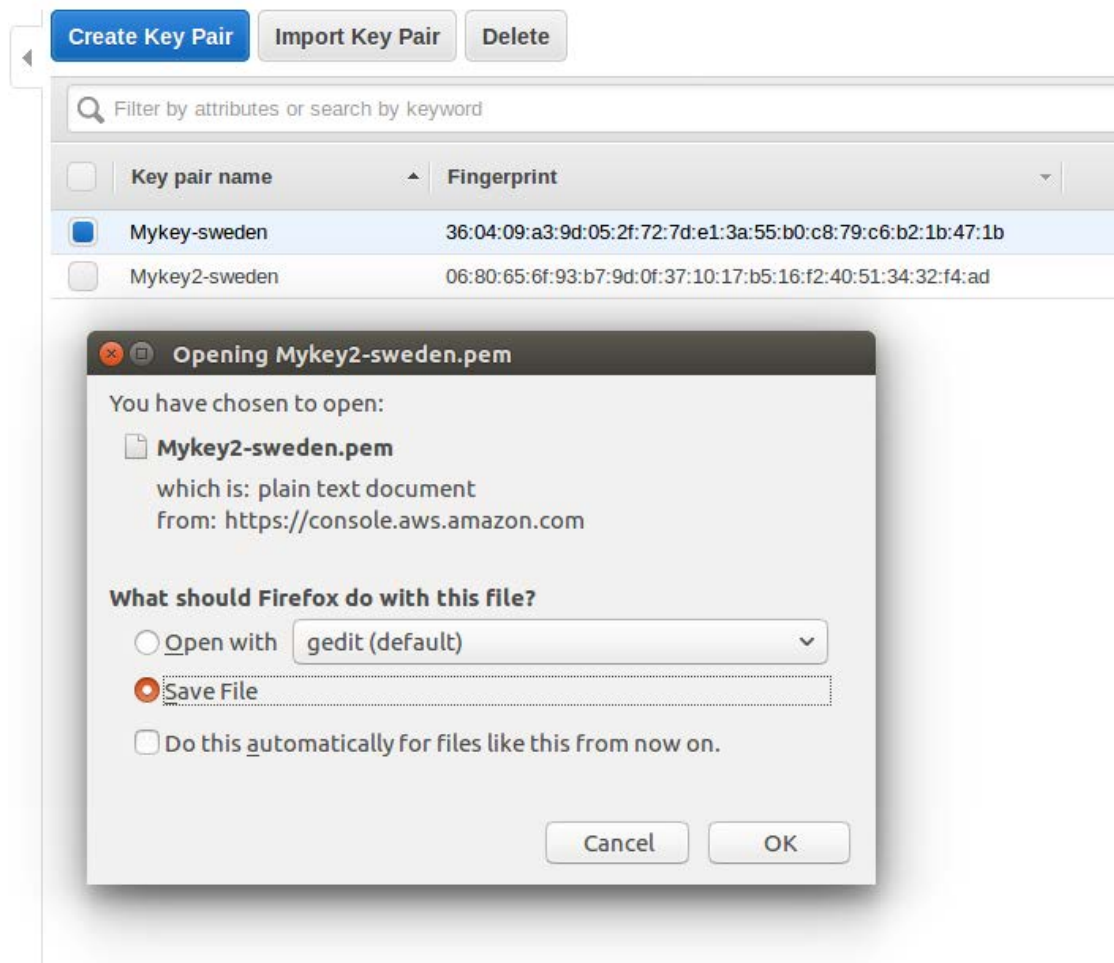
1. Create a key pair

a. In the navigation pane, under NETWORK & SECURITY, choose Key Pairs, then click Create Key Pair and enter the key pair name.





b. After clicking Create, the private key file is automatically downloaded by our browser.



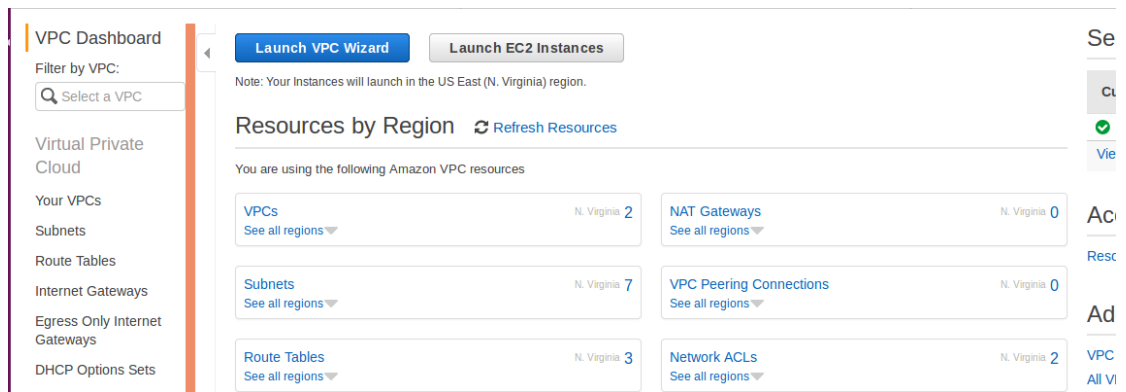
c. Set permission for the key by using the command below.

```
chmod 400 Mykey2-sweden.pem
```

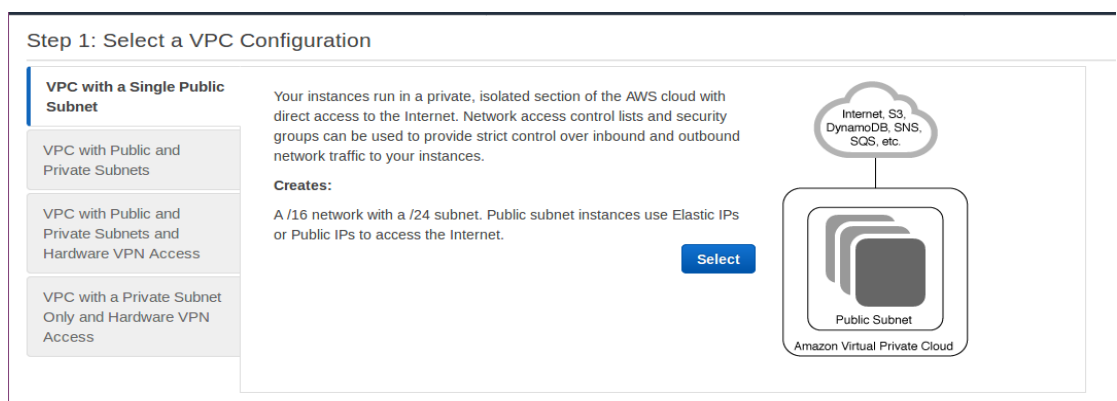
Without setting this permission, we cannot connect to our instance using this key pair.

## 2. Creating a nondefault Virtual Private Cloud (VPC)

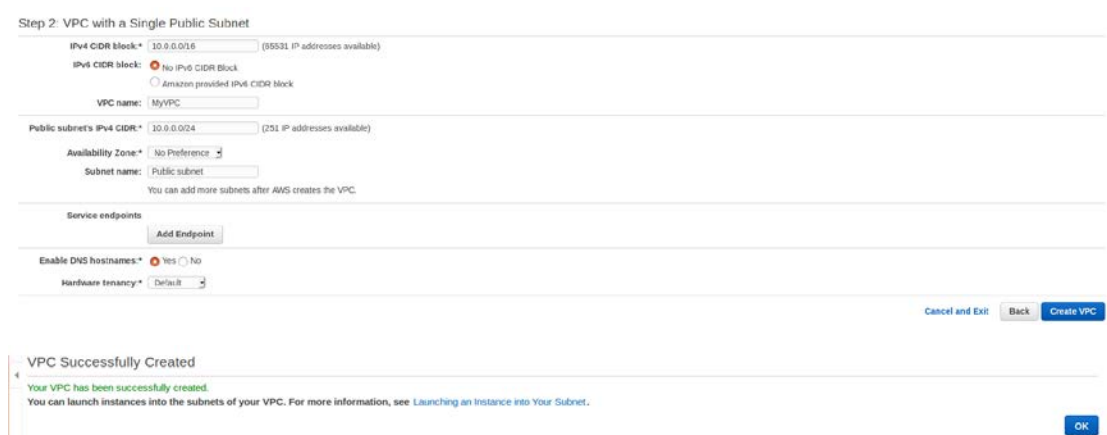
a. On the VPC dashboard, click Launch VPC Wizard



- b. On the Select a VPC Configuration page, ensure that VPC with a Single Public Subnet is selected, and choose Select.



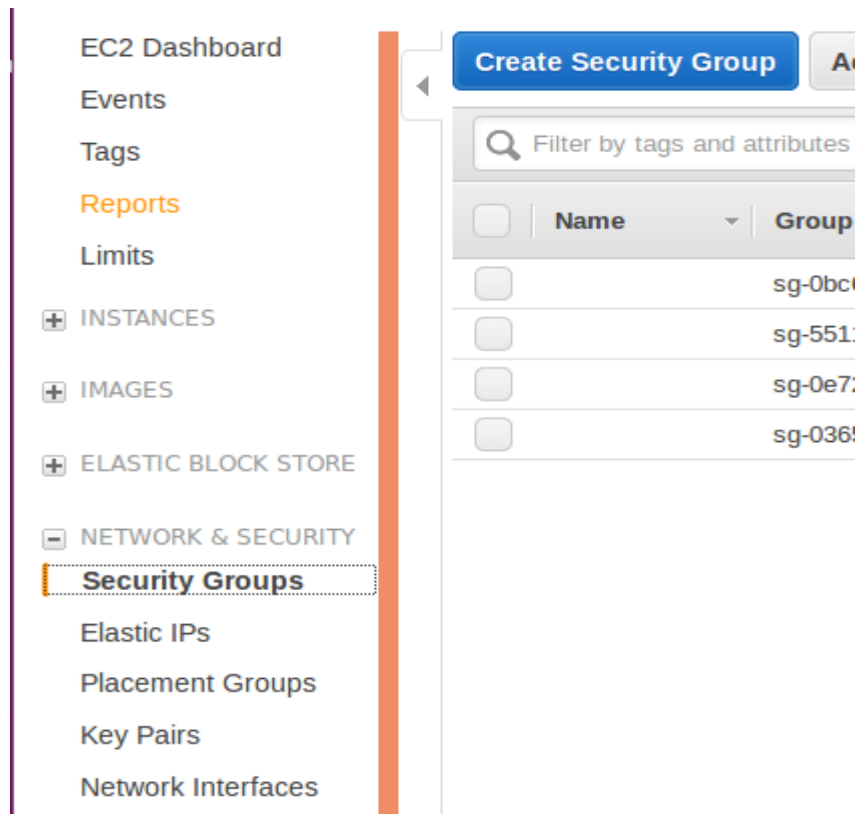
- c. On the VPC with a Single Public Subnet page, enter a friendly name for your VPC in the VPC name field. Leave the other default configuration settings, and click Create VPC. On the confirmation page, click OK.



Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	Tenancy	Default VPC
MyVPC	vpc-03a7a04e381921506	available	10.0.0.0/16	-	dopt-f560358e	rtb-07cfb0b2b9049a3ef	acl-0626beb3f6526022e	default	No
	vpc-6e7d2514	available	172.31.0.0/16	-	dopt-f560358e	rtb-d3e7d0ac	acl-09c1d213	default	Yes

### 3. Creating a security group

- a. In the navigation pane, under NETWORK & SECURITY, choose Security Groups, then click Create Security Groups.



- b. Configure the Security Group

Since we will connect the instance in the VPC from our computer, in the SSH row, we will choose My IP as the Source.

**Create Security Group** ✕

Security group name i

Description i

VPC i

Security group rules:

**Inbound** **Outbound**

Type <span>i</span>	Protocol <span>i</span>	Port Range <span>i</span>	Source <span>i</span>	Description <span>i</span>
SSH <span>v</span>	TCP	22	My IP <span>v</span> 80.78.218.217/32	e.g. SSH for Ad
HTTP <span>v</span>	TCP	80	Anywhere <span>v</span> 0.0.0.0/0, ::/0	e.g. SSH for Ad
HTTPS <span>v</span>	TCP	443	Anywhere <span>v</span> 0.0.0.0/0, ::/0	e.g. SSH for Ad

**Add Rule**

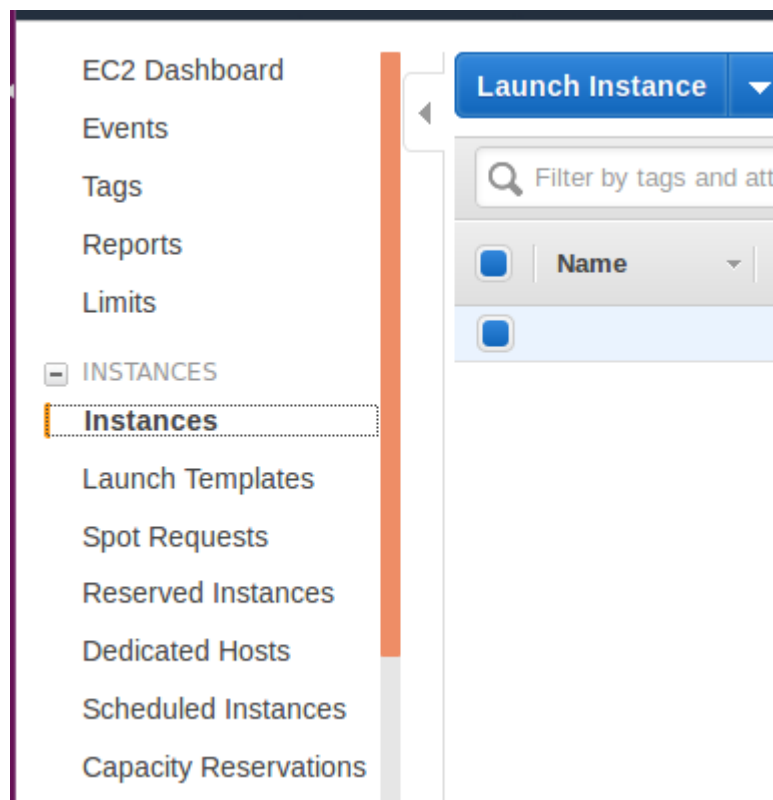
Cancel Create

---

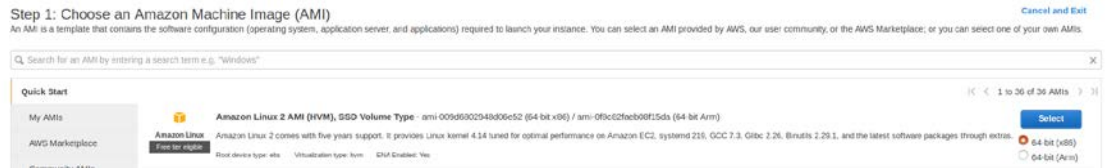
☐ sg-0365ebb17e2ab79e2 Mygroup2 vpc-6e7dd514 Mygroup2-sweden

#### 4. Launch an instance in VPC

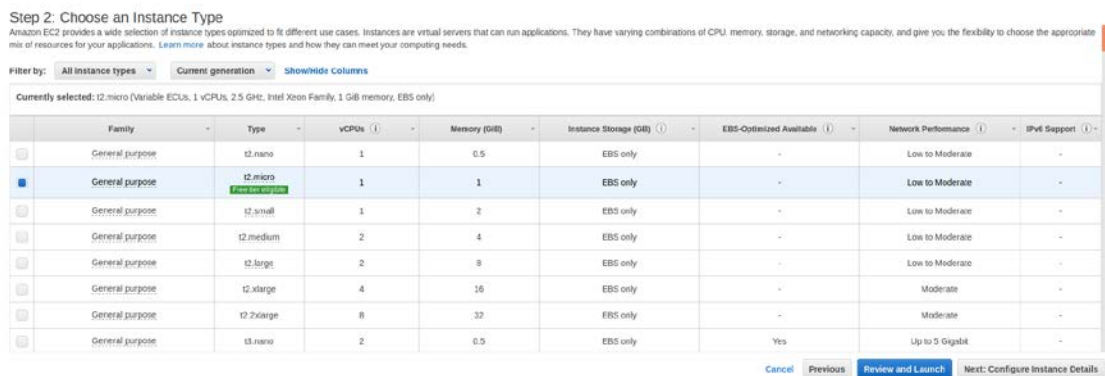
- From the console dashboard, click Launch Instance.



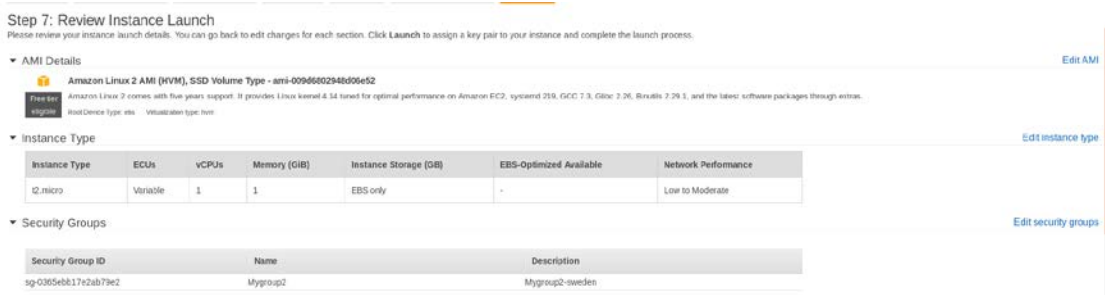
- On the The Choose an Amazon Machine Image (AMI) page, Select an HVM version of Amazon Linux 2.



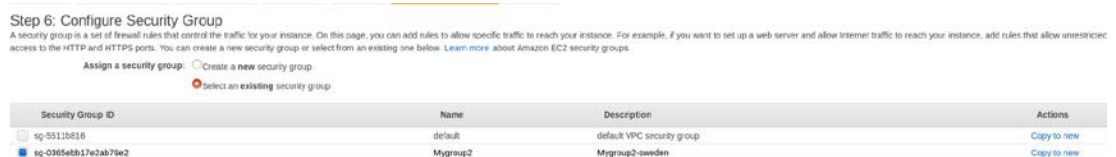
- c. On the Choose an Instance Type page, we select the t2.micro type, since this instance type is eligible for the free tier.



- d. Click Review and Launch to let the wizard complete the other configuration settings.



- e. On the Configure Security Group, we can select the security group that we created before.



- f. On the Review Instance Launch page, click Launch.

**Step 7: Review Instance Launch**  
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**AMI Details** Edit AMI

**Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-009d6802348d96452**  
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.  
Root Device Type: ebs    Virtualization type: hvm

**Instance Type** Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

**Security Groups** Edit security groups

Security Group ID	Name	Description
sg-0305e0b17e2ab79e2	Mygroup2	Mygroup2-sweden

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description

Cancel Previous Launch

- g. Select Choose an existing key pair, then select the key pair that we created before.

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

Mykey-sweden

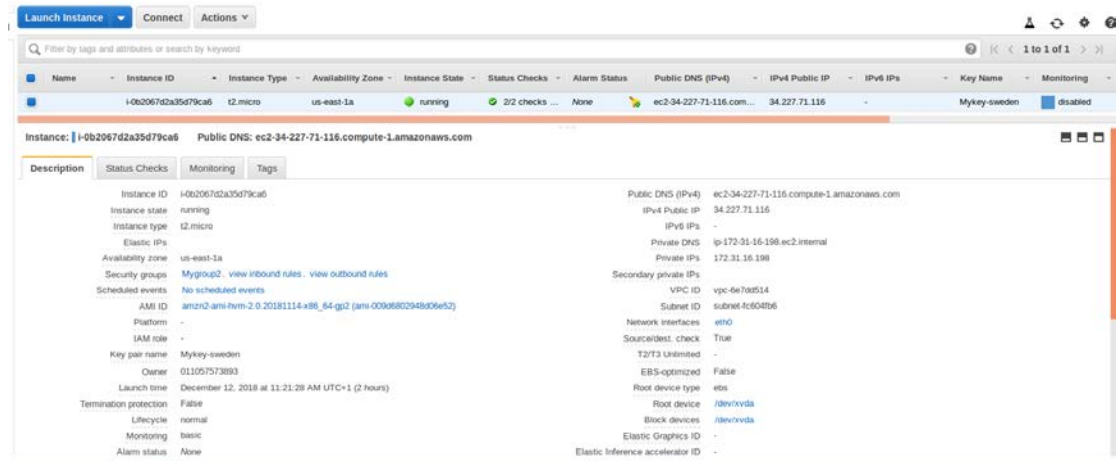
☒ I acknowledge that I have access to the selected private key file (Mykey-sweden.pem), and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

- h. On the Instances screen, we can view the status of the launch.





## 5. Connect to the instance

Use the ssh command to connect to the instance. We specify the private key (.pem) file and user\_name@public\_dns\_name.

The private key file:

```
/home/li/Mykey-sweden.pem
```

The public\_dns\_name:

```
ec2-34-227-71-116.compute-1.amazonaws.com
```

Since we use Amazon Linux 2 AMI, the user name is ec2-user.

The ssh command are below:

```
li@li-virtual-machine:~$ ssh -i /home/li/Mykey-sweden.pem ec2-user@ec2-34-227-71-116.compute-1.amazonaws.com
```

We connect the instance successfully.

```
li@li-virtual-machine:~$ ssh -i /home/li/Mykey-sweden.pem ec2-user@ec2-34-227-71-116.compute-1.amazonaws.com
Last login: Wed Dec 12 10:40:32 2018 from 80.78.218.217

  _ _ _ _ _
 _/   ( _/   \
 _/_ _/   _/_  Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 5 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-16-198 ~]$
```

## Task 2 Monitoring VMs

### 1. Generate load with sysbench to CPU

#### a. Install sysbench in the EC2 instance

```
ubuntu@ip-172-31-41-59:~$ sudo apt-get update
```

```
ubuntu@ip-172-31-41-59:~$ sudo apt-get install sysbench
```

#### b. check that sysbench works

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=cpu --cpu-max-prime=200000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line.
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 200000
Initializing worker threads...

Threads started!

CPU speed:
  events per second:      13.96

General statistics:
  total time:              10.0280s
  total number of events:  140

Latency (ms):
  min:                     69.89
  avg:                     71.62
  max:                     72.91
  95th percentile:        71.83
  sum:                     10026.71

Threads fairness:
  events (avg/stddev):      140.0000/0.00
  execution time (avg/stddev): 10.0267/0.00
```

#### c. Do a more intense CPU stress test with the following commands.

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=1 run
```

```
total time: 10.0364s
```

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=2 run
total time: 10.0377s
```

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=4 run
total time: 10.1149s
```

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=8 run
total time: 10.3193s
```

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=16 run
total time: 10.3363s
```

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=32 run
total time: 11.4607s
```

```
sudo sysbench --test=cpu --cpu-max-prime=200000 --num-threads=64 run
total time: 13.7506s
```

SysBench runs a specified number of threads and they all execute requests in parallel. In this task, when testing the cpu, we set the number of threads to 1, 2, 4, 8, 16, 32, 64. Then we can get an impression about system performance from the value “total time”.

## 2. Generate load with sysbench to Disk

- a. Create the file set with following command

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=fileio --file-total-size=6G --file-num=3 prepare
```

We will get three test files, and their total size is 6GB.

- b. Run the random read and write test for 5 times.

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=fileio --file-total-size=6G --file-test-mode=rndwr --max-time=240 --max-requests=0 --file-block-size=4k --file-num=3 --num-threads=1 run
```

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=fileio --file-total-size=6G --file-test-mode=rndwr  
--max-time=240 --max-requests=0 --file-block-size=4k --file-num=3 --num-threads=4 run
```

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=fileio --file-total-size=6G --file-test-mode=rndwr  
--max-time=240 --max-requests=0 --file-block-size=4k --file-num=3 --num-threads=8 run
```

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=fileio --file-total-size=6G --file-test-mode=rndwr  
--max-time=240 --max-requests=0 --file-block-size=4k --file-num=3 --num-threads=16 run
```

```
ubuntu@ip-172-31-41-59:~$ sudo sysbench --test=fileio --file-total-size=6G --file-test-mode=rndwr  
--max-time=240 --max-requests=0 --file-block-size=4k --file-num=3 --num-threads=32 run
```

In this task, we will test random read and write of the disk with that three test files, and we set the number of threads to 1, 4, 8, 16, 32. Then we can get an impression about performance of IOPS of the disk from the values below.

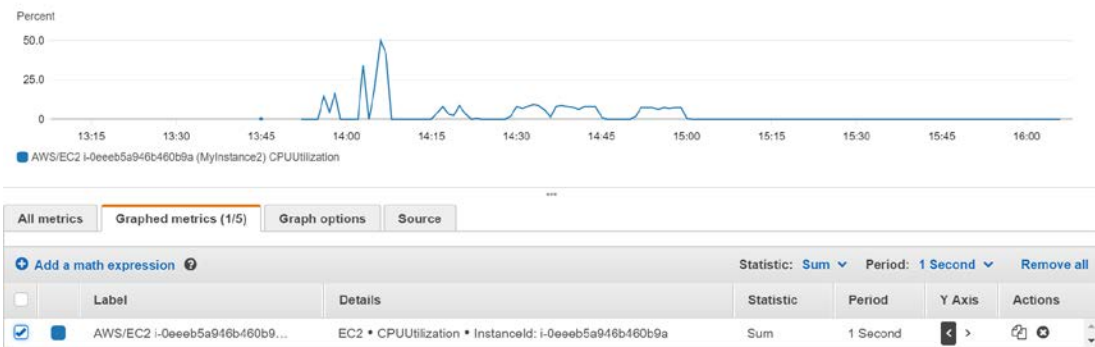
```
Operations:  
reads/s: 0.00  
writes/s: 3041.45
```

3. After some intense CPU and Disk stress tests, monitor the EC2 instance and customize dashboard with metrics CPUUtilization, VolumeReadBytes, VolumeWriteOps, VolumeReadOps, VolumeWriteBytes.

Since t2.micro uses ESB storage, we have to select CPU utilization from EC2 metrics and Disk related metrics from EBS metrics.

a. EC2/ CPUUtilization

This metric identifies the processing power required to run an application upon the EC2 instance.



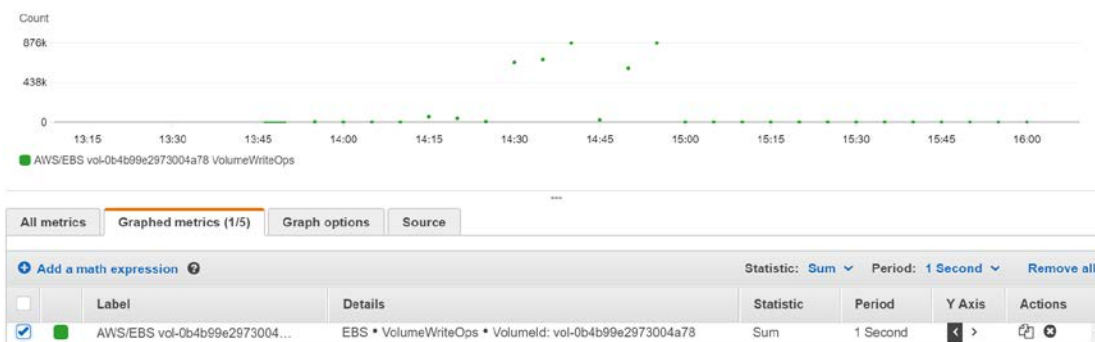
## b. EBS/ VolumeReadBytes

Provides information on the read operations in the specified period of time of generating load. The Sum statistic reports the total number of bytes transferred during the period.



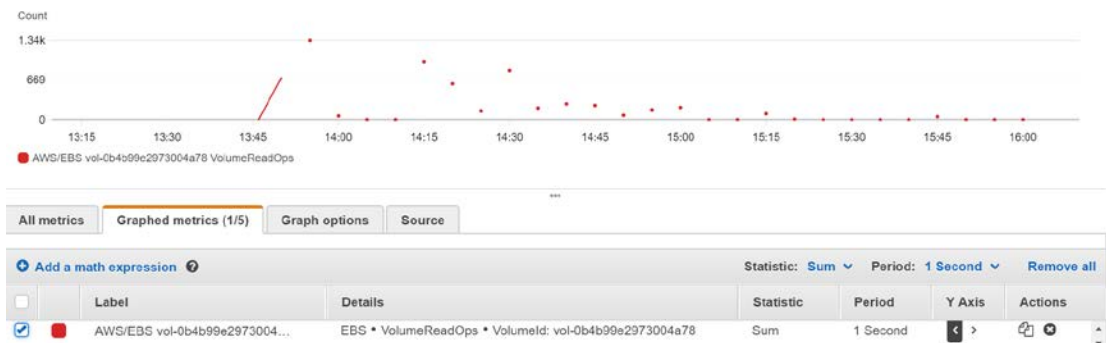
## c. EBS/ VolumeWriteOps

Provide the total number of write operations in the specified period of time of generating load.



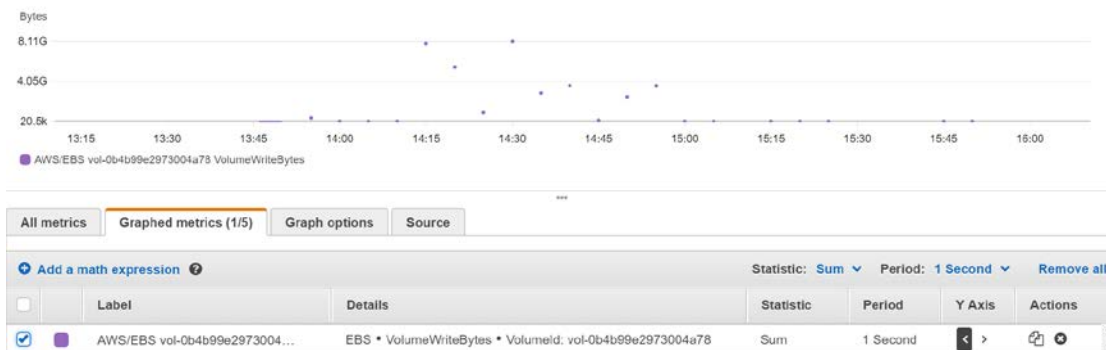
#### d. EBS/ VolumeReadOps

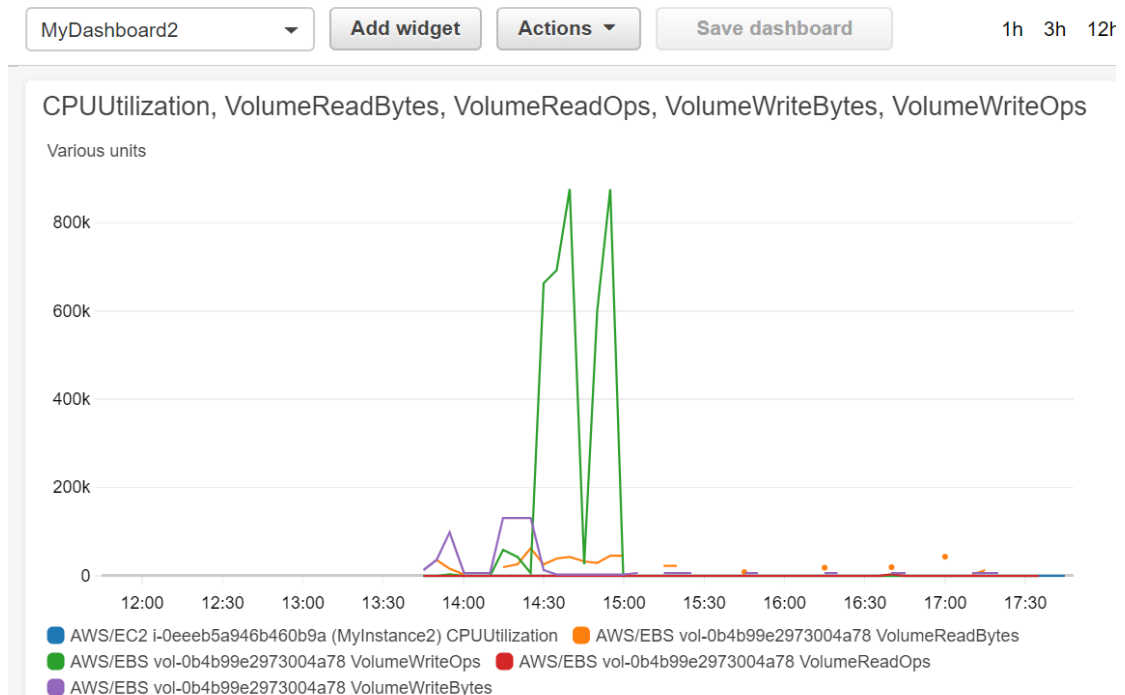
Provide the total number of read operations in the specified period of time of generating load.



#### e. EBS/ VolumeWriteBytes

Provides information on the write operations in the specified period of time of generating load. The Sum statistic reports the total number of bytes transferred during the period.





After generating the load from CPU and disk in the EC2 instance, we could monitor the whole result above from the CloudWatch.

### Task 3 Communication

1. Launch VM1 and VM2 in the nondefault VPC of task 1

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input type="checkbox"/>	VM1	i-0060f8bb18e97aeed	t2.micro	us-east-1d	running	2/2 checks ...
<input type="checkbox"/>	VM2	i-03e2c8c2bc6222650	t2.micro	us-east-1d	running	2/2 checks ...

2. Generate load on VM1 with the Apachebench running on VM2

- a. Install a web server apache2 on VM1 and apache2-utils on VM2 with the following commands

```
ubuntu@ip-172-31-83-56:~$ sudo apt-get install apache2
```

```
ubuntu@ip-172-31-89-101:~$ sudo apt-get install apache2-utils
```

- b. Stress the apache httpd on VM1 with the following

command on VM2

```
ubuntu@ip-172-31-89-101:~$ ab -n 200000 -c 60 http://ip-172-31-83-56.ec2.internal/index.html
```

This command means that we send 200000 http requests, generated from 60 concurrent users, to the web server hosted on the VM1 with its private IP address 172.31.83.56 and with internal hostname "ip-172-31-83-56.ec2.internal".

3. After an intense apache httpd stress test on VM1, monitor the VM1 instance and customize dashboard with metrics CPUUtilization, NetworkIn, NetworkOut, NetworkPacketsIn, NetworkPacketsOut, MemoryUsed, MemoryUtilization, MemoryAvailable, VolumeWriteOps, VolumeReadOps, VolumeWriteBytes, VolumeReadBytes

#### a. EC2/ CPUUtilization

This metric identifies the processing power required to run the web server apache upon VM1.

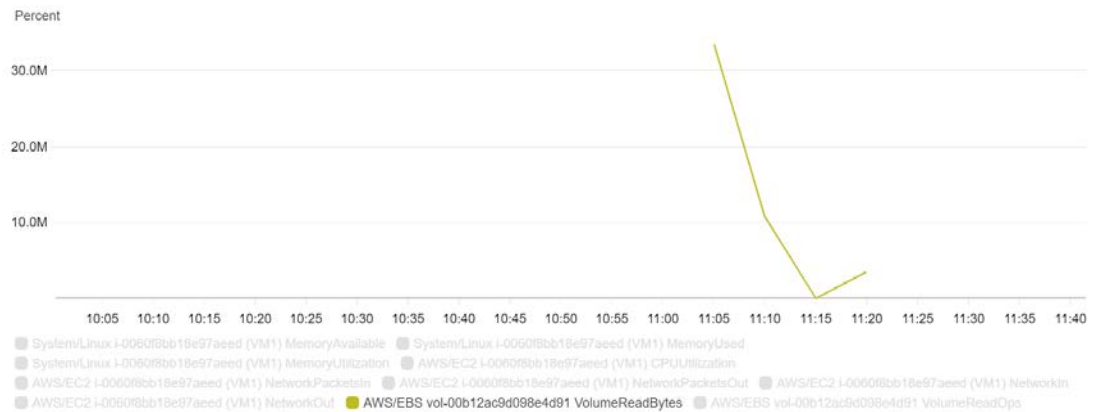


#### b. EBS/ VolumeReadBytes

Provides information on the read operations in the specified

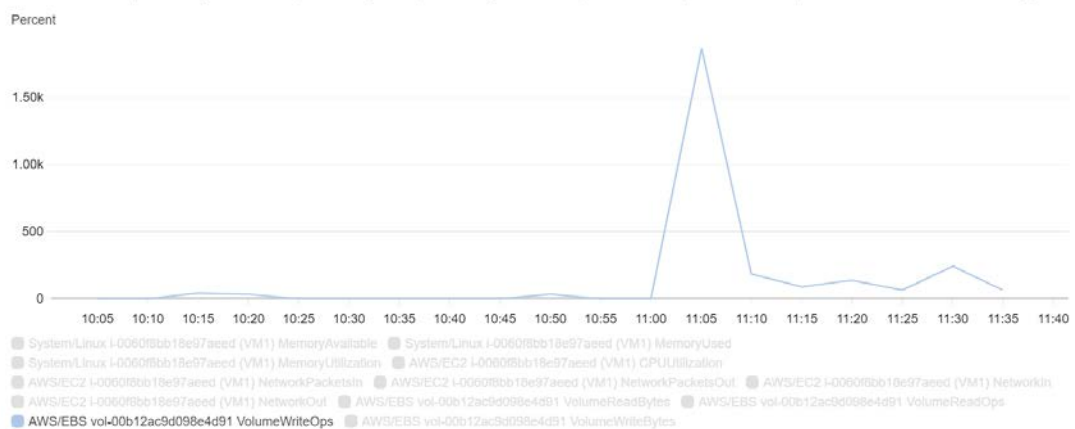


period of time of generating load. The Sum statistic reports the total number of bytes transferred during the period.



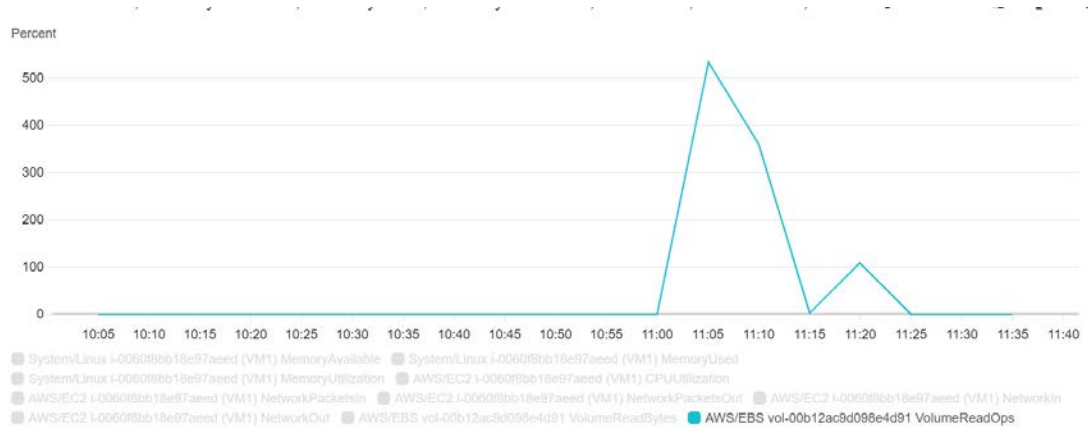
### c. EBS/ VolumeWriteOps

Provide the total number of write operations in the specified period of time of generating load.



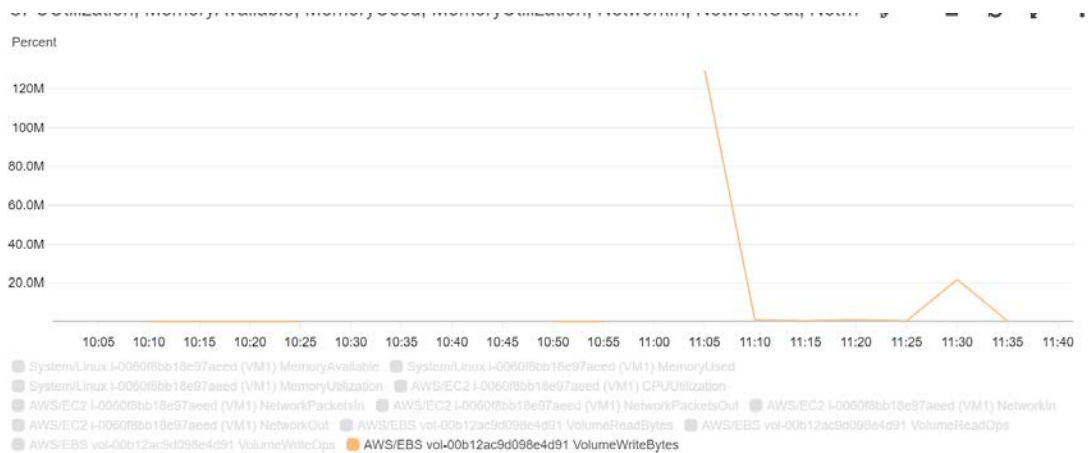
### d. EBS/ VolumeReadOps

Provide the total number of read operations in the specified period of time of generating load.



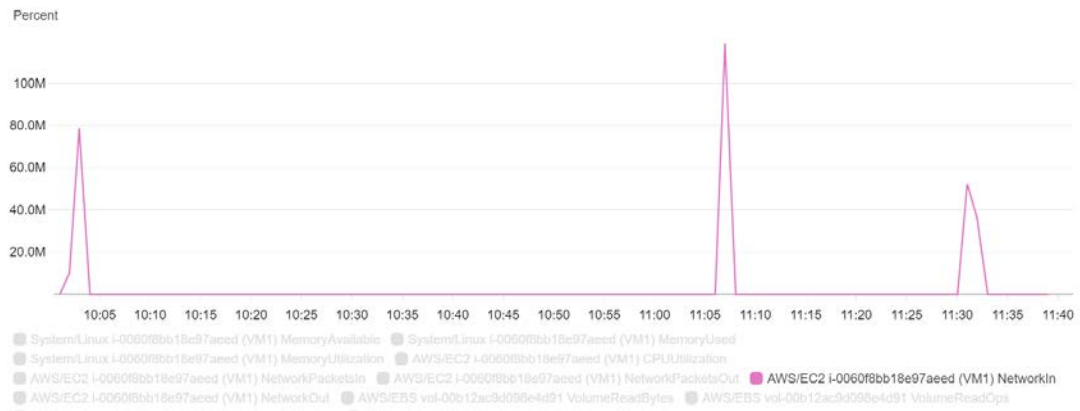
#### e. EBS/ VolumeWriteBytes

Provides information on the write operations in the specified period of time of generating load. The Sum statistic reports the total number of bytes transferred during the period.



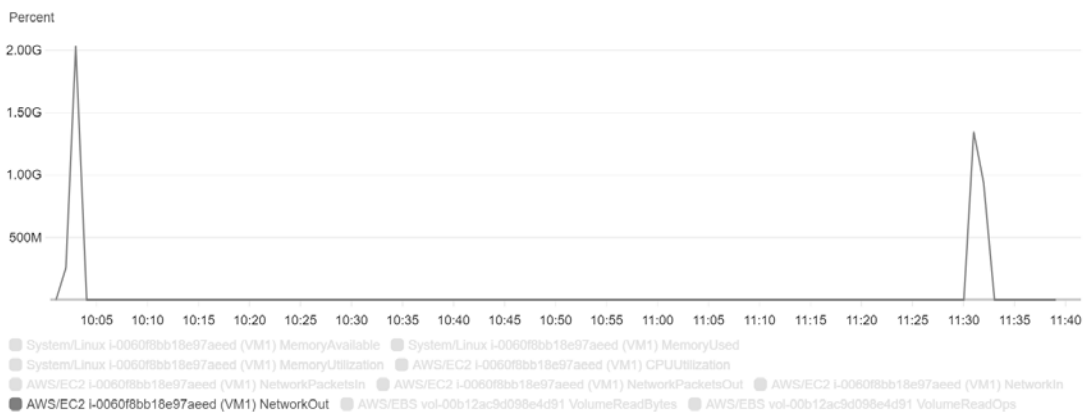
#### f. EC/NetworkIn

The number of bytes received on all network interfaces by the instance VM1. This metric identifies the volume of incoming network traffic to the instance VM1.



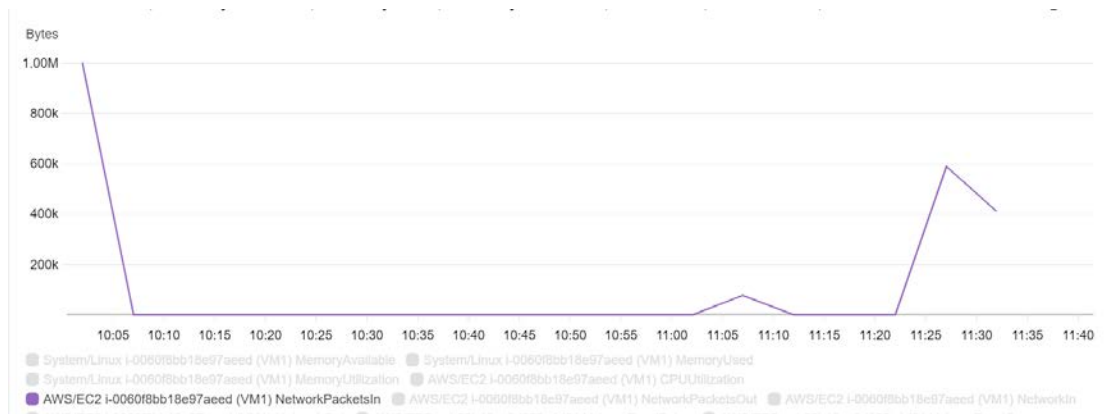
### g. EC/NetworkOut

The number of bytes sent out on all network interfaces by the instance VM1. This metric identifies the volume of outgoing network traffic from the instance VM1.



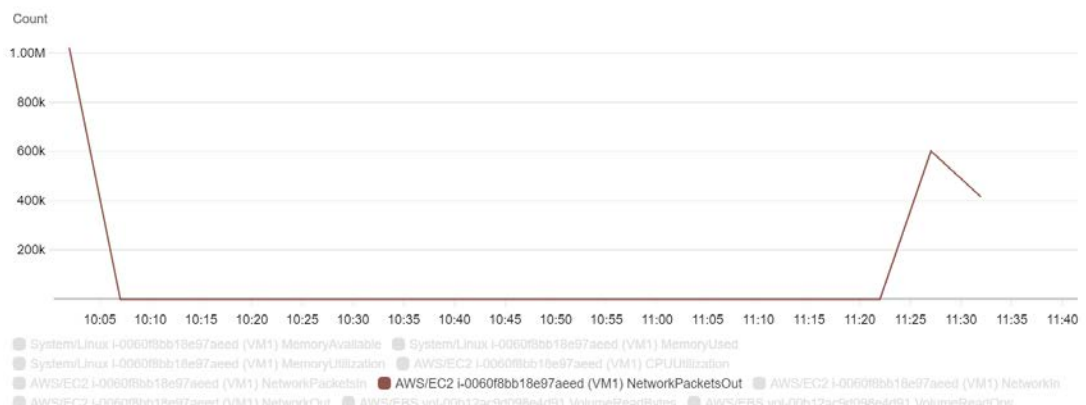
### h. EC/NetworkPacketsIn

The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on the instance VM1.



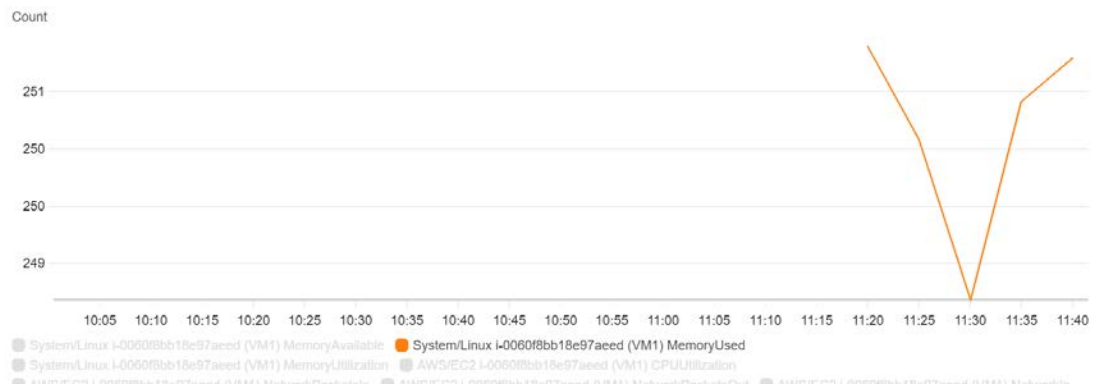
#### i. EC/NetworkPacketsOut

The number of packets sent out on all network interfaces by the instance. This metric identifies the volume of outgoing traffic in terms of the number of packets on the instance VM1.



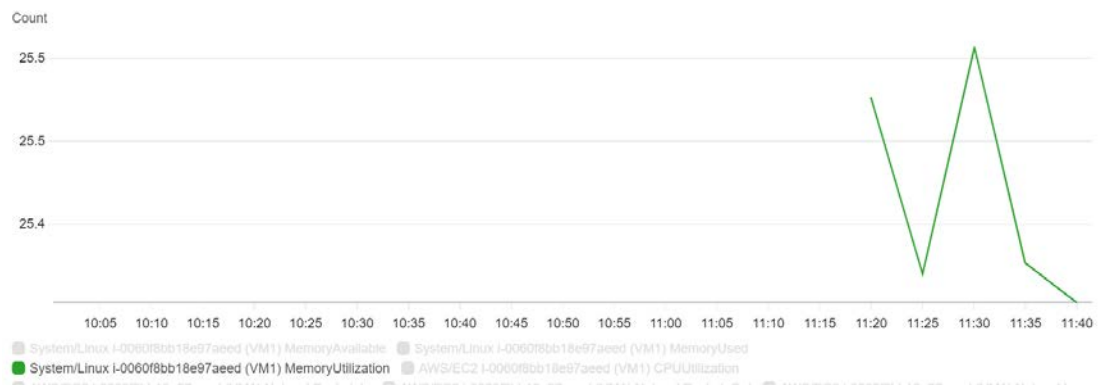
#### j. Linux System/ MemoryUsed

This metric counts memory allocated by the web server apache and the operating system Linux as used.



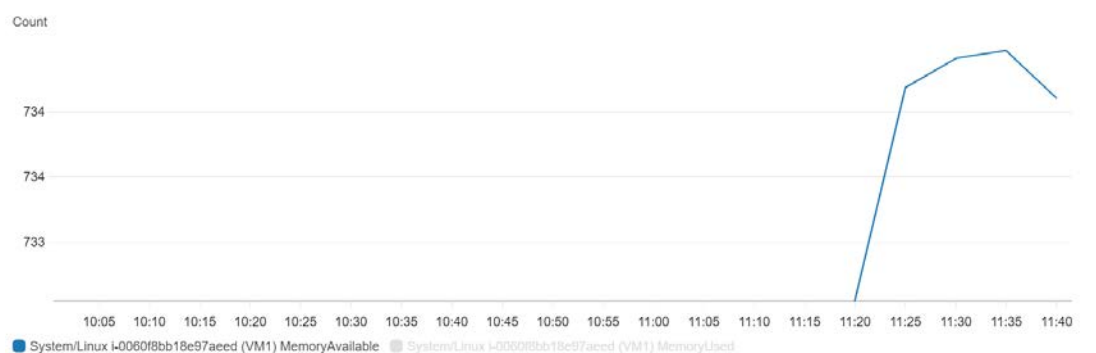
#### k. Linux System/ MemoryUtilization

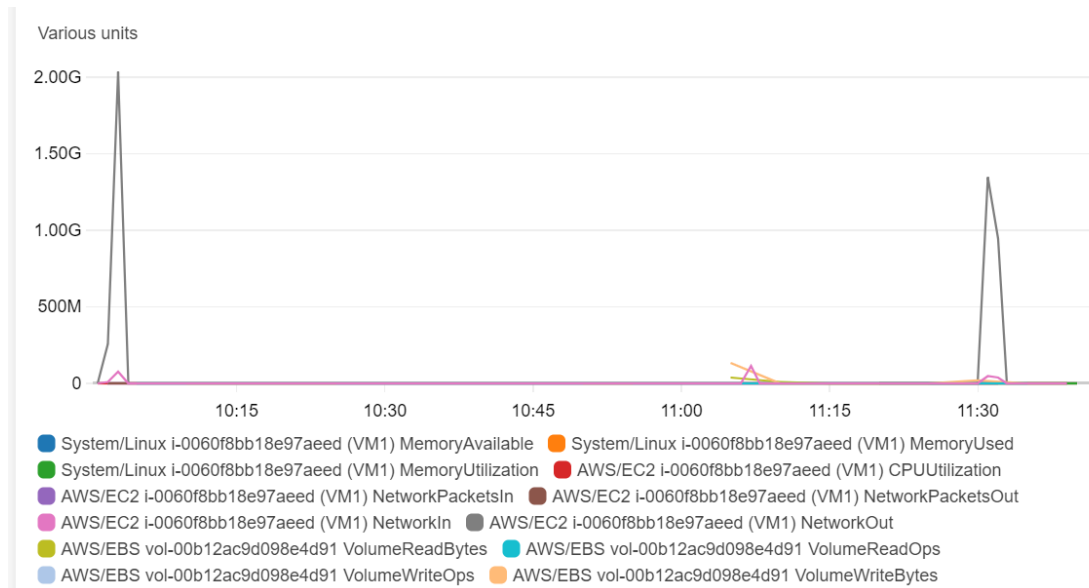
This metric counts memory allocated by the web server apache and the operating system Linux as used.



#### l. Linux System/ MemoryAvailable

This metric counts memory allocated by the web server apache and the operating system Linux as used.





After generating the load from the web server apache on the instance VM1, we could monitor the whole result above from the CloudWatch.