

# DV1566

## INTRODUCTION TO CLOUD COMPUTING

### GROUP ASSIGNMENT -2

#### **Laboratory 2: Monitoring VM performance counters**

**Group Members:**

Boya Sada Siva Kumar

Gundala Harshitha

Konka Cherishma

# Task 1 - Setup and run an amazon EC2 instance

## Creating VPC's

We created a VPC named “**Common+VPC**” and allocated an IP range of 10.50.0.0/16, where the last two digits of the IP range will vary and are dedicated to the hosts.

## VPC Details

**Our VPC:** vpc-09046937d79bac5c9

**Route Table:** rtb-06dbcac129b3842b9

**IPv4 CIDR:** 10.50.0.0/16

**Tags:** Common+VPC

vpc-09046937d79bac5c9 / Common+VPC			
Details		Actions ▾	
VPC ID vpc-09046937d79bac5c9	State <span style="color: green;">Available</span>	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-00c4744d83040dc7a	Main route table rtb-06dbcac129b3842b9	Main network ACL acl-0b3eaa0c9861082d5
Default VPC No	IPv4 CIDR 10.50.0.0/16	IPv6 pool -	IPv6 CIDR -
Route 53 Resolver DNS Firewall rule groups -	Owner ID 811572106700		

## Internet Gateway

We Created and named the Internet Gateway as “**Common\_IGW**”.

## Internet Gateway Details

**Internet gateway ID:** igw-0609f41b29bb87d1d

**VPC ID:** vpc-09046937d79bac5c9 | Common+VPC

**Tags:** Common\_IGW

igw-0609f41b29bb87d1d / Common\_IGW

Actions ▾

Details		Info	
Internet gateway ID igw-0609f41b29bb87d1d	State Attached	VPC ID vpc-09046937d79bac5c9   Common+VPC	Owner 811572106700

Tags

Manage tags

Key	Value
Name	Common_IGW

### Creating Subnets:

We created “**VM1\_Subnet**” with IPv4 range as 10.50.0.0/24 and “**VM2\_Subnet**” with IPv4 range as 10.50.1.0/24, which has the first 3 digits of IP is constant and the last digit varies with different hosts. The third digit of the IP’s are different which indicates the sub-ranges of the common VPC IP range.

### Subnet Details:

#### **VM1\_Subnet:** -

**Subnet ID:** subnet-0d9a3e52212104403

**VPC:** vpc-09046937d79bac5c9 | Common+VPC

**Available IPv4 addresses:** 251

**Route table:** rtb-0dca284bc902f62a7 | VM1\_RouteTable

**IPv4 CIDR:** 10.50.0.0/24

subnet-0d9a3e52212104403 / VM1_Subnet				Actions ▾
Details				
Subnet ID subnet-0d9a3e52212104403	Subnet ARN arn:aws:ec2:us-east-2:811572106700:subnet/subnet-0d9a3e52212104403	State Available	IPv4 CIDR 10.50.0.0/24	
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone us-east-2a	Availability Zone ID use2-az1	
VPC vpc-09046937d79bac5c9   Common+VPC	Route table rtb-0dca284bc902f62a7   VM1_RouteTable	Network ACL acl-0b3eaa0c9861082d5	Default subnet No	
Auto-assign public IPv4 address No		Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -	
Outpost ID -	Auto-assign IPv6 address No	IPv6 CIDR reservations -	IPv6-only No	
Hostname type IP name	IPv4 CIDR reservations -	Resource name DNS AAAA record Disabled	Owner 811572106700	
	Resource name DNS A record Disabled			

## VM2\_Subnet: -

**Subnet ID:** subnet-0b2d7482a0427ec12

**VPC:** vpc-09046937d79bac5c9 | Common+VPC

**Available IPv4 addresses:** 251

**Route table:** rtb-079fad805be83112a | VM2\_RouteTable

**IPv4 CIDR:** 10.50.1.0/24

subnet-0b2d7482a0427ec12 / VM2_Subnet				Actions ▾
Details				
Subnet ID subnet-0b2d7482a0427ec12	Subnet ARN arn:aws:ec2:us-east-2:811572106700:subnet/subnet-0b2d7482a0427ec12	State Available	IPv4 CIDR 10.50.1.0/24	
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone us-east-2a	Availability Zone ID use2-az1	
VPC vpc-09046937d79bac5c9   Common+VPC	Route table rtb-079fad805be83112a   VM2_RouteTable	Network ACL acl-0b3eaa0c9861082d5	Default subnet No	
Auto-assign public IPv4 address No		Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -	
Outpost ID -	Auto-assign IPv6 address No	IPv6 CIDR reservations -	IPv6-only No	
Hostname type IP name	IPv4 CIDR reservations -	Resource name DNS AAAA record Disabled	Owner 811572106700	
	Resource name DNS A record Disabled			

## Creating a Security Group:

We created “**VM1\_SG**” with three Inbound rules and single outbound rule, and “**VM2\_SG**” with two Inbound rules and single outbound rule. We had given source as “**My IP**” in Inbound rules and “**IPv4 Anywhere**” in outbound rules. We used two types to route the connections they are “**SSH**” and “**HTTP**”.

### Security Group Details:

#### **VM1\_SG:** -

**Security Group Name:** VM1\_SG

**Security group ID:** sg-0f3930a67c746f9d2

**VPC ID:** vpc-09046937d79bac5c9

#### **Inbound rules:**

HTTP - 80.78.216.68/32 (sgr-0f0937ec024459300)

HTTP - 10.50.1.4/32 (sgr-062ac8afdf7c01eee)

SSH - 80.78.216.68/32 (sgr-0aee94a3ff6c05175)

#### **Outbound rules:**

All traffic – 0.0.0.0/0

The screenshot shows the AWS EC2 Security Groups details page for the security group "sg-0f3930a67c746f9d2 - VM1\_SG". The top navigation bar includes "EC2 > Security Groups > sg-0f3930a67c746f9d2 - VM1\_SG" and an "Actions" dropdown. The main content area has a "Details" tab selected, displaying the following information:

Security group name	Security group ID	Description	VPC ID
VM1_SG	sg-0f3930a67c746f9d2	Allowed	vpc-09046937d79bac5c9

Below this, there are additional sections for Owner (811572106700), Inbound rules count (3 Permission entries), and Outbound rules count (1 Permission entry).

**Edit inbound rules** Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <small>Info</small>						Description - optional <small>Info</small>
Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>		
sgr-062ac8afdf7c01eee	HTTP	TCP	80	Custom ▾	Q <input type="text"/> 10.50.1.4/32 <span style="color: red;">X</span>	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>
sgr-0f0937ec024459300	HTTP	TCP	80	Custom ▾	Q <input type="text"/> 80.78.216.68/52 <span style="color: red;">X</span>	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>
sgr-0aee94a3ff6c05175	SSH	TCP	22	Custom ▾	Q <input type="text"/> 80.78.216.68/32 <span style="color: red;">X</span>	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>

Add rule Cancel Preview changes Save rules

**Edit outbound rules** Info

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules <small>Info</small>						Description - optional <small>Info</small>
Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>		
sgr-0198aec82fbfd9cf	All traffic	All	All	Custom ▾	Q <input type="text"/> 0.0.0.0/0 <span style="color: red;">X</span>	<span style="border: 1px solid #ccc; padding: 2px;">Delete</span>

Add rule Cancel Preview changes Save rules

**VM2\_SG:** -

**Security Group Name:** VM2\_SG

**Security group ID:** sg-0f93467e0f30a54c4

**VPC ID:** vpc-09046937d79bac5c9

#### **Inbound rules:**

HTTP - 80.78.216.68/32 (sgr-0c2f55c043e459156)

SSH- 80.78.216.68/32 (sgr-0916c01f1cc131024)

#### **Outbound rules:**

All traffic – 0.0.0.0/0

EC2 > Security Groups > sg-0f93467e0f30a54c4 - VM2\_SG

### sg-0f93467e0f30a54c4 - VM2\_SG

[Actions ▾](#)

Details			
Security group name VM2_SG	Security group ID sg-0f93467e0f30a54c4	Description Allowed	VPC ID vpc-09046937d79bac5c9
Owner 811572106700	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

### Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <a href="#">Info</a>						
Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-0916c01f1cc131024	SSH	TCP	22	Custom	<input type="text" value="80.78.216.68/32"/> X	<a href="#">Delete</a>
sgr-0c2f55c043e459156	HTTP	TCP	80	Custom	<input type="text" value="80.78.216.68/32"/> X	<a href="#">Delete</a>

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

### Edit outbound rules [Info](#)

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules <a href="#">Info</a>						
Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-00a38c2de11eaf909	All traffic	All	All	Custom	<input type="text" value="0.0.0.0/0"/> X	<a href="#">Delete</a>

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

## Motivation

We choose “My IP” as our source because if we had taken IPv4 anywhere then our system or instance would be vulnerable. So we decided to use “My IP” only.

## Creating a Route Table:

We created a route tables named “**VM1\_RouteTable**”and “**VM2\_RouteTable**”. And here we need to edit the routes by “edit routes” option and also connect the routes to the subnet through subnet associations. In edit routes we connected the routes or different IPs which are destinations and the target will be connected to internet gateways or the network connections.

### Route Table Details:

#### **VM1\_Route Table: -**

**Route Table Name:** VM1\_RouteTable

**Route Table ID:** rtb-0dca284bc902f62a7

**VPC ID:** vpc-09046937d79bac5c9 | Common+VPC

The screenshot shows the AWS VPC Route Tables page. The top navigation bar includes 'VPC' and 'Route tables'. Below the navigation, the specific route table is identified as 'rtb-0dca284bc902f62a7 / VM1\_RouteTable'. A callout box provides a tip: 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button and a close 'X' button. The main content area displays the 'Details' tab of the route table configuration. The table contains the following information:

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0dca284bc902f62a7	No	subnet-0d9a3e52212104403 / VM1_Subnet	-
VPC	Owner ID		
vpc-09046937d79bac5c9   Common+VPC	811572106700		

VPC > Route tables > rtb-0dca284bc902f62a7 > Edit routes

### Edit routes

Destination	Target	Status	Propagated
10.50.0.0/16	<input type="text" value="local"/> <span>X</span>	<span>Active</span>	No
<input type="text" value="80.78.216.68/32"/> <span>X</span>	<input type="text" value="igw-0609f41b29bb87d1d"/> <span>X</span>	<span>Active</span>	No <span>Remove</span>
<input type="text" value="0.0.0.0/0"/> <span>X</span>	<input type="text" value="igw-0609f41b29bb87d1d"/> <span>X</span>	<span>Active</span>	No <span>Remove</span>
<span>Add route</span>			

Cancel Preview Save changes

### VM2\_Route Table: -

**Route Table Name:** VM2\_RouteTable

**Route Table ID:** rtb-079fad805be83112a

**VPC ID:** vpc-09046937d79bac5c9 | Common+VPC

VPC > Route tables > rtb-079fad805be83112a

### rtb-079fad805be83112a / VM2\_RouteTable

Actions ▾

ⓘ You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer X

Details		Info	
Route table ID	<input type="text" value="rtb-079fad805be83112a"/> <span>&gt;Main</span>	Explicit subnet associations	Edge associations
VPC	<input type="text" value="vpc-09046937d79bac5c9   Common+VPC"/> <span>No</span>	<a href="#">subnet-0b2d7482a0427ec12 / VM2_Subnet</a>	-
	Owner ID	<input type="text" value="811572106700"/>	

## Edit routes

Destination	Target	Status	Propagated
10.50.0.0/16	Q local X	Active	No
Q 80.78.216.68/32 X	Q igw-0609f41b29bb87d1d X	Active	No
Q 0.0.0.0/0 X	Q igw-0609f41b29bb87d1d X	Active	No
<a href="#">Add route</a>			
			<a href="#">Cancel</a> <a href="#">Preview</a> <a href="#">Save changes</a>

## Creating an EC2 Instance:

Step 1: Initially we need to select which AMI template we need to create so, we selected Ubuntu server (64 bit) from different templates provided by AWS.

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review

Step 1: Choose an Amazon Machine Image (AMI)

<b>SUSE Linux Enterprise Server 15 SP3 (HVM), SSD Volume Type</b> - ami-0aa107e3dfaef94861 (64-bit x86) / ami-0726f047f10548ec5 (64-bit Arm)	<a href="#">Select</a>
<b>Ubuntu Server 20.04 LTS (HVM), SSD Volume Type</b> - ami-0fb653ca2d3203ac1 (64-bit x86) / ami-02af65b2d1ebdfafcc (64-bit Arm)	<a href="#">Select</a>
<b>Microsoft Windows Server 2019 Base</b> - ami-0c95aae83dc5e60ac	<a href="#">Select</a>
<b>Microsoft Windows Server 2019 Base with Containers</b> - ami-0ba02e7ab1c9dd052	<a href="#">Select</a>

Step 2: we choose t2 micro instance type.

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	<b>t2 micro</b> <small>Free tier eligible</small>	1	<b>1</b>	EBS only	-	Low to Moderate	Yes
	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes

Step 3: Configuring the instance details - network is selected COMMON+VPC where subnet will be directly connected and public IP is enabled.

### 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	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-09046937d79bac5c9   Common+VPC	<input type="button" value="Create new VPC"/>
Subnet	subnet-0d9a3e52212104403   VM1_Subnet   us-east-1	<input type="button" value="Create new subnet"/> 250 IP Addresses available
Auto-assign Public IP	Enable	
Hostname type	<input type="button" value="Use subnet setting (IP name)"/>	
DNS Hostname	<input type="checkbox"/> Enable IP name IPv4 (A record) DNS requests <input checked="" type="checkbox"/> Enable resource-based IPv4 (A record) DNS requests <input type="checkbox"/> Enable resource-based IPv6 (AAAA record) DNS requests	

### Step 4: Adding storage here we took 8 GB of storage or the instance.

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-092498e2cc2e3eb82	<input type="text" value="8"/>	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
<a href="#">Add New Volume</a>								

### Step 5: Here we added the tag name as COMMON+VPC in the value column.

#### Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name		Common+VPC		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Add another tag</a> (Up to 50 tags maximum)						<input type="button" value="X"/>

### Step 6: Here security group is created or an existing security group is selected.

#### 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 ID	Name	Description	Actions
sg-0811bab376b1dac9	default	default VPC security group	<a href="#">Copy to new</a>
sg-0f3930a67c7469d2	VM1_SG	Allowed	<a href="#">Copy to new</a>
sg-0f93467e0f30a54c4	VM2_SG	Allowed	<a href="#">Copy to new</a>

Step 7: Review instance launch. And existing key pair can be choose here or we can create new key pair here.

**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**

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0fb653ca2d3203ac1  
Free tier eligible  
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

**Instance Type**

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

**Security Groups**

Security Group ID	Name	Description
sg-0f3930a67c746f9d2	VM1_SG	Allowed

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	80.78.216.68/32	
HTTP	TCP	80	10.50.1.4/32	
SSH	TCP	22	80.78.216.68/32	

**AMI Details**

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0fb653ca2d3203ac1  
Free tier eligible  
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

**Instance Type**

Instance Type	ECUs	vCPUs	Memory (GiB)
t2.micro	-	1	1

**Security Groups**

Security Group ID	Name
sg-0f3930a67c746f9d2	VM1_SG

All selected security groups inbound rules

Type	Protocol
------	----------

**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. Amazon EC2 supports ED25519 and RSA key pair types.

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  
Common\_KeyPair | RSA

I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

**Cancel** **Launch Instances**

Now the instance will get created and launched.

## Task 2 – Cloud watch monitoring instance

### Generating the workload:

#### **Sysbench:**

Here for the workload generation we used Sysbench tool.

We installed Sysbench tool by accessing EC2 and running following commands.

“**sudo apt-get update**”

“**sudo apt-get install sysbench** “.

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

System information disabled due to load higher than 1.0

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-50-0-46:~$ sudo su
root@ip-10-50-0-46:/home/ubuntu# apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
0% [5 Packages store 0 B] [6 Translation-en 25.9 kB/5124 kB 1%]
```

```

Reading package lists... Done
root@ip-10-50-0-46:/home/ubuntu# apt-get install sysbench
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common
The following NEW packages will be installed:
  libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common sysbench
0 upgraded, 6 newly installed, 0 to remove and 16 not upgraded.
Need to get 1794 kB of archives.
After this operation, 9189 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 libluajit-5.1-common all 2.1.0~beta3+dfsg-5.1build1 [44.3 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 libluajit-5.1-2 amd64 2.1.0~beta3+dfsg-5.1build1 [228 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 mysql-common all 5.8+1.0.5ubuntu2 [7496 B]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libmysqlclient21 amd64 8.0.27-0ubuntu0.20.04.1 [1291 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpq5 amd64 12.9-0ubuntu0.20.04.1 [117 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 sysbench amd64 1.0.18+ds-1 [107 kB]
Fetched 1794 kB in 0s (20.7 MB/s)

```

The load will be generated by using these commands “ **for each in 1 2 4 8 16 32 64; do sysbench --test=cpu --cpu-maxprime=200000 --num-threads=\$each run; done** ”.

```

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 64
Initializing random number generator from current time

Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
    events per second: 13.85

General statistics:
    total time:          13.8635s
    total number of events: 192

Latency (ms):
    min:                4315.84
    avg:                4567.74
    max:                4884.37
    95th percentile:    4855.31
    sum:                877006.84

Threads fairness:
    events (avg/stddev): 3.0000/0.00
    execution time (avg/stddev): 13.7032/0.08

```

The Statistics for the execution is:

Average Execution Time – 13.7032 s

CPU speed – 13.85

We created a 1 GB file and performed read and write operations 5 times to generate CPU load with the following command. “**for each in 1 4 8 16 32; do sysbench --test=fileio--file-total-size=1G --file-test-mode=rndwr --max-time=240 --max-requests=0 --file-block-size=4K --file-num=3 --num-threads=\$each run; done;**”

```
187371382 bytes written in 0.62 seconds (116111 KB/sec).
root@ip-10-50-0-46:/home/ubuntu# for each in 1 4 8 16 32; do sysbench --test=fileio --file-total-size=1G --file-test-mode=rndwr --max-time=240 --max-requests=0 --file-block-size=4K --file-num=3 --num-threads=$each run; done;
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

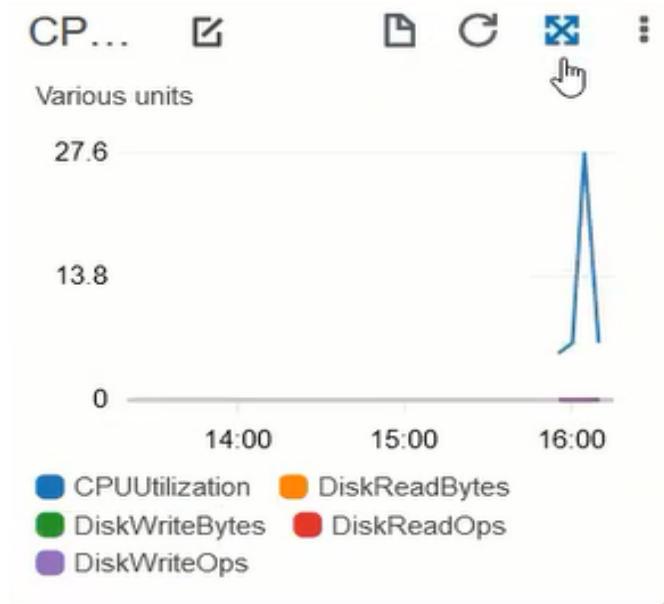
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
3 files, 341.33MiB each
1024MiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!
```

For monitoring the load we created a dashboard “**Monitoring**”.

And for the metrics we used line widget. CPU utilization, Disk Read Bytes, Disk Read Ops, Disk Write Bytes, Disk Write Ops.



## Task 3 – Communication

We have created a non-default VPC and launched two VM's – VM1 and VM2 in the VPC.

In VM1, we have installed Apache Web Server (apache2) using the below command:

“**sudo apt-get install apache2**” .

```
root@ip-10-50-0-46:/home/ubuntu# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4
  liblua5.2-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libjansson4 liblua5.2-0 ssl-cert
0 upgraded, 11 newly installed, 0 to remove and 16 not upgraded.
Need to get 1866 kB of archives.
After this operation, 8091 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36-6ubuntu1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
```

And in VM2, we have installed Apache web utils(apache2) using the following command: “**sudo apt-get install -y apache2-utils**”

```
root@ip-10-50-1-4:/home/ubuntu# apt-get install apache2.utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'apache2-utils' for regex 'apache2.utils'
The following additional packages will be installed:
  libapr1 libaprutil1
The following NEW packages will be installed:
  apache2-utils libapr1 libaprutil1
0 upgraded, 3 newly installed, 0 to remove and 16 not upgraded.
Need to get 261 kB of archives.
After this operation, 970 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1.6.5-1ubuntu1 [91.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libaprutil1 amd64 1.6.1-4ubuntu2 [84.7 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-utils amd64 2.4.41-4ubuntu3.8 [84.5 kB]
Fetched 261 kB in 0s (8594 kB/s)
```

Now the connection between two VM's will be established.

Here we used the command to generate Apache bench stress: "ab -n 100000 -c 100 http://10.50.0.46 / var/www/html/index.html".

```
root@ip-10-50-0-46:/var/www/html#
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36-6ubuntu1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
root@ip-10-50-0-46:/home/ubuntu# /var/www/html/
bash: /var/www/html/: Is a directory
root@ip-10-50-0-46:/home/ubuntu# cd /var/www/html/
root@ip-10-50-0-46:/var/www/html# vi index
root@ip-10-50-1-4:/home/ubuntu#
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-utils amd64 2.4.41-4ubuntu3.8 [84.5 kB]
Fetched 261 kB in 0s (8594 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 63895 files and directories currently installed.)
Preparing to unpack .../libapr1_1.6.5-1ubuntu1_amd64.deb ...
Unpacking libapr1:amd64 (1.6.5-1ubuntu1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../libaprutil1_1.6.1-4ubuntu2_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-4ubuntu2) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../apache2-utils_2.4.41-4ubuntu3.8_amd64.deb ...
Unpacking apache2-utils (2.4.41-4ubuntu3.8) ...
Setting up libapr1:amd64 (1.6.5-1ubuntu1) ...
Setting up libaprutil1:amd64 (1.6.1-4ubuntu2) ...
Setting up apache2-utils (2.4.41-4ubuntu3.8) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
root@ip-10-50-1-4:/home/ubuntu# ab -n 2000 -c 10 http://10.50.0.46/index.html
```

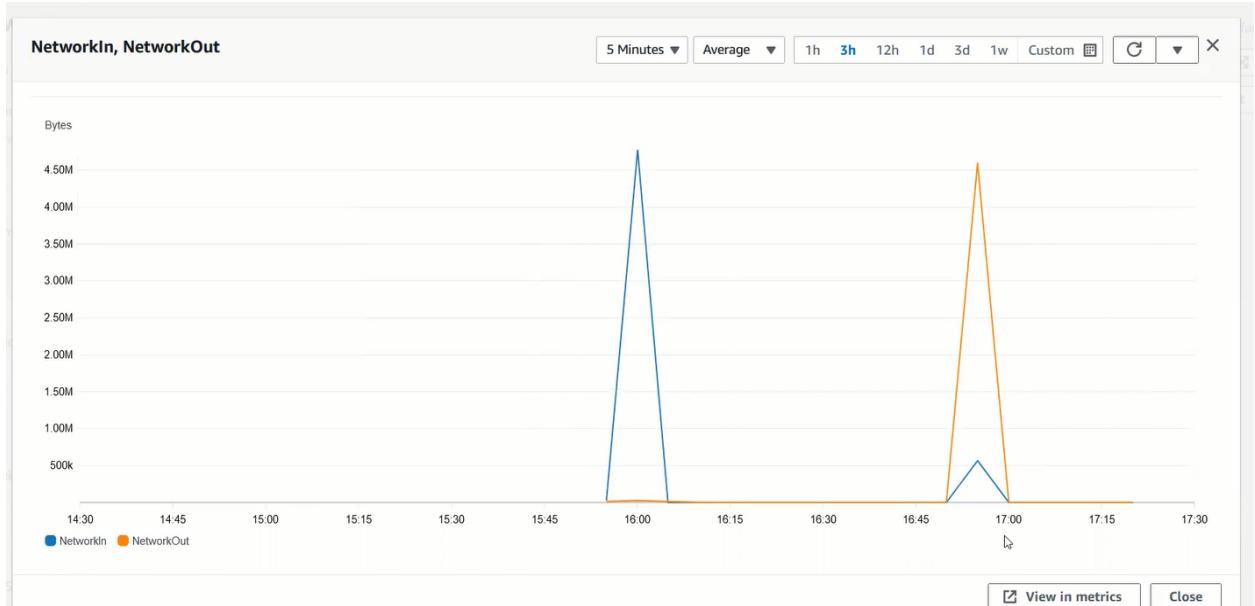
This is the index html file which is created by the apache web server.

```
root@ip-10-50-0-46: /var/www/html
[
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Apache2 Ubuntu Default Page: It works</title>
<style type="text/css" media="screen">
* {
    margin: 0px 0px 0px 0px;
"index.html" 375L, 10918C
1,0-1          Top
```

## Networking Graphs:

VM1\_Instance Network in – Blue line

VM1\_Instance Network out- Orange line



## Motivation

We had chosen Line widget in the metrics to represent “Network-In” and “Network-Out” as it is easy to analyze the outcomes.