# Galera & Xpand Assignment

## 1. Setup Master-Slave Replication b/ two mysql server.

STEPS:

Step 1: Create two virtual machines and install MariabDB on it.

- sudo apt update && sudo apt upgrade
- sudo apt -y install software-properties-common
- Import mariaDB GPG key
  sudo apt-key adv –fetch-keys ‘https://mariadb.org/mariadb_release_signing_key.asc’
- Add the mariaDB apt repository
  Sudo add-apt-repository ‘deb[arch=amd64,arm64,ppc64el] https://archive.mariadb.org/mariadb-10.5.6/repo/ubuntu/ focal main’

```
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo apt -y install software-properties
-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.99.9.8).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo apt-key adv --fetch-keys 'https://
mariadb.org/mariadb_release_signing_key.asc'
Executing: /tmp/apt-key-gpghome.ACkCxIlF9J/gpg.1.sh --fetch-keys https://mariad
b.org/mariadb_release_signing_key.asc
gpg: requesting key from 'https://mariadb.org/mariadb_release_signing_key.asc'
gpg: key F1656F24C74CD1D8: public key "MariaDB Signing Key <signing-key@mariadb
.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo add-apt-repository 'deb [arch=amd6
4,arm64,ppc64el] http://archive.mariadb.org/mariadb-10.5.6/repo/ubuntu/ focal m
ain'
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
H  Terminal  p://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Htt:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 Packages [970 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal/main i386 Packages [718 kB]
```

- Install mariaDB server
  sudo apt update
  sudo apt install mariadb-server mariadb-client

```
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo apt install mariadb-server mariadb
-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-4 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libdbd-mariadb-perl
  libdbi-perl libfcgi-perl libhtml-template-perl libmariadb3 libreadline5
  libsigsegv2 libterm-readkey-perl mariadb-client-10.5
  mariadb-client-core-10.5 mariadb-common mariadb-server-10.5
  mariadb-server-core-10.5 socat
```

- Secure the mariadb server using
  sudo mysql_secure_installation

```
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.
 Help
Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.
```

Step 2: Configure Master Server on one virtual machine

- Open the mariadb config file and configure the following changes .

```
  GNU nano 4.8          /etc/mysql/mariadb.conf.d/50-server.cnf
[mysqld]

#
# * Basic Settings
#

user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
basedir                 = /usr
#datadir                 = /var/lib/mysql
#tmpdir                  = /tmp
#lc-messages-dir         = /usr/share/mysql
#lc-messages             = en_US
#skip-external-locking
```

Ubuntu Software

```
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

```
  GNU nano 4.8          /etc/mysql/mariadb.conf.d/50-server.cnf
# https://mariadb.com/kb/en/innodb-system-variables/#innodb_buffer_pool_size
#innodb_buffer_pool_size = 8G

# this is only for embedded server
[embedded]

# This group is only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

# This group is only read by MariaDB-10.5 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mariadb-10.5]

server-id = 1
log_bin = /var/log/mysql/mysql-bin.log
log_bin_index = /var/log/mysql/mysql-bin.log.index
relay_log = /var/log/mysql/mysql-relay-bin
relay_log_index = /var/log/mysql/mysql-relay-bin.index
```

- After doing the changes , restart the server.
  sudo systemctl restart mariadb

```
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo nano /etc/mysql/mariadb.conf.d/50-
server.cnf
ubuntumaster@ubuntumaster-VirtualBox:~$ sudo systemctl restart mariadb
```

- Start the mariadb shell using sudo mysql -u root -p and create a user and grant replication privileges.

Do show master status and get the details of file and position remembered to use it in the slave server.

```
MariaDB [(none)]> CREATE USER 'replication'@'%' identified by 'root';
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%';
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Step 3: Configure Slave Server on the Virtual Machine

- Open the mariadb config file and do the configuration.

```
user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
basedir                 = /usr
#datadir                 = /var/lib/mysql
#tmpdir                  = /tmp
#lc-messages-dir         = /usr/share/mysql
#lc-messages             = en_US
#skip-external-locking

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0

#
# * Fine Tuning
#
#                       = 128M
#max_allowed_packet     = 1G
```

Software Updater

```
# Most important is to give InnoDB 80 % of the system RAM for buffer use:
# https://mariadb.com/kb/en/innodb-system-variables/#innodb_buffer_pool_size
#innodb_buffer_pool_size = 8G

# this is only for embedded server
[embedded]

# This group is only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

# This group is only read by MariaDB-10.5 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mariadb-10.5]

server-id = 2
log_bin = /var/log/mysql/mysql-bin.log
log_bin_index = /var/log/mysql/mysql-bin.log.index
relay_log = /var/log/mysql/mysql-relay-bin
relay_log_index = /var/log/mysql/mysql-relay-bin.index
```

- Restart mariadb service and run the mariadb shell and perform the operation to setup the slave to replicate the master.

```
slave@slave-VirtualBox:~$ sudo systemctl restart mariadb
slave@slave-VirtualBox:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.5.15-MariaDB-1:10.5.15+maria~focal-log mariadb.org binary di
stribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> stop slave;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST = '10.0.2.15', MASTER_USER ='rep
lication', MASTER_PASSWORD='root', MASTER_LOG_FILE='mysql-bin.000002', MASTER_L
OG_POS = 342;
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> start slave;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
```

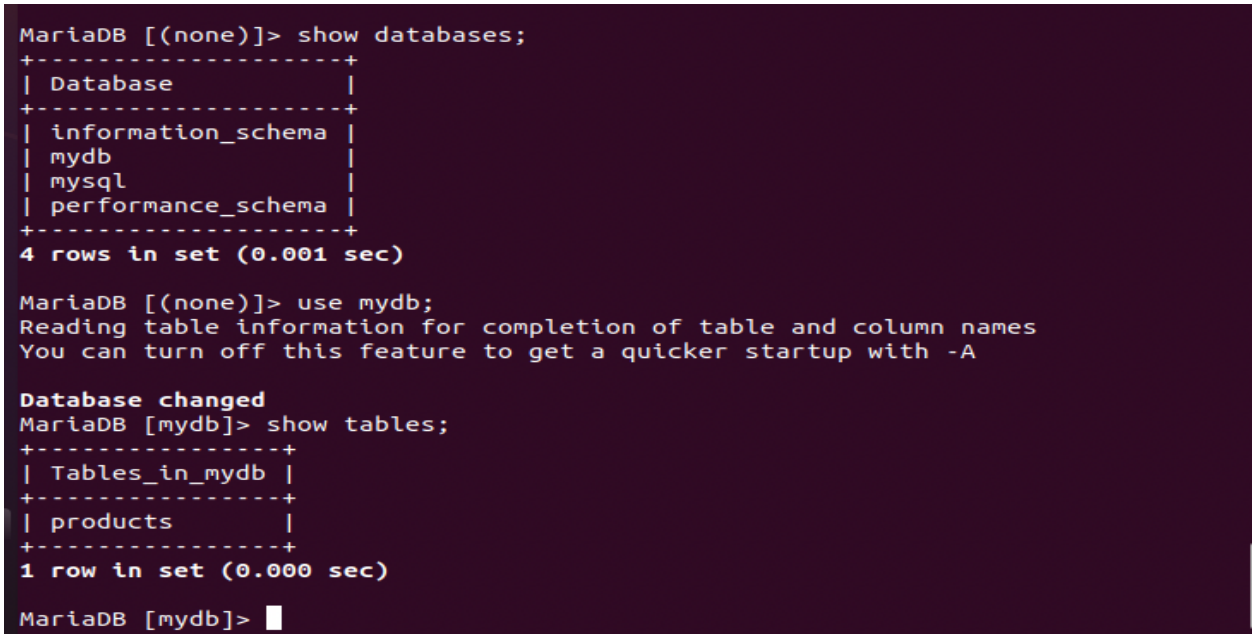
The Master-Slave replication has been created.
Test the Master-Slave Replication with a test database.

Create a database and create a table inside it on the master server.
On the mariadb shell
Create database.
Create a table inside it.

Now go to the slave server .
Check the slave status on the slave server.

```
MariaDB [(none)]> SHOW SLAVE STATUS \G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.0.2.15
                  Master_User: replication
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000002
          Read_Master_Log_Pos: 765
               Relay_Log_File: mysql-relay-bin.000002
                Relay_Log_Pos: 978
        Relay_Master_Log_File: mysql-bin.000002
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 765
              Relay_Log_Space: 1287
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
```

Use query show database there and you can have your database replicated to your slave server.




## 2. Setup a Galera Cluster with 3 nodes on version 10.5.13 and the sst method should be mariabackup.

STEPS:

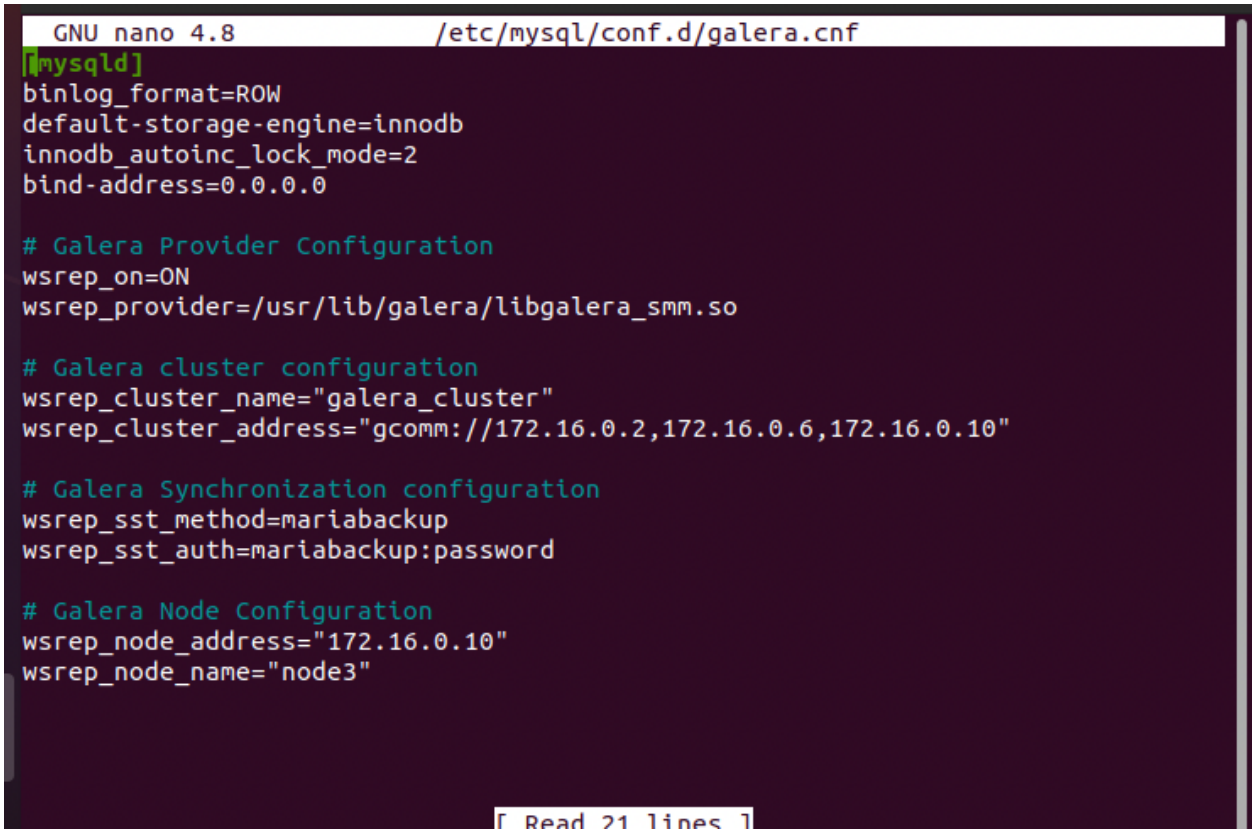
1. Setup 3 VM's and install mariadb on all the three VM's.
2. Create a galera.cnf file is doesn't exist in the following path /etc/mysql/conf.d/galera.cnf
   NODE 1:

```
  GNU nano 4.8                /etc/mysql/conf.d/galera.cnf
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_node=2
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

# Galera Cluster Configuration
wsrep_cluster_name="galera_cluster"
wsrep_cluster_address="gcomm://172.16.0.2,172.16.0.6,172.16.0.10"

# Galera Synchronization configuration
wsrep_sst_method=mariabackup
wsrep_sst_auth=mariabackup:password

# Galera Node Configuration
wsrep_node_address="172.16.0.2"
wsrep_node_name="node1"
```

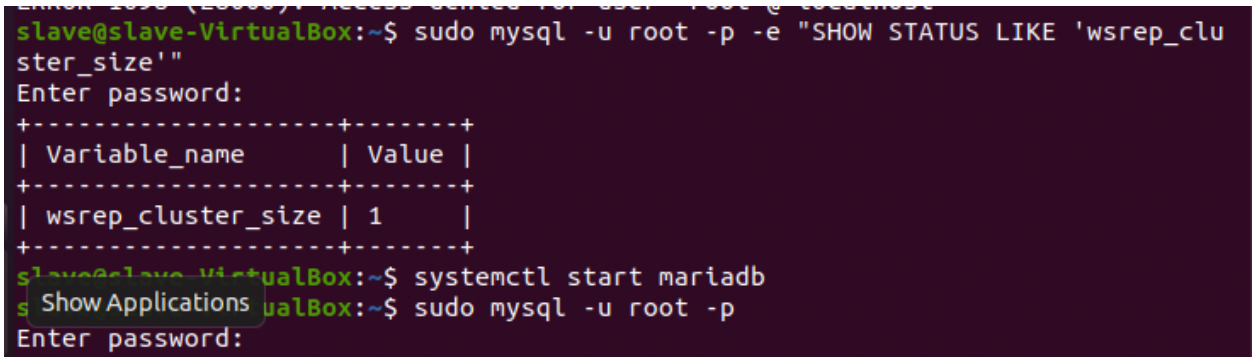
NODE 2:

```
  GNU nano 4.8                /etc/mysql/conf.d/galera.cnf
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera Provider Cofiguration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

# Galera Cluster Configuration
wsrep_cluster_name="galera_cluster"
wsrep_cluster_address="gcomm://172.16.0.2,172.16.0.6,172.16.0.10"
# Glera synchronization Configuration
wsrep_sst_method=mariabackup
wsrep_sst_auth=mariabackup:password

# Galera Node Configuration
wsrep_node_address="172.16.0.6"
wsrep_node_name="node2"


                        [ Read 21 lines ]
```

NODE 3:




3. Stop mariadb on all the nodes using systemctl stop mariadb.
4. Start the cluster on 1st node.



5. Create a user with the same name and password as in the cluster configuration file on node1 and grant replication privileges.

```
MariaDB [(none)]> CREATE USER 'mariabackup' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.008 sec)

MariaDB [(none)]> GRANT RELOAD, LOCK TABLES, PROCESS, REPLICATION CLIENT ON *.*
 TO 'mariabackup';
Q Files OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> select user,host from mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| mariabackup | %         |
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-------------+-----------+
4 rows in set (0.001 sec)

MariaDB [(none)]> show grants for 'mariabackup'
    -> ;
+-----------------------------------------------------------------------------------
-----------------------------------------------------------------+
| Grants for mariabackup@%
                                                                 |
+-----------------------------------------------------------------------------------
-----------------------------------------------------------------+
| GRANT RELOAD, PROCESS, LOCK TABLES, BINLOG MONITOR ON *.* TO `mariabackup`@`%
```

6. Start mariadb on 2nd node .
   Sudo mysql -u root -p -e “SHOW STATUS LIKE ‘wsrep_cluster_size’”

```
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 2     |
+--------------------+-------+
```

7. Do the same for 3rd node also.
   Sudo mysql -u root -p -e “SHOW STATUS LIKE ‘wsrep_cluster_size’”

```
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 3     |
+--------------------+-------+
```

8. The replication cluster is set up now.

To check if the replication is done . Create a database on Node1 and check its replication on the Node2 and Node3.

Database on node 1

```
MariaDB [(none)]> create database testCluster;
Query OK, 1 row affected (0.009 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| testCluster        |
+--------------------+
4 rows in set (0.000 sec)

MariaDB [(none)]>
```

Replication on other 2 nodes:

node-2

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| testCluster        |
+--------------------+
4 rows in set (0.000 sec)
```

node-3

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| testCluster        |
+--------------------+
4 rows in set (0.000 sec)
```

## 3. A sql file is attached, that has the nginx logs in a table, you will have to restore on the above setup and provide the below asked information.

Steps:

1. Create a database

```
MariaDB [mydb]> create database nginx_log;
Query OK, 1 row affected (0.000 sec)

MariaDB [mydb]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| mysql              |
| nginx_log          |
| performance_schema |
+--------------------+
5 rows in set (0.000 sec)

MariaDB [mydb]>
```

2. Import the sql file .

```
master@master-VirtualBox:~$ sudo mysql -u root -p nginx_log < ./Desktop/ngnix_a
ccess_log.sql
[sudo] password for master:
Enter password:
```

3. Perform all the queries on the sql log file.

a) Summary for the day/week/month:

- Highest requested host

```
MariaDB [nginx_log]> select host, count(host) as times from ngnix_access_log gr
oup by host order by times desc limit 1;
+----------------+-------+
| host           | times |
+----------------+-------+
| prod.ppops.pm5 | 11091 |
+----------------+-------+
1 row in set (0.116 sec)

MariaDB [nginx_log]>
```

- Highest requested upstream_ip

```
MariaDB [nginx_log]> select LEFT(upstream_ip_port,LOCATE(':',upstream_ip_port)-
1) as upstream_ip , count(LEFT(upstream_ip_port,LOCATE(':',upstream_ip_port)-1)
) as times from ngnix_access_log group by upstream_ip order by times desc limit
 1;
+-------------+-------+
| upstream_ip | times |
+-------------+-------+
| 10.77.22.11 |  4844 |
+-------------+-------+
1 row in set (0.101 sec)
```

- Highest requested path (upto 2 subdirectories ex : /check/balance)

```
MariaDB [nginx_log]> select LEFT(path,LOCATE('/',path,2)+LOCATE('/',path,3)-1)
as path2sub , count(LEFT(path,LOCATE('/',path,2)+LOCATE('/',path,3)-1)) as NoOf
Time from ngnix_access_log group by path2sub order by NoOfTime desc limit 1;
+---------------+----------+
| path2sub      | NoOfTime |
+---------------+----------+
| /myapi/consum |     9151 |
+---------------+----------+
1 row in set (0.145 sec)

MariaDB [nginx_log]>
```

b) Total requests per status code

```
MariaDB [nginx_log]> select statusCode , count(statusCode) as frequency from ng
nix_access_log group by statusCode order by frequency
    -> ;
+------------+-----------+
| statusCode | frequency |
+------------+-----------+
| 504        |       414 |
| 401        |       419 |
| 405        |       440 |
| 301        |       464 |
| 201        |       500 |
| 501        |       501 |
| 304        |       510 |
| 403        |       519 |
| 503        |       519 |
| 406        |       521 |
| 500        |       521 |
| 404        |       522 |
| 400        |       522 |
| 416        |       533 |
| 100        |       548 |
| 205        |       570 |
| 203        |       571 |
| 302        |       585 |
| 204        |       626 |
| 502        |       641 |
| 200        |     43826 |
+------------+-----------+
21 rows in set (0.076 sec)
```

c) Top requests

- Top 5 requests by upstream_ip

```
MariaDB [nginx_log]> select LEFT(upstream_ip_port,LOCATE(':',upstream_ip_port)-
1) as ip,count(LEFT(upstream_ip_port,LOCATE(':',upstream_ip_port)-1)) as freque
ncy from ngnix_access_log group by ip order by frequency desc limit 5;
+-------------+-----------+
| ip          | frequency |
+-------------+-----------+
| 10.77.22.14 |         7 |
| 10.77.23.10 |         7 |
| 10.77.23.11 |         7 |
| 10.77.23.12 |         7 |
| 10.77.22.11 |         7 |
+-------------+-----------+
5 rows in set, 1 warning (0.044 sec)
```

- Top 5 requests by host

```
MariaDB [nginx_log]> select host , count(host) as frequncy from ngnix_access_lo
g group by host order by frequncy desc limit 5;
+----------------------+----------+
| host                 | frequncy |
+----------------------+----------+
| prod.ppops.pm5       |    11091 |
| appone.ppops.com     |    11062 |
| api.ppops.com        |    10815 |
| apptwo.ppops.com     |    10713 |
| apptwo-new.ppops.com |    10591 |
+----------------------+----------+
5 rows in set (0.101 sec)
```

- Top 5 requests by bodyBytesSent

```
MariaDB [nginx_log]> select bodyBytesSent , count(bodyBytesSent) as frequency f
rom ngnix_access_log group by bodyBytesSent order by frequency desc limit 5;
+---------------+-----------+
| bodyBytesSent | frequency |
+---------------+-----------+
| 38            |       179 |
| 70            |       160 |
| 51            |       158 |
| 61            |       157 |
| 60            |       157 |
+---------------+-----------+
5 rows in set (0.263 sec)
```

- Top 5 requests by path

```
MariaDB [nginx_log]> select LEFT(path,LOCATE('/',path,2)+LOCATE('/',path,3)-1)
as pathh,count(LEFT(path,LOCATE('/',path,2)+LOCATE('/',path,3)-1)) as frequency
 from ngnix_access_log group by pathh order by frequency desc limit 5;
+--------------+-----------+
| pathh        | frequency |
+--------------+-----------+
| /myapi/consum |      9151 |
| /check/balanc |      9041 |
| /myapi/mercha |      9010 |
| /myapi/kyc/co |       431 |
| /myapi/kyc/in |       420 |
+--------------+-----------+
5 rows in set (0.150 sec)
```

- Top 5 requests with the highest response time.

```
MariaDB [nginx_log]> select * from ngnix_access_log order by responseTime desc
limit 5;
+-----------------+----------------------------+------------+-----------------
----------------------+-------------+-----------+-------------+-------------
-----+---------------+----------+-----------------------------------------------
-----------------------------------------------------+-------------------
-----------------+-------------------------------------------------------+-----
-----------+
| ip              | time                       | httpMethod | path
                     | httpVersion | statusCode | responseTime | upstream_ip_
port | bodyBytesSent | referrer | userAgent
                                                                 | ssl_protocol
                 | content_type                                                | host
          |
+-----------------+----------------------------+------------+-----------------
----------------------+-------------+-----------+-------------+-------------
-----+---------------+----------+-----------------------------------------------
-----------------------------------------------------+-------------------
-----------------+-------------------------------------------------------+-----
-----------+
| 132.175.122.124 | 06/Mar/2021:07:31:52 +5270 | PUT        | /myapi/consumers/
software%20Right-sized | HTTP/1.1    | 200        | 9.999        | 10.77.27.14:
443   | 2343          | -        | Mozilla/5.0 (X11; Linux i686) AppleWebKit/534
0 (KHTML, like Gecko) Chrome/37.0.810.0 Mobile Safari/5340 | TLSv1.2/ECDHE-RSA-
AES256-GCM-SHA384 | 62b923587b05d59b-BOM application/json; charset=utf-8 | prod
.ppops.pm5 |
| 132.175.122.124 | 06/Mar/2021:08:31:52 +5270 | PUT        | /myapi/consumers/
software%20Right-sized | HTTP/1.1    | 200        | 9.999        | 10.77.27.14:
443   | 2343          | -        | Mozilla/5.0 (X11; Linux i686) AppleWebKit/534
```

- Get top 5 requests returning 200/5xx/4xx per host

```
MariaDB [nginx_log]> select host , count(host) as freq from ngnix_access_log wh
ere statusCode LIKE '5%' or statusCode = 200 or statusCode LIKE '4%' group by h
ost order by freq desc limit 5;
+----------------------+-------+
| host                 | freq  |
+----------------------+-------+
| prod.ppops.pm5       | 10251 |
| appone.ppops.com     | 10220 |
| api.ppops.com        |  9919 |
| apptwo.ppops.com     |  9838 |
| apptwo-new.ppops.com |  9670 |
+----------------------+-------+
5 rows in set (0.102 sec)
```

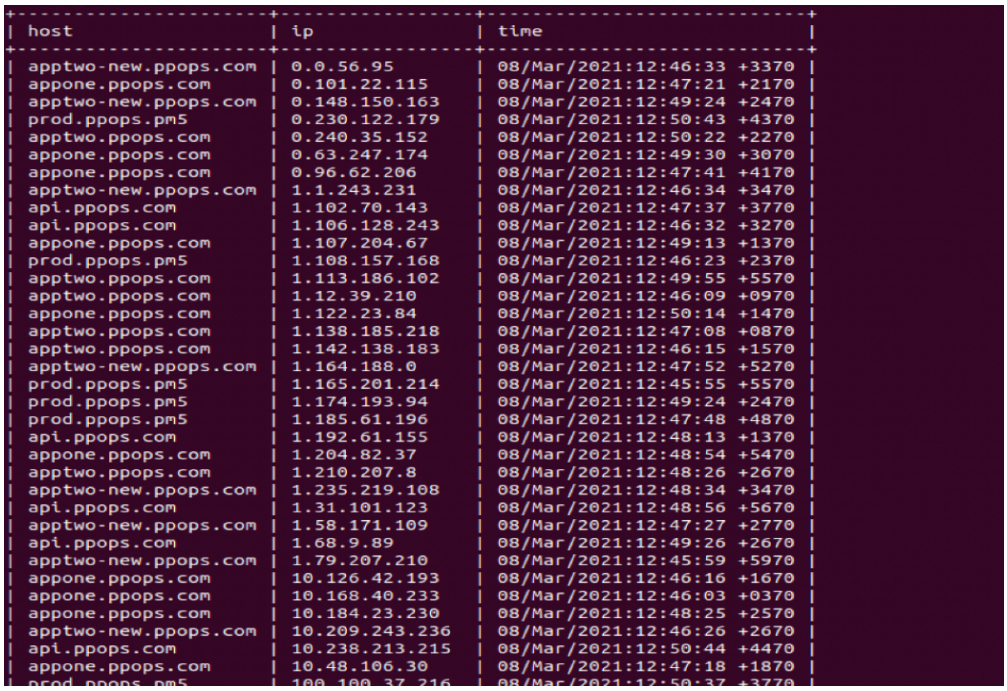


d) Find the time the last 200/5xx/4xx was received for a particular host.

```
MariaDB [nginx_log]> select host, right(left(time,LOCATE(' ', time)), LOCATE(':
',left(time,locate(' ',time)),3)-3) as last,statusCode from ngnix_access_log wh
ere statusCode like '5%' or statusCode = '200' or statusCode like '4%' group by
 host order by time;
+----------------------+----------+------------+
| host                 | last     | statusCode |
+----------------------+----------+------------+
| api.ppops.com        | 07:16:58 | 200        |
| apptwo.ppops.com     | 07:18:51 | 500        |
| prod.ppops.pm5       | 07:27:39 | 200        |
| appone.ppops.com     | 07:31:36 | 200        |
| apptwo-new.ppops.com | 16:46:33 | 200        |
+----------------------+----------+------------+
5 rows in set (0.093 sec)
```

e) Get all request for the last 10 minutes

```
select host, ip, time from ngnix_access_log where
str_to_date(left(left(time,locate(' ', time)), locate(':', left(time,locate(' ',
time)))-1),'%d/%b/%Y') = (select str_to_date(left(left(time,locate(' ', time)),
locate(':', left(time,locate(' ', time)))-1),'%d/%b/%Y') as date from
ngnix_access_log order by date desc, str_to_date(right(left(time,locate(' ', time)),
locate(':', left(time,locate(' ', time)))-3),'%H:%i:%s') desc limit 1) and
  str_to_date(right(left(time,locate(' ', time)), locate(':', left(time,locate(' ',
time)))-3),'%H:%i:%s')>=(select str_to_date(right(left(time,locate(' ', time)),
locate(':', left(time,locate(' ', time)))-3),'%H:%i:%s') as tiime from
ngnix_access_log order by str_to_date(left(left(time,locate(' ', time)), locate(':',
left(time,locate(' ', time)))-1),'%d/%b/%Y') desc, tiime desc limit 1)  - interval 10
minute;
```



| host | ip | time |
|---|---|---|
| apptwo-new.ppops.com | 0.0.56.95 | 08/Mar/2021:12:46:33 +3370 |
| appone.ppops.com | 0.101.22.115 | 08/Mar/2021:12:47:21 +2170 |
| apptwo-new.ppops.com | 0.148.150.163 | 08/Mar/2021:12:49:24 +2470 |
| prod.ppops.pm5 | 0.230.122.179 | 08/Mar/2021:12:50:43 +4370 |
| apptwo.ppops.com | 0.240.35.152 | 08/Mar/2021:12:50:22 +2270 |
| appone.ppops.com | 0.63.247.174 | 08/Mar/2021:12:49:30 +3070 |
| appone.ppops.com | 0.96.62.206 | 08/Mar/2021:12:47:41 +4170 |
| apptwo-new.ppops.com | 1.1.243.231 | 08/Mar/2021:12:46:34 +3470 |
| api.ppops.com | 1.102.70.143 | 08/Mar/2021:12:47:37 +3770 |
| api.ppops.com | 1.106.128.243 | 08/Mar/2021:12:46:32 +3270 |
| appone.ppops.com | 1.107.204.67 | 08/Mar/2021:12:49:13 +1370 |
| prod.ppops.pm5 | 1.108.157.168 | 08/Mar/2021:12:46:23 +2370 |
| apptwo.ppops.com | 1.113.186.102 | 08/Mar/2021:12:49:55 +5570 |
| apptwo.ppops.com | 1.12.39.210 | 08/Mar/2021:12:46:09 +0970 |
| appone.ppops.com | 1.122.23.84 | 08/Mar/2021:12:50:14 +1470 |
| apptwo.ppops.com | 1.138.185.218 | 08/Mar/2021:12:47:08 +0870 |
| apptwo.ppops.com | 1.142.138.183 | 08/Mar/2021:12:46:15 +1570 |
| apptwo-new.ppops.com | 1.164.188.0 | 08/Mar/2021:12:47:52 +5270 |
| prod.ppops.pm5 | 1.165.201.214 | 08/Mar/2021:12:45:55 +5570 |
| prod.ppops.pm5 | 1.174.193.94 | 08/Mar/2021:12:49:24 +2470 |
| prod.ppops.pm5 | 1.185.61.196 | 08/Mar/2021:12:47:48 +4870 |
| api.ppops.com | 1.192.61.155 | 08/Mar/2021:12:48:13 +1370 |
| appone.ppops.com | 1.204.82.37 | 08/Mar/2021:12:48:54 +5470 |
| apptwo.ppops.com | 1.210.207.8 | 08/Mar/2021:12:48:26 +2670 |
| apptwo-new.ppops.com | 1.235.219.108 | 08/Mar/2021:12:48:34 +3470 |
| api.ppops.com | 1.31.101.123 | 08/Mar/2021:12:48:56 +5670 |
| apptwo-new.ppops.com | 1.58.171.109 | 08/Mar/2021:12:47:27 +2770 |
| api.ppops.com | 1.68.9.89 | 08/Mar/2021:12:49:26 +2670 |
| apptwo-new.ppops.com | 1.79.207.210 | 08/Mar/2021:12:45:59 +5970 |
| appone.ppops.com | 10.126.42.193 | 08/Mar/2021:12:46:16 +1670 |
| appone.ppops.com | 10.168.40.233 | 08/Mar/2021:12:46:03 +0370 |
| appone.ppops.com | 10.184.23.230 | 08/Mar/2021:12:48:25 +2570 |
| apptwo-new.ppops.com | 10.209.243.236 | 08/Mar/2021:12:46:26 +2670 |
| api.ppops.com | 10.238.213.215 | 08/Mar/2021:12:50:44 +4470 |
| appone.ppops.com | 10.48.106.30 | 08/Mar/2021:12:47:18 +1870 |
| prod.ppops.pm5 | 100.100.37.216 | 08/Mar/2021:12:50:37 +3770 |

f) Get all the requests taking more than 2/5/10 sec to respond.

- More than 2 secs
  Select host, ip, responseTime from nginx_access_log where responseTime>2;

```
MariaDB [nginx_log]> select host,ip , responseTime from ngnix_access_log where
responseTime>2;
```

| host | ip | responseTime |
|---|---|---|
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo.ppops.com | 0.105.114.213 | 9.437 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| apptwo-new.ppops.com | 0.107.38.97 | 9.093 |
| appone.ppops.com | 0.116.70.167 | 6.674 |
| appone.ppops.com | 0.150.51.80 | 17.656 |
| appone.ppops.com | 0.150.51.80 | 17.656 |
| appone.ppops.com | 0.150.51.80 | 17.656 |
| appone.ppops.com | 0.150.51.80 | 17.656 |
| appone.ppops.com | 0.150.51.80 | 17.656 |
| appone.ppops.com | 0.150.51.80 | 17.656 |

- More than 5 secs

Select host, ip, responseTime from nginx_access_log where responseTime>5;

```
MariaDB [nginx_log]> select host,ip,responseTime from ngnix_access_log where re
sponseTime>5;
+----------------------+-----------------+--------------+
| host                 | ip              | responseTime |
+----------------------+-----------------+--------------+
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo.ppops.com     | 0.105.114.213   | 9.437        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| apptwo-new.ppops.com | 0.107.38.97     | 9.093        |
| appone.ppops.com     | 0.116.70.167    | 6.674        |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
```

- More than 10 secs

  Select host, ip, responseTime from nginx_access_log where responseTime>10;

```
MariaDB [nginx_log]> select host,ip , responseTime from ngnix_access_log where
responseTime>10;
+----------------------+-----------------+--------------+
| host                 | ip              | responseTime |
+----------------------+-----------------+--------------+
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| appone.ppops.com     | 0.150.51.80     | 17.656       |
| api.ppops.com        | 0.166.73.137    | 15.249       |
| api.ppops.com        | 0.166.73.137    | 15.249       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| apptwo.ppops.com     | 0.179.116.77    | 11.321       |
| prod.ppops.pm5       | 0.230.122.179   | 13.408       |
| prod.ppops.pm5       | 0.230.122.179   | 13.408       |
| prod.ppops.pm5       | 0.230.122.179   | 13.408       |
| apptwo-new.ppops.com | 0.58.166.58     | 11.933       |
| apptwo-new.ppops.com | 0.58.166.58     | 11.933       |
```

g) Get all the requests in the specified timestamp (Ex: from 06/Mar/2021:04:48 to 06/Mar/2021:04:58)

```
MariaDB [nginx_log]> select host,ip,time from ngnix_access_log where ( str_to_d
ate(left(left(time,locate(' ',time)),locate(':',left(time,locate(' ',time)))-1)
,'%d/%b/%Y') >=  str_to_date(left(left('06/Mar/2021:04:48:00 ',locate(' ','06/M
ar/2021:04:48:00 ')),locate(':',left('06/Mar/2021:04:48:00 ',locate(' ','06/Mar
/2021:04:48:00 ')))-1),'%d/%b/%Y') and str_to_date(right(left(time,locate(' ',ti
me)),locate(':',left(time,locate(' ',time)))-3),'%H:%i:%s') >= str_to_date(righ
t(left('06/Mar/2021:04:48:00 ',locate(' ','06/Mar/2021:04:48:00 ')),locate(':',
left('06/Mar/2021:04:48:00 ',locate(' ','06/Mar/2021:04:48:00 ')))-3),'%H:%i:%s
') ) and ( str_to_date(left(left(time,locate(' ',time)),locate(':',left(time,lo
cate(' ',time)))-1),'%d/%b/%Y') <= str_to_date(left(left('06/Mar/2021:04:58:00
',locate(' ','06/Mar/2021:04:58:00 ')),locate(':',left('06/Mar/2021:04:58:00 ',
locate(' ','06/Mar/2021:04:58:00 ')))-1),'%d/%b/%Y') and str_to_date(right(left
(time,locate(' ',time)),locate(':',left(time,locate(' ',time)))-3),'%H:%i:%s')
<= str_to_date(right(left('06/Mar/2021:04:58:00 ',locate(' ','06/Mar/2021:04:58
:00 ')),locate(':',left('06/Mar/2021:04:58:00 ',locate(' ','06/Mar/2021:04:58:0
0 ')))-3),'%H:%i:%s') );
Empty set (0.153 sec)
```

h) Create partitioning on this table using the time values, the table should have weekly partitions.
ALTER TABLE ngnix_access_log PARTITION BY RANGE(time)
INTERVAL(NUMTODSINTERVAL(7,'DAYS'))
( PARTITION p0 VALUES less than (select str_to_date(left(left(time,locate(' ', time)), locate(':', left(time,locate(' ', time)))-1),'%d/%b/%Y') as date from ngnix_access_log order by date LIMIT 1));

i) Truncate the partitions from week 21 to week 25
ALTER TABLE ngnix_access_log TRUNCATE PARTITIONS p20, p21, p22, p23, p24;