Introduction to Cloud Computing (DV1566)

Laboratory 2: Monitoring VM performance counters

Group – 18: -

Ananya Reddivari.

Harika Kosuru.

Monica Gattupalli.

Sri Sai Ganesh Satyadeva Naidu Totakura.

Syam Kumar Vemana.

TASK-1: -

❖ Creation of Security group: -

- To a create a security group we need to select "Security Group" in EC2 management console, on entering the security group name, VPC, Inbound rule and Outbound rule a security group with given credentials will be created.

- Configurations of our Security Group: -
  Security group name
  HMSSSECURITYGROUP

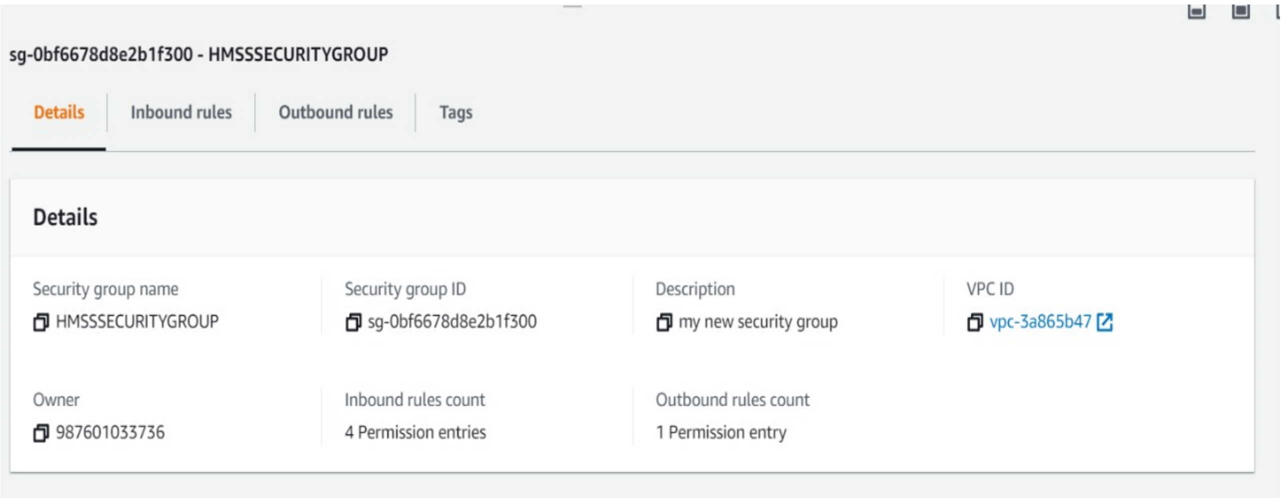
  Security group ID
  sg-0bf6678d8e2b1f300

  Description
  My new security group

  VPC ID
  vpc-3a865b47

  Inbound rules
  HTTP – 0.0.0.0/0, ::/0
  SSH – 0.0.0.0/0, ::/0

  Outbound rules
  All traffic – 0.0.0.0/0

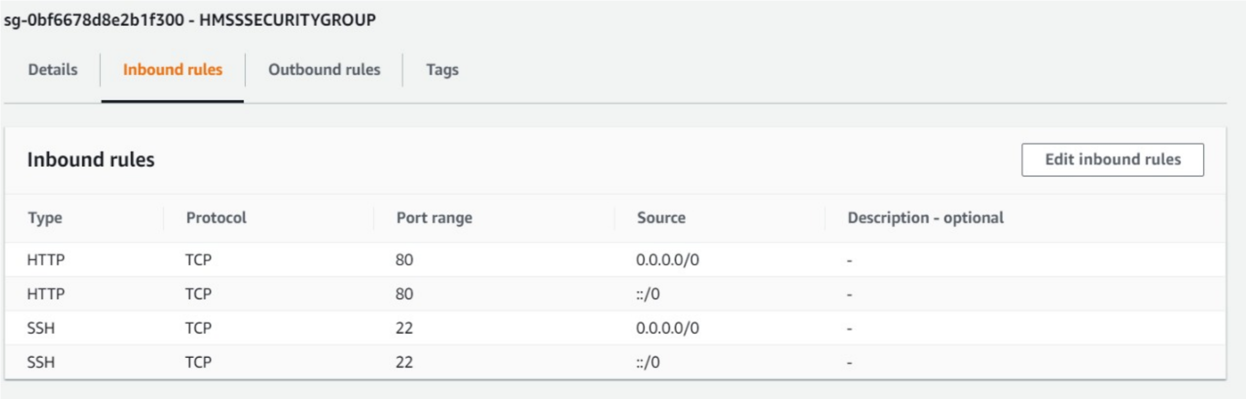


sg-0bf6678d8e2b1f300 - HMSSSECURITYGROUP

Details

Security group name	Security group ID	Description	VPC ID
HMSSSECURITYGROUP	sg-0bf6678d8e2b1f300	my new security group	vpc-3a865b47
Owner	**Inbound rules count**	**Outbound rules count**	
987601033736	4 Permission entries	1 Permission entry	

Figure 1 - Security group details.

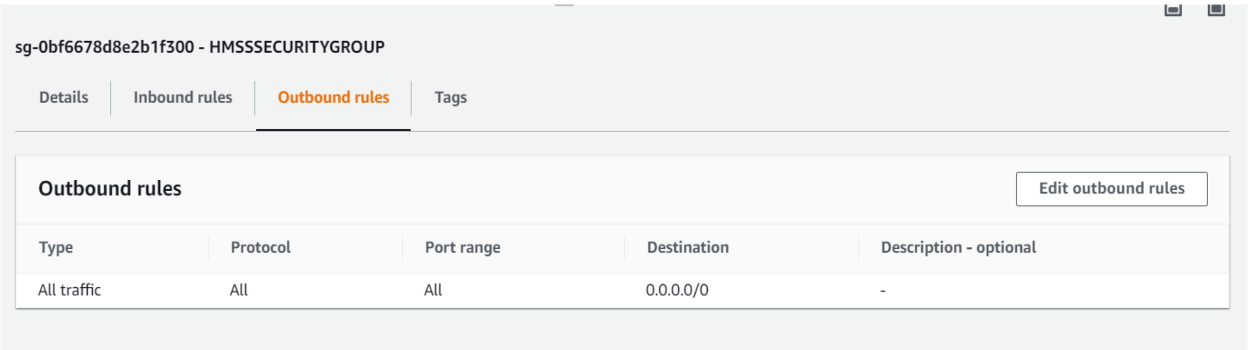


sg-0bf6678d8e2b1f300 - HMSSSECURITYGROUP

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-

Figure 2 - Inbound rules.



sg-0bf6678d8e2b1f300 - HMSSSECURITYGROUP

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

Figure 3 - Outbound rules.

Motivation: - We used HTTP including with SSH in Inbound rules because of better web services. We used All traffic rules in Out bound rules as we don’t know which web service will be used at a particular time.

❖ Creation of VPC: -

- We need to complete four steps to create a VPC.
  Step-1: - Enter configurations of our VPC.
  Step-2: - Creation of subnet.
  Step-3: - Creation of Route tables.
  Step-4: - Creation of internet gateway.

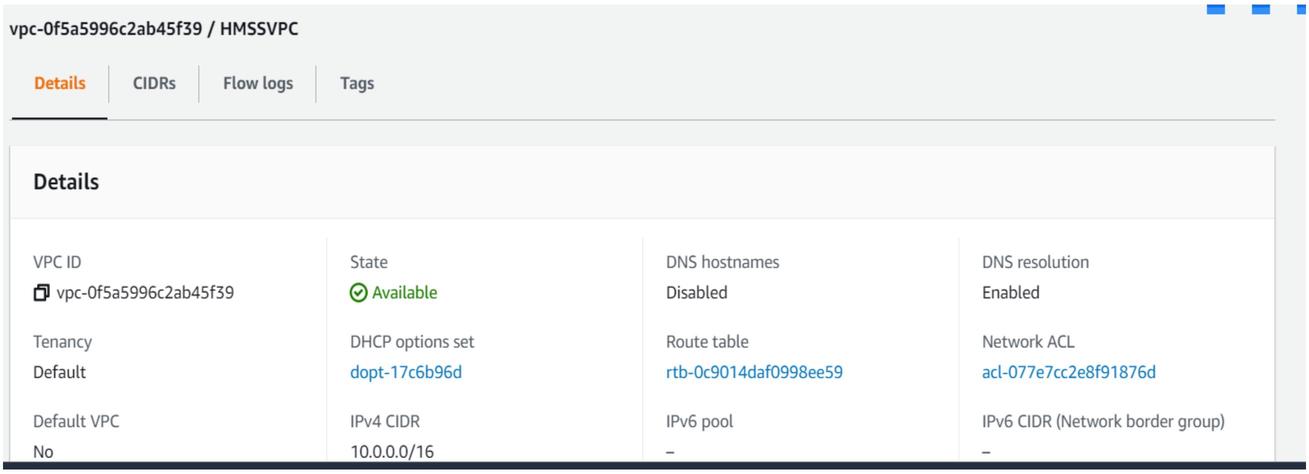
- Step-1: - Configurations of our VPC
  VPC ID
  vpc-0f5a5996c2ab45f39

  State
  Available

  Tenancy
  Default

Route table
rtb-0c9014daf0998ee59

Network ACL
acl-077e7cc2e8f91876d



vpc-0f5a5996c2ab45f39 / HMSSVPC

Details

VPC ID	State	DNS hostnames	DNS resolution
vpc-0f5a5996c2ab45f39	Available	Disabled	Enabled
Tenancy	DHCP options set	Route table	Network ACL
Default	dopt-17c6b96d	rtb-0c9014daf0998ee59	acl-077e7cc2e8f91876d
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	10.0.0.0/16	–	–

Figure 4 - Configuration of VPC.

- Step-2: - Creation of Subnet
  Subnet ID
  subnet-0cc08e53e0c03f7ca

  State
  Available

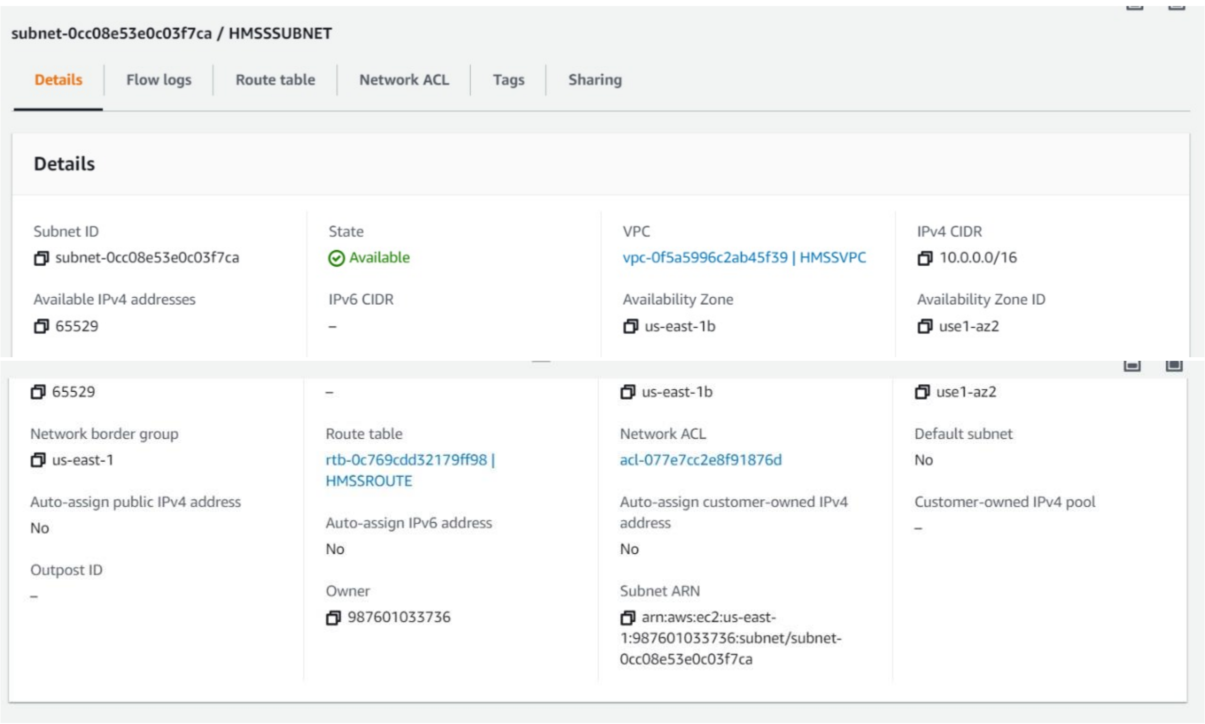
  VPC
  vpc-0f5a5996c2ab45f39 | HMSSVPC

  Available IP address
  65529

  Route table
  rtb-0c769cdd32179ff98 | HMSSROUTE

  Network ACL
  acl-077e7cc2e8f91876d

  Subnet ARN
  arn:aws:ec2:us-east-1:987601033736:subnet/subnet-0cc08e53e0c03f7ca



subnet-0cc08e53e0c03f7ca / HMSSSUBNET

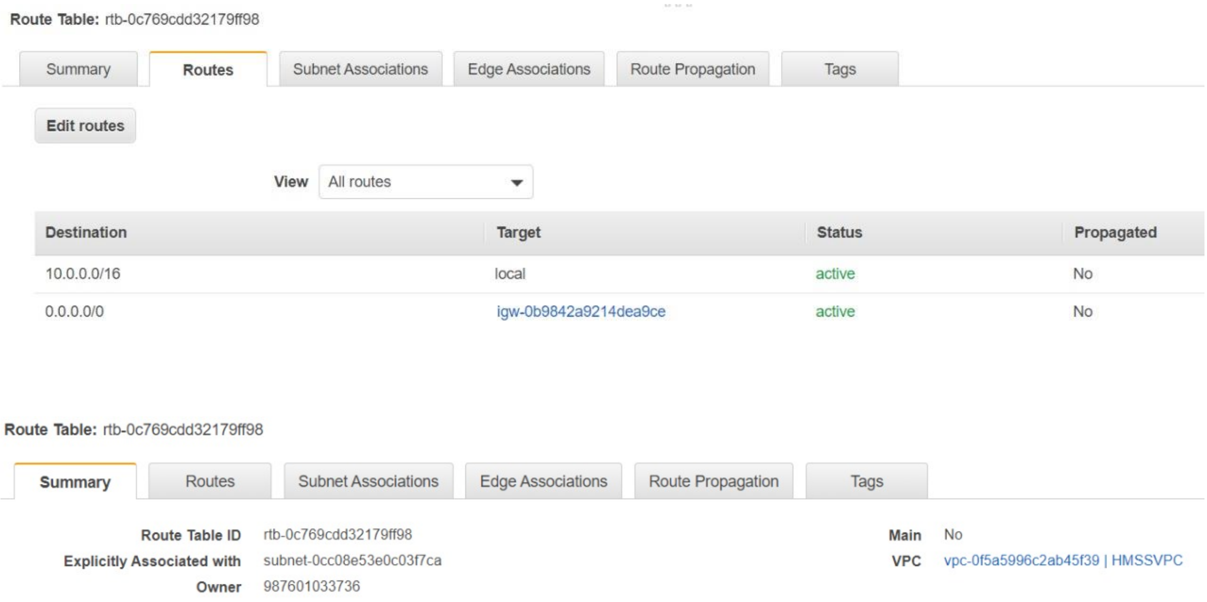
Details

Subnet ID subnet-0cc08e53e0c03f7ca	State Available	VPC vpc-0f5a5996c2ab45f39 | HMSSVPC	IPv4 CIDR 10.0.0.0/16
Available IPv4 addresses 65529	IPv6 CIDR –	Availability Zone us-east-1b	Availability Zone ID use1-az2
65529	–	us-east-1b	use1-az2
Network border group us-east-1	Route table rtb-0c769cdd32179ff98 | HMSSROUTE	Network ACL acl-077e7cc2e8f91876d	Default subnet No
Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool –
Outpost ID –	Owner 987601033736	Subnet ARN arn:aws:ec2:us-east-1:987601033736:subnet/subnet-0cc08e53e0c03f7ca	

Figure 5 - Creating a Subnet

Motivation: - We used 10.0.0.0/16 here 16 indicates no.of active hosts, we have chosen 16 instead of 24 to decrease CPU load.

- Step-2: - Creation of route tables.



Route Table: rtb-0c769cdd32179ff98

View: All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0b9842a9214dea9ce	active	No

Route Table: rtb-0c769cdd32179ff98

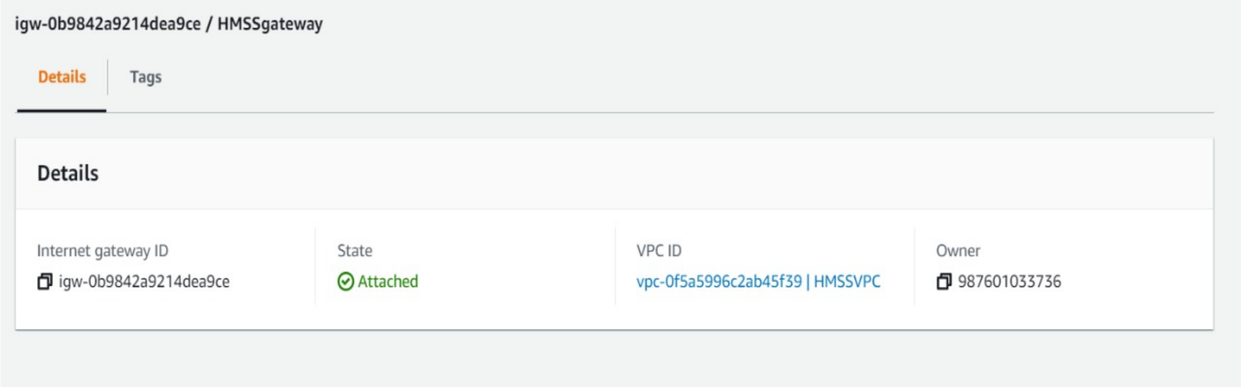
Route Table ID rtb-0c769cdd32179ff98
Explicitly Associated with subnet-0cc08e53e0c03f7ca
Owner 987601033736
Main No
VPC vpc-0f5a5996c2ab45f39 | HMSSVPC

Figure 6 - creating a Route table.

- Step-4: - Creation of internet gateway.
  Internet gateway ID
  igw-0b9842a9214dea9ce

  State
  Attached

  VPC ID
  vpc-0f5a5996c2ab45f39 | HMSSVPC



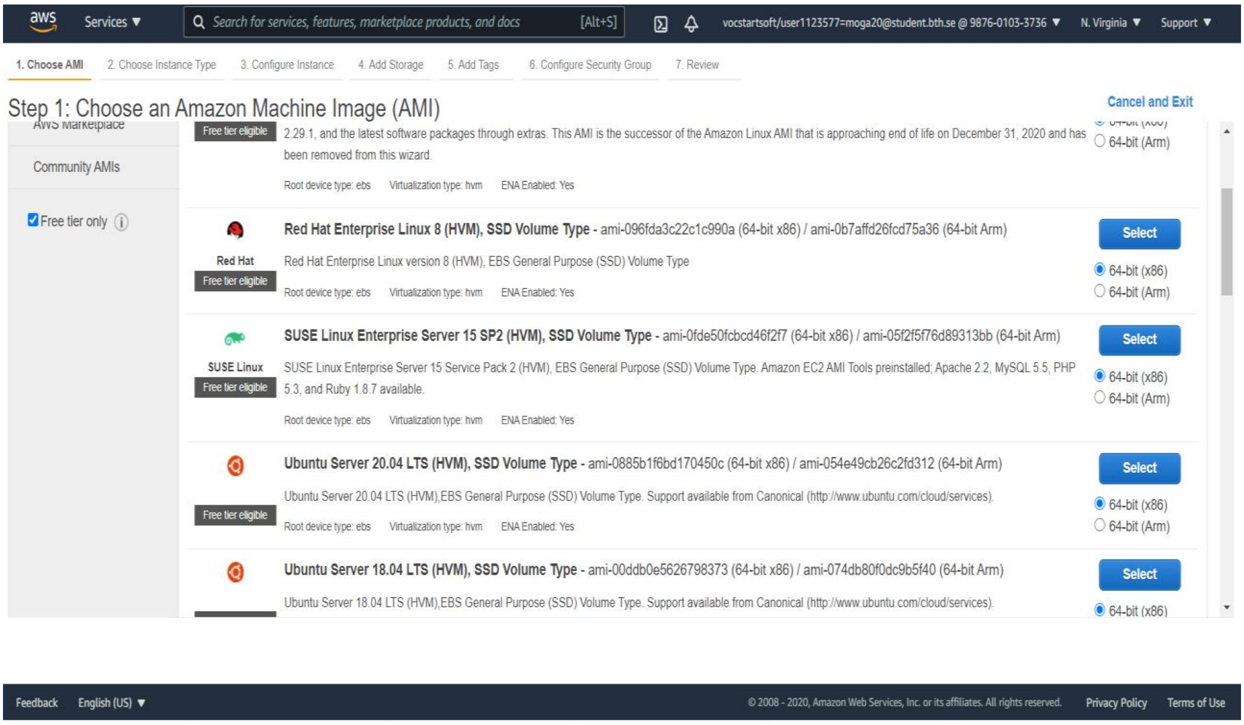
igw-0b9842a9214dea9ce / HMSSgateway

Details

Internet gateway ID	State	VPC ID	Owner
igw-0b9842a9214dea9ce	✓ Attached	vpc-0f5a5996c2ab45f39 | HMSSVPC	987601033736

Figure 7 - Creating internet gateway.

❖ <u>Launching an EC2 instance</u>: -



Step 1: Choose an Amazon Machine Image (AMI)

Free tier eligible — 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
○ 64-bit (Arm)

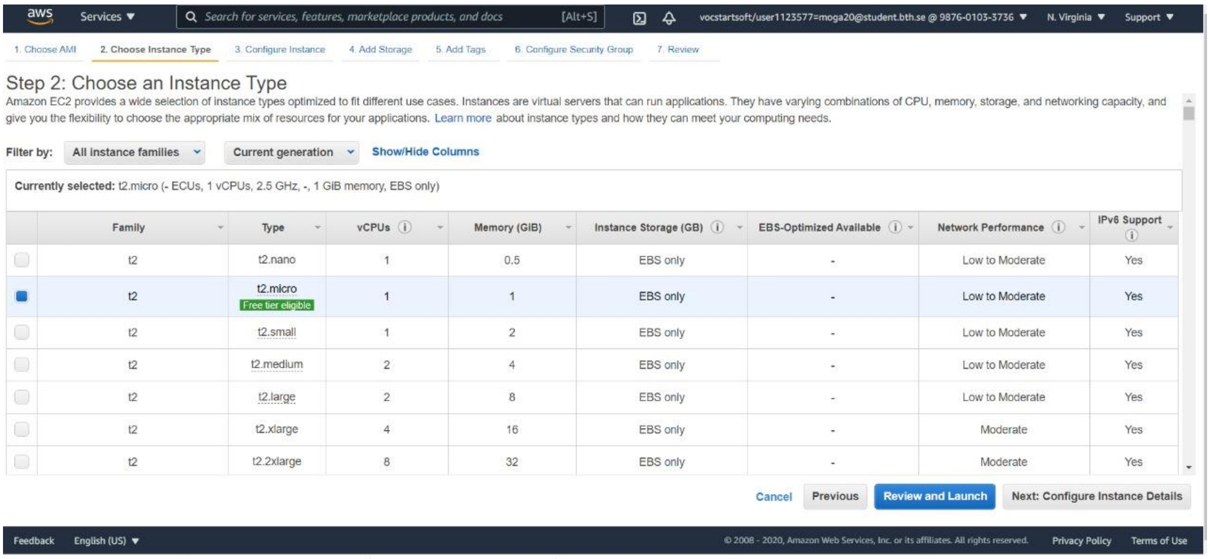
Red Hat — Free tier eligible
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-096fda3c22c1c990a (64-bit x86) / ami-0b7affd26fcd75a36 (64-bit Arm)
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select ● 64-bit (x86) ○ 64-bit (Arm)

SUSE Linux — Free tier eligible
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0fde50fcbcd46f2f7 (64-bit x86) / ami-05f2f5f76d89313bb (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select ● 64-bit (x86) ○ 64-bit (Arm)

Free tier eligible
Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0885b1f6bd170450c (64-bit x86) / ami-054e49cb26c2fd312 (64-bit Arm)
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select ● 64-bit (x86) ○ 64-bit (Arm)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-00ddb0e5626798373 (64-bit x86) / ami-074db80f0dc9b5f40 (64-bit Arm)
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Select ● 64-bit (x86)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1123577=moga20@student.bth.se @ 9876-0103-3736 ▼
N. Virginia ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 2: Choose an Instance Type

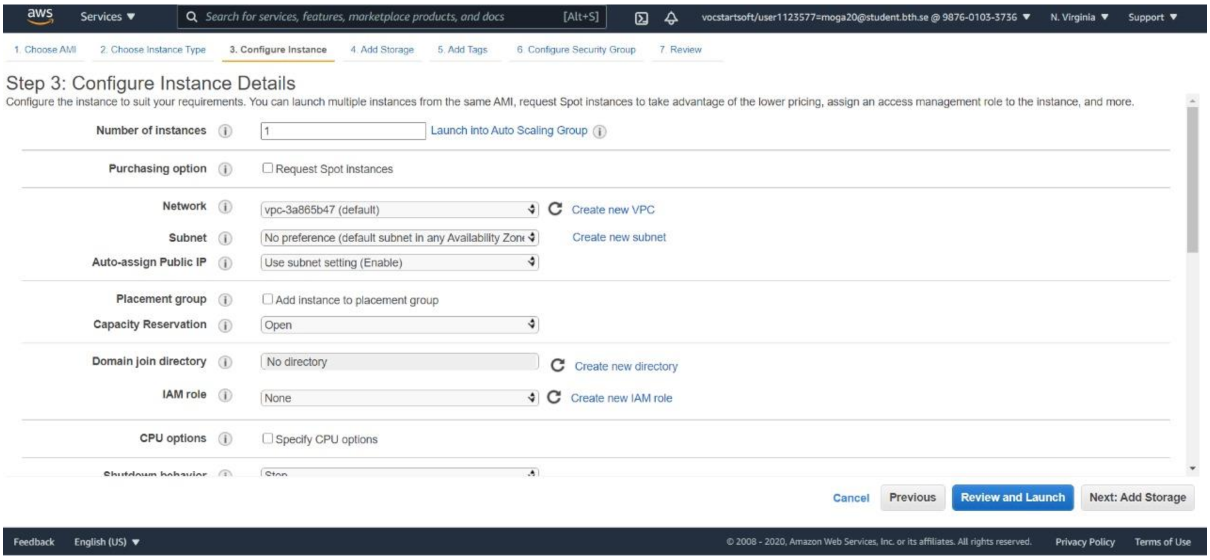
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Cancel
Previous
Review and Launch
Next: Configure Instance Details
Feedback
English (US) ▼
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1123577=moga20@student.bth.se @ 9876-0103-3736 ▼
N. Virginia ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-3a865b47 (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: □ Add instance to placement group

Capacity Reservation: Open

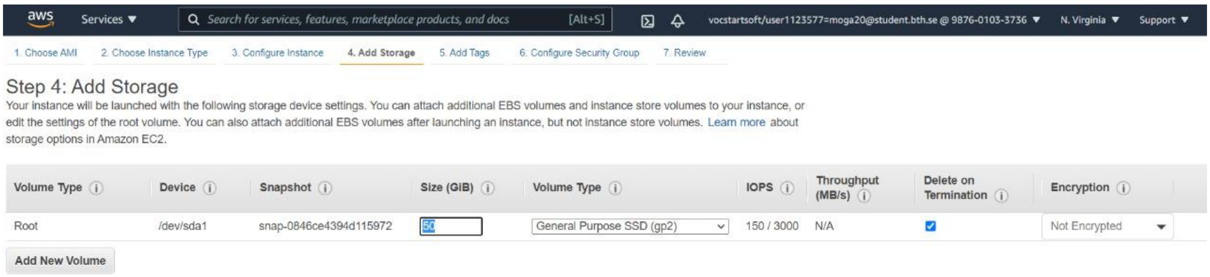
Domain join directory: No directory Create new directory

IAM role: None Create new IAM role

CPU options: □ Specify CPU options

Shutdown behavior: Stop

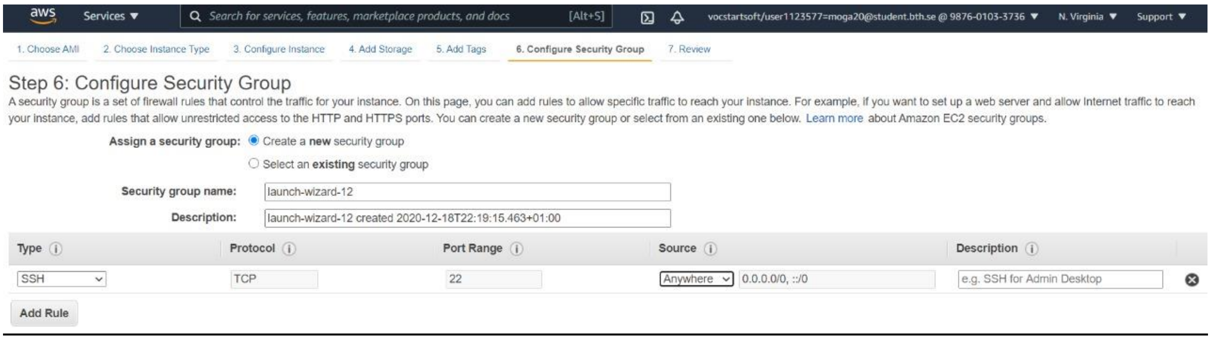
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English (US) ▼
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1123577=moga20@student.bth.se @ 9876-0103-3736 ▼
N. Virginia ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0846ce4394d115972	50	General Purpose SSD (gp2)	150 / 3000	N/A	☑	Not Encrypted


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1123577=moga20@student.bth.se @ 9876-0103-3736 ▼
N. Virginia ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 6: Configure Security Group

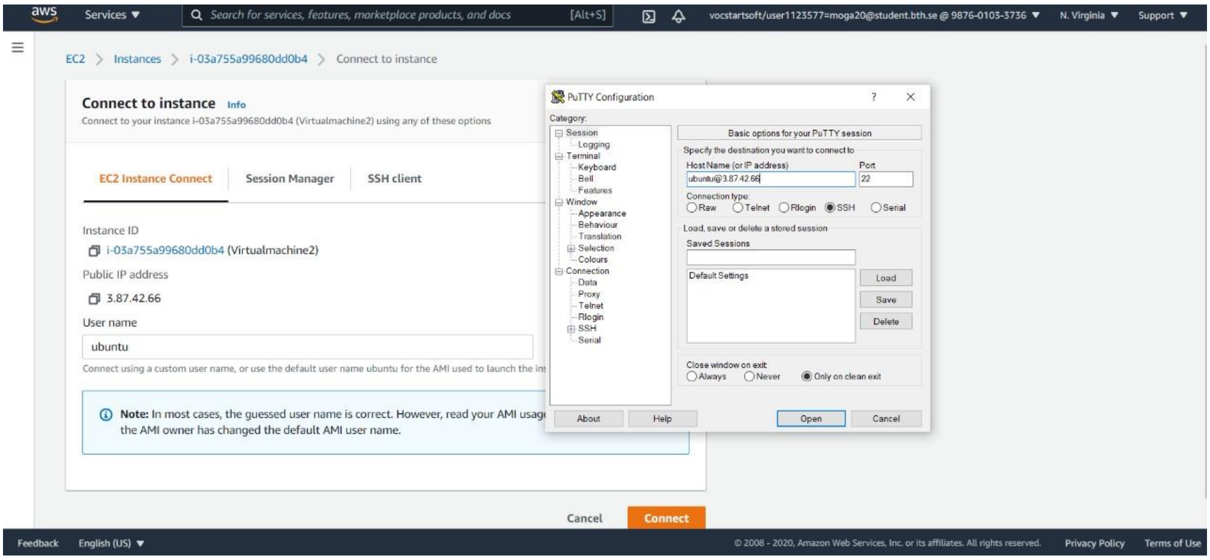
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted 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: launch-wizard-12

Description: launch-wizard-12 created 2020-12-18T22:19:15.463+01:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1123577=moga20@student.bth.se @ 9876-0103-3736 ▼
N. Virginia ▼
Support ▼
EC2 > Instances > i-03a755a99680dd0b4 > Connect to instance
EC2 Instance Connect
Session Manager
SSH client
Cancel
Connect
Feedback
English (US) ▼
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Connect to instance Info

Connect to your instance i-03a755a99680dd0b4 (Virtualmachine2) using any of these options

Instance ID

i-03a755a99680dd0b4 (Virtualmachine2)

Public IP address

3.87.42.66

User name

ubuntu

Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the ins

Note: In most cases, the guessed user name is correct. However, read your AMI usag the AMI owner has changed the default AMI user name.

PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ubuntu@3.87.42.66

Port: 22

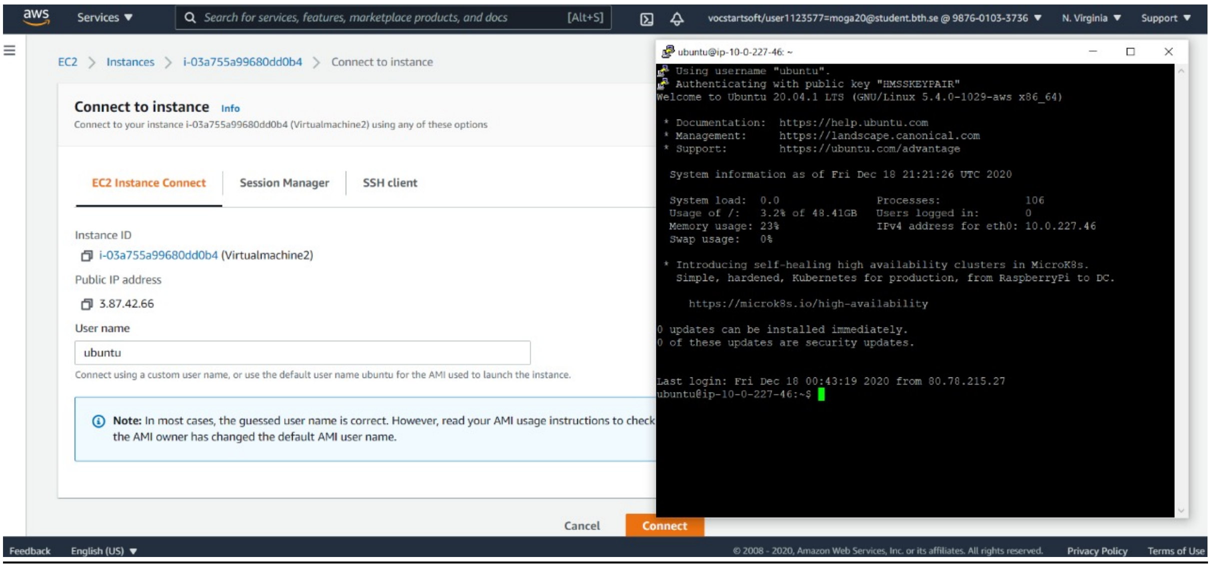
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1123577=moga20@student.bth.se @ 9876-0103-3736 ▼
N. Virginia ▼
Support ▼
EC2 > Instances > i-03a755a99680dd0b4 > Connect to instance
EC2 Instance Connect
Session Manager
SSH client
Cancel
Connect
Feedback
English (US) ▼
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Connect to instance Info

Connect to your instance i-03a755a99680dd0b4 (Virtualmachine2) using any of these options

Instance ID

i-03a755a99680dd0b4 (Virtualmachine2)

Public IP address

3.87.42.66

User name

ubuntu

Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check the AMI owner has changed the default AMI user name.

ubuntu@ip-10-0-227-46: ~

```
Using username "ubuntu".
Authenticating with public key "HMSSKEYPAIR"
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Dec 18 21:21:26 UTC 2020

  System load:  0.0               Processes:             106
  Usage of /:   3.2% of 48.41GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for eth0: 10.0.227.46
  Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

0 updates can be installed immediately.
0 of these updates are security updates.


Last login: Fri Dec 18 00:43:19 2020 from 80.78.215.27
ubuntu@ip-10-0-227-46:~$
```

TASK-2: -

- ❖ Cloud watch monitoring instance

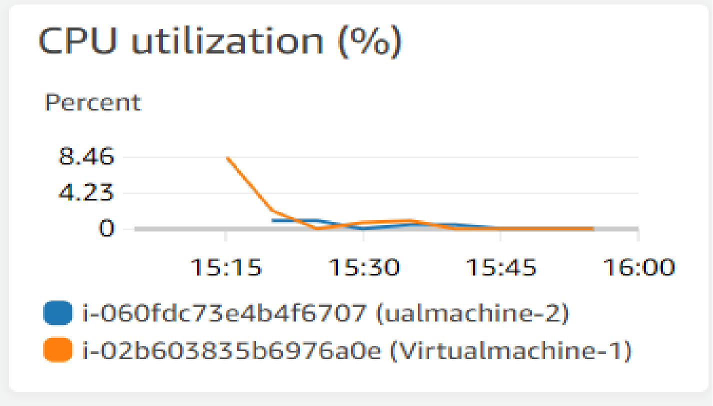


Figure 8: - CPU utilization of virtual machine-1 & 2.

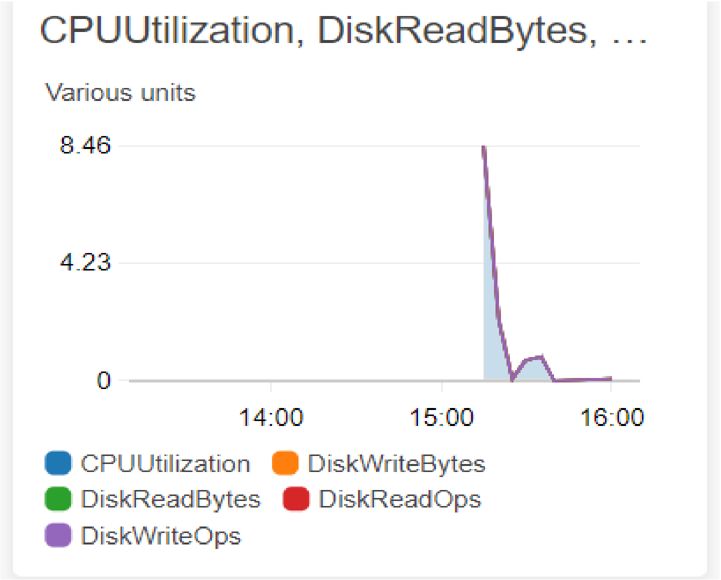


Figure 9: - Per Instance matrix of Virtual Machine-1.

As shown in figure-4 before 16:00 there is maximum utilization of CPU upto 8.46 units after there is normal utilization of CPU by virtual machine-1.

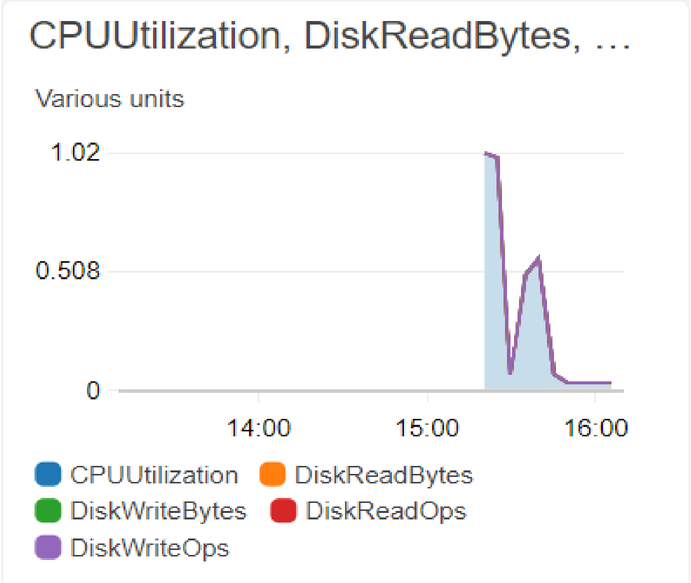


Figure 10: - Per Instance matrix of Virtual Machine-2.

As shown in figure-4 before 16:00 there is maximum utilization of CPU upto 1.02 units after there is normal utilization of CPU by virtual machine-2.

Motivation: -

We selected CPU utilization DiskReadBytes, DiskWriteBytes, DiskWriteOps, DiskReadOps from EC2 metrics to monitor the instances workload.

❖ How we generated the load: -

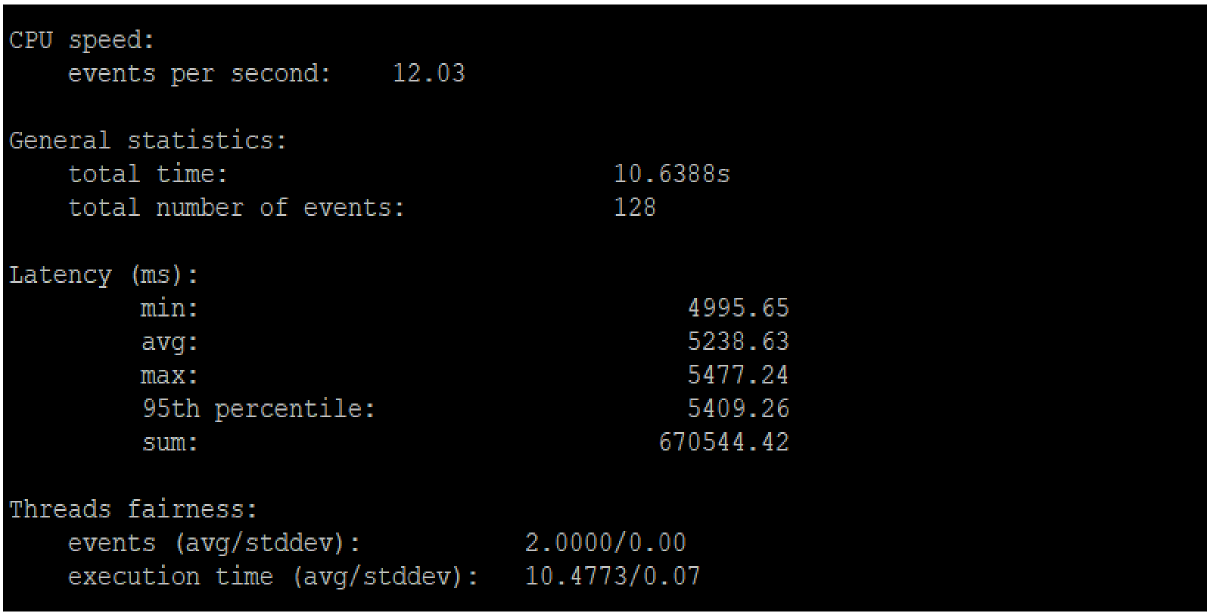
1) SYSBENCH
2) APACHEBENCH

→SYSBENCH: -

“ for each in 1 2 4 8 16 32 64; do sysbench --test=cpu--cpu-max-prime=200000 --num-threads=$each run; done “

Here in above threads increases from 1 to 64 total of 7 times SYSBENCH will run.

For executing SYSBENCH 7 times the avg execution time is 10.4773sec with CPU speed of 12.03sec.

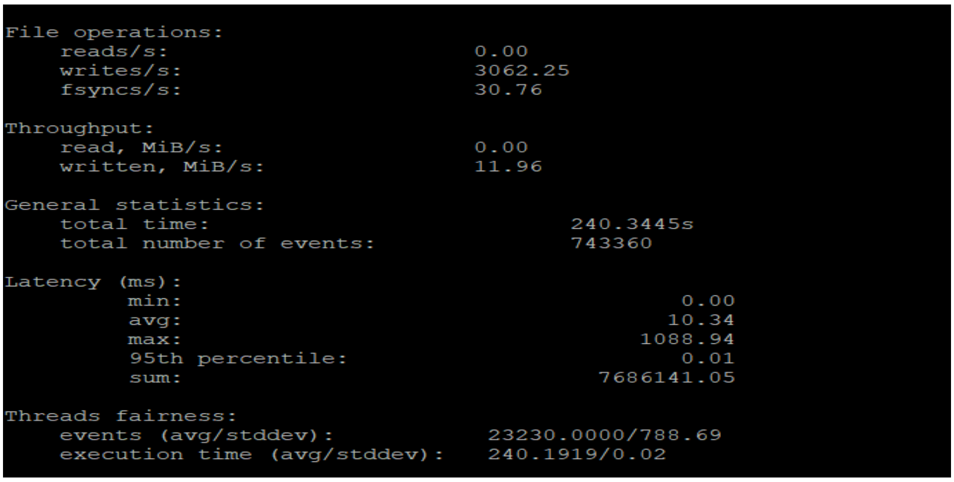


```
CPU speed:
    events per second:    12.03

General statistics:
    total time:                          10.6388s
    total number of events:              128

Latency (ms):
         min:                                 4995.65
         avg:                                 5238.63
         max:                                 5477.24
         95th percentile:                     5409.26
         sum:                               670544.42

Threads fairness:
    events (avg/stddev):           2.0000/0.00
    execution time (avg/stddev):   10.4773/0.07
```

Figure 11 - SYSBENCH output.

for each in 1 4 8 16 32; do sysbench --test=fileio--file-total-size=8G --file-test-mode=rndwr --max-time=240 --max-requests=0 --file-block-size=4K --file-num=1 --num-threads=$each run; done;

In above command we are creating a single file and performing read and write operations 5 times to generate CPU load.



```
File operations:
    reads/s:                      0.00
    writes/s:                     3062.25
    fsyncs/s:                     30.76

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               11.96

General statistics:
    total time:                          240.3445s
    total number of events:              743360

Latency (ms):
         min:                                    0.00
         avg:                                   10.34
         max:                                 1088.94
         95th percentile:                        0.01
         sum:                              7686141.05

Threads fairness:
    events (avg/stddev):           23230.0000/788.69
    execution time (avg/stddev):   240.1919/0.02
```

For performing read and write operation 5 times SYSBENCH took execution time of 240sec.

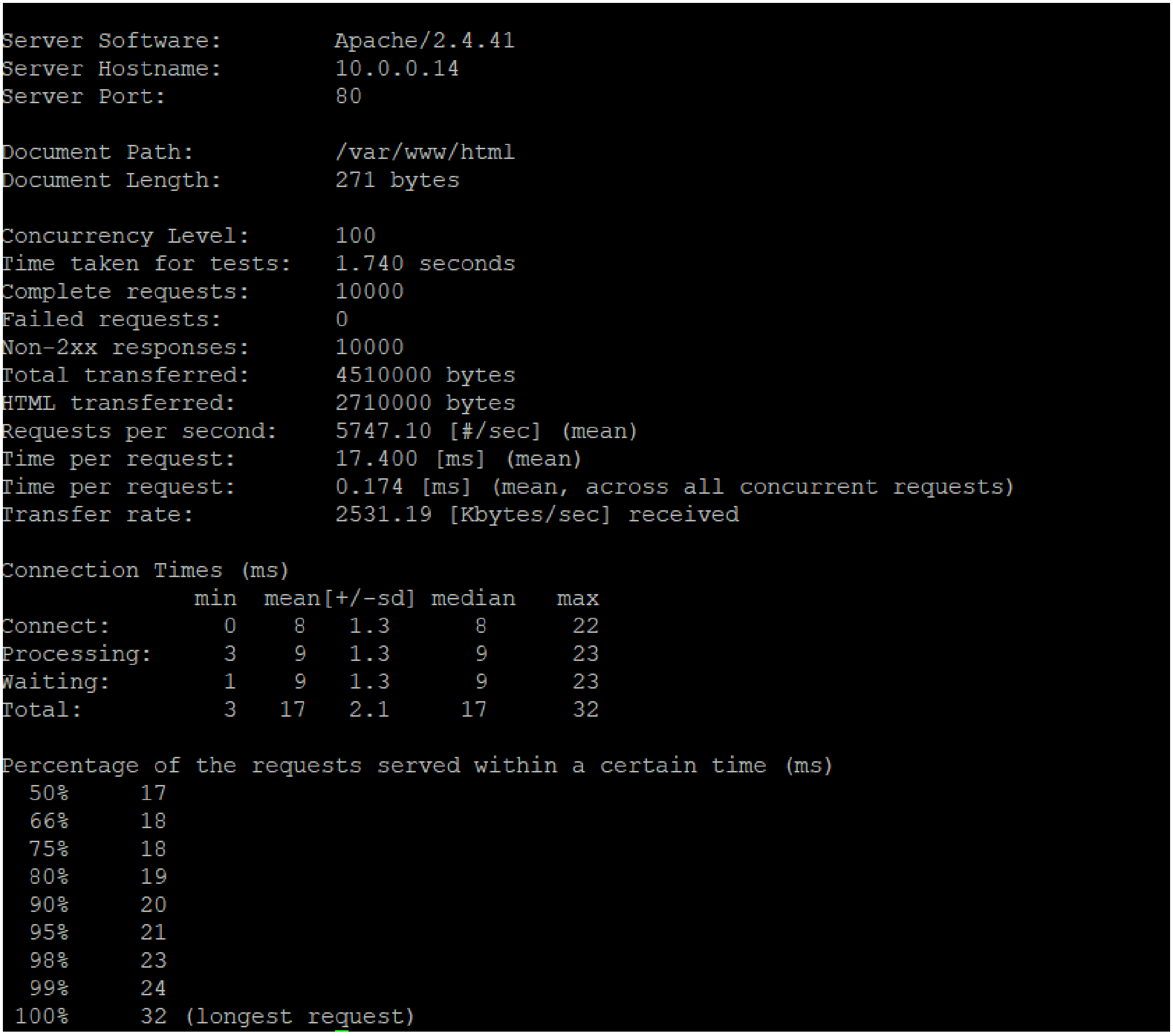
➔ APACHEBENCH: -

ab -n 10000 -c 100 **http://10.0.0.14/var/www/html**

no.of HTTP requests are 10000

no.of concurrent users are 100

and the accessing site is http://10.0.0.14/var/www/html (10.0.0.14 is the IP address).



```
Server Software:        Apache/2.4.41
Server Hostname:        10.0.0.14
Server Port:            80

Document Path:          /var/www/html
Document Length:        271 bytes

Concurrency Level:      100
Time taken for tests:   1.740 seconds
Complete requests:      10000
Failed requests:        0
Non-2xx responses:      10000
Total transferred:      4510000 bytes
HTML transferred:       2710000 bytes
Requests per second:    5747.10 [#/sec] (mean)
Time per request:       17.400 [ms] (mean)
Time per request:       0.174 [ms] (mean, across all concurrent requests)
Transfer rate:          2531.19 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    8   1.3      8      22
Processing:     3    9   1.3      9      23
Waiting:        1    9   1.3      9      23
Total:          3   17   2.1     17      32

Percentage of the requests served within a certain time (ms)
  50%     17
  66%     18
  75%     18
  80%     19
  90%     20
  95%     21
  98%     23
  99%     24
 100%     32 (longest request)
```

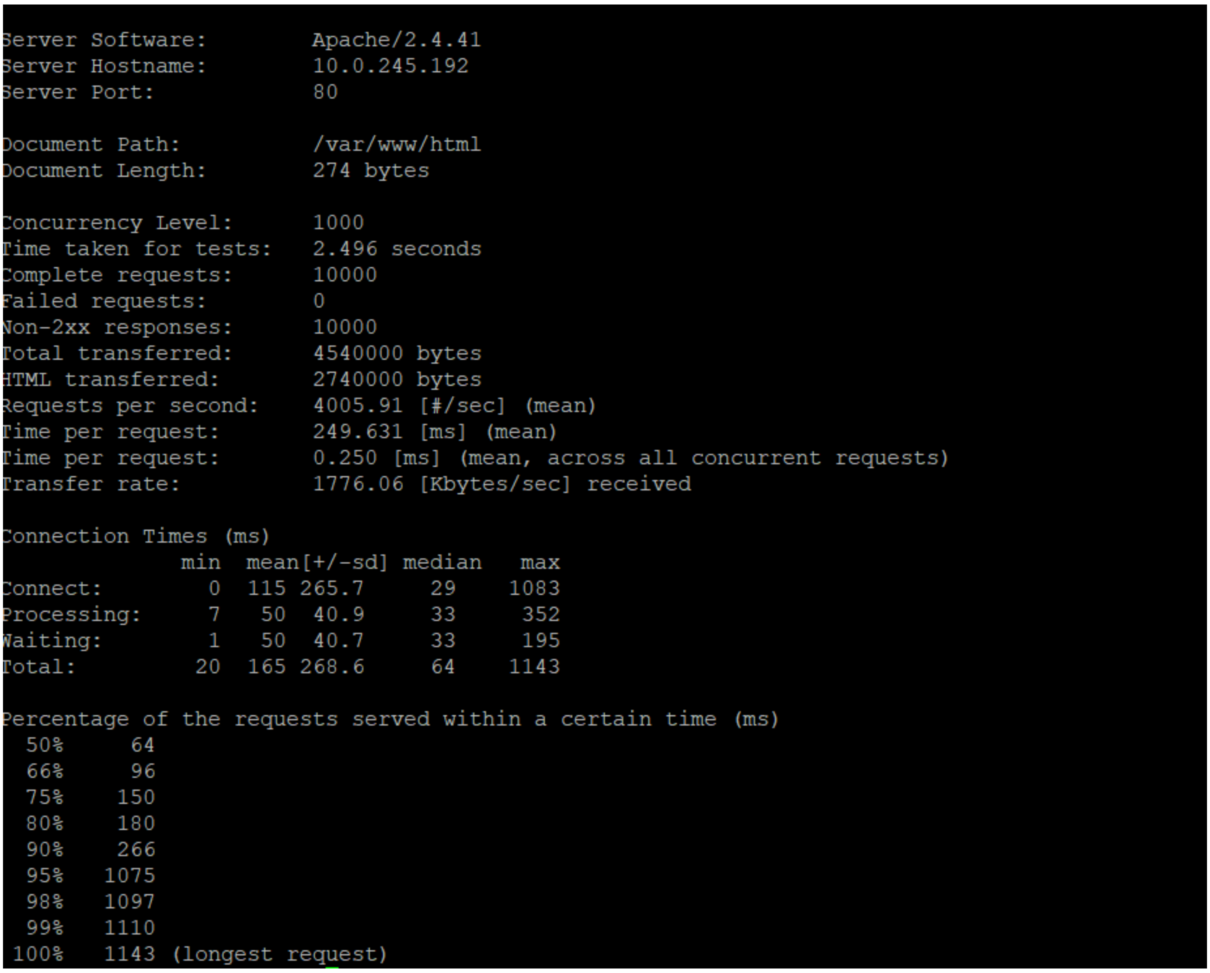
From the server port 80, we are accessing the site where time taken for each request is 17millisec and the transfer rate is 2531.19 Kb/s.

ab -n 10000 -c 1000 http://10.0.245.192/var/www/html

no.of HTTP requests are 10000

no.of concurrent users are 1000

and the accessing site is http://10.0.245.192/var/www/html (10.0.0.14 is the IP address).



```
Server Software:        Apache/2.4.41
Server Hostname:        10.0.245.192
Server Port:            80

Document Path:          /var/www/html
Document Length:        274 bytes

Concurrency Level:      1000
Time taken for tests:   2.496 seconds
Complete requests:      10000
Failed requests:        0
Non-2xx responses:      10000
Total transferred:      4540000 bytes
HTML transferred:       2740000 bytes
Requests per second:    4005.91 [#/sec] (mean)
Time per request:       249.631 [ms] (mean)
Time per request:       0.250 [ms] (mean, across all concurrent requests)
Transfer rate:          1776.06 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0  115 265.7     29    1083
Processing:     7   50  40.9     33     352
Waiting:        1   50  40.7     33     195
Total:         20  165 268.6     64    1143

Percentage of the requests served within a certain time (ms)
  50%     64
  66%     96
  75%    150
  80%    180
  90%    266
  95%   1075
  98%   1097
  99%   1110
 100%   1143 (longest request)
```

From the server port 80, we are accessing the site where time taken for each request is 249millisec and the transfer rate is 1776.06 Kb/s.

TASK-3: -

In task -1 we have created a non-default VPC using this VPC we have launched 2 VM's named as Virtual Machine-1 and Virtual Machine-2.

In Virtual Machine-1, we have installed "apache2"

sudo apt-get install apache2

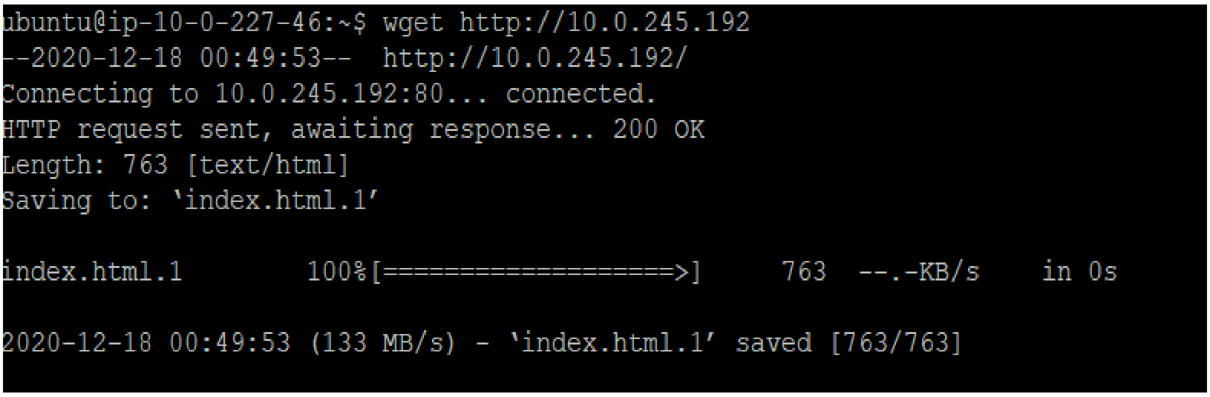
In Virtual Machine-2, we have installed “apache2-utils”

sudo apt-get install -y apache2-utils

In Virtual Machine-1 we will upload an HTML page using a command “sudo mv index.html index.html.bak”.

To establish the connections between two virtual machines we have to establish communication between 2 VM’s the below command helps us to do this work.

wget http://10.0.245.192



```
ubuntu@ip-10-0-227-46:~$ wget http://10.0.245.192
--2020-12-18 00:49:53--  http://10.0.245.192/
Connecting to 10.0.245.192:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 763 [text/html]
Saving to: ‘index.html.1’

index.html.1        100%[===================>]     763  --.-KB/s    in 0s

2020-12-18 00:49:53 (133 MB/s) - ‘index.html.1’ saved [763/763]
```

Now we have to find the workload generated by the HTML page in Virtual Machine-1 in Virtual Machine-2.

ab -n 10000 -c 100 http://10.0.245.192/var/www/html

N (no.of HTTP requests)	C (concurrent users)	Time per request (milli sec)
10000	100	11.027
10000	1000	192
100000	1000	134
100000	100	11
100000	10	1

In the above table we have observed the n & c values for 2 VM’s and we have observed that depending on concurrent users the time per request is changing.

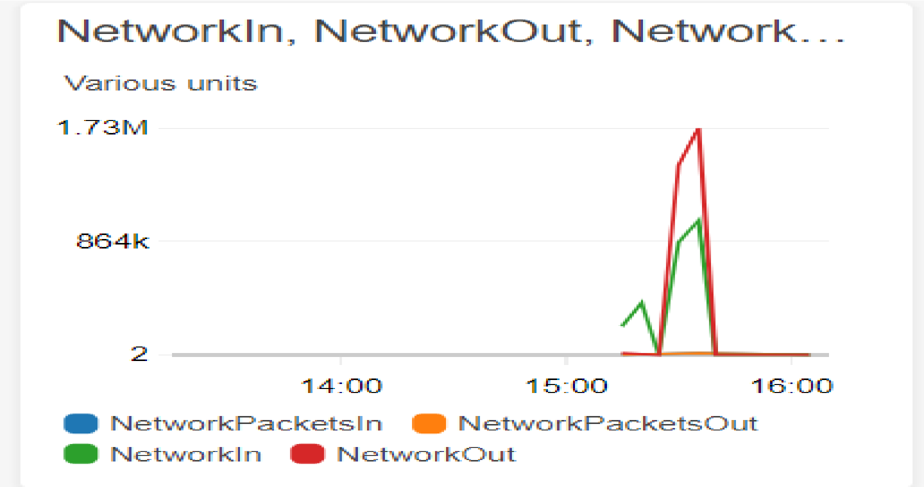


Figure 12: - Networking of Virtual Machine-1.

In the above graph it indicates the network traffic in the virtual machine-1, in the Network-in graph we observed transfer bytes up to 864k, in Network-out graph we observed received bytes up to 1.73M. Network packets-in and network packets-out are about 2k.

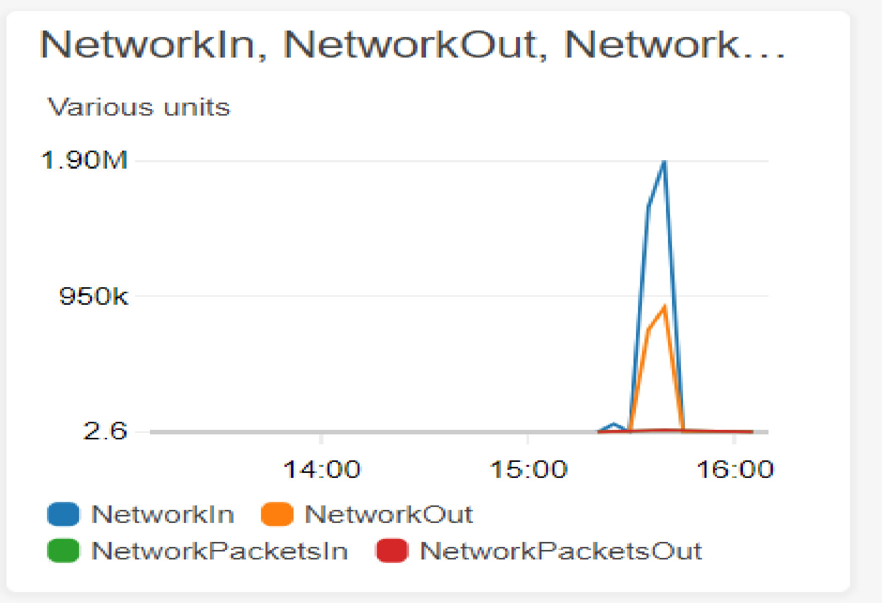


Figure 13: - Networking of Virtual Machine-2.

In the above graph it indicates the network traffic in the virtual machine-2, in the Network-in graph we observed transfer bytes up to 1.90M, in Network-out graph we observed received bytes up to 950k. Network packets-in and network packets-out are about 2k.

Motivation-

As we have chosen 16 subnets in non-default VPC, so we have selected no.of HTTP requests upto 1lakh and no.of concurrent users upto 1000 and observed the variations. Here no.of hosts are based on subnets.