

SOP: Technical KenyaEMR3.x Server Optimization Userguide

(Last update: Mar 2024)

TASK:	Technical KenyaEMR Server Optimization
OBJECTIVE:	To improve server performance by following some configuration steps
WHO:	Facility System Admin, Partner supporting site HIS staff
REQUIRED MATERIALS:	Functional KenyaEMR

Background:

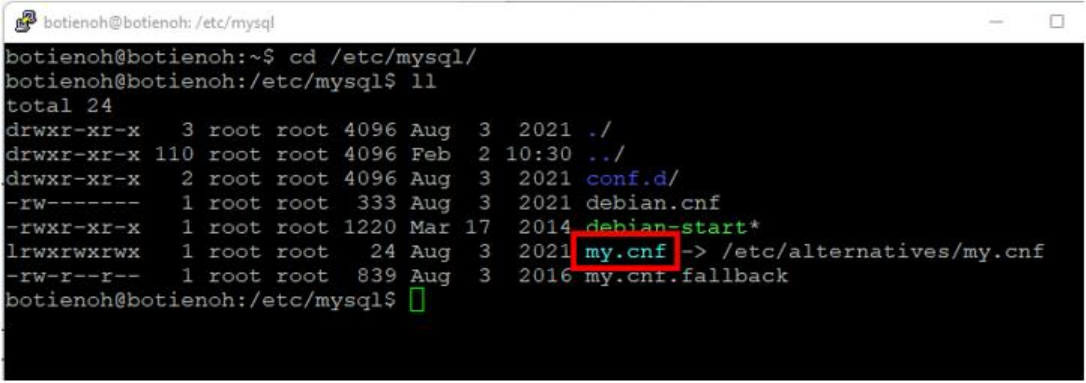
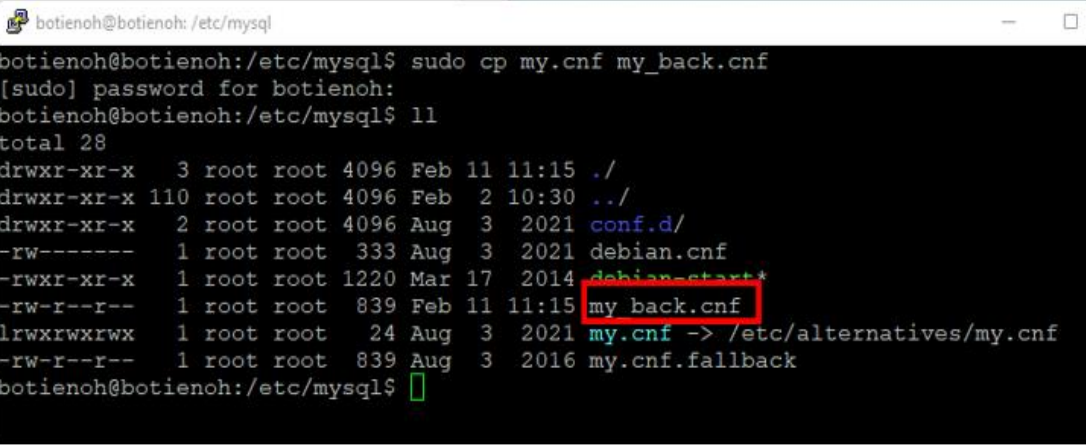
Many facilities, especially high-volume facilities, have reported EMR server performance issues when accessing their KenyaEMR instance. Some of the performance issues reported include slow response when retrieving patient information, slow execution of reports and in some adverse cases, users experienced a lag while logging into the system. While some of these issues can be attributed to structural design of EMR, most of them are as result of misconfigurations in the EMR dependencies (Java, MySQL, Tomcat). This guide outlines the requisite configuration that may help improve your EMR performance for a better user experience.

Objective:

To outline various EMR configuration steps and checks that would result in better server performance.

SECTION 1: MYSQL CONFIGURATIONS

Sometimes MySQL instance does not pick the configurations that you saved in **my.cnf** file. This has been noted often on Ubuntu 20.04. Check to be sure all the configurations are properly set and are being picked up.

<p>i. MySQL configurations</p> <ul style="list-style-type: none"> - Open Terminal - Navigate to mysql directory and open the my.cnf file <p>You should be able to see the file as shown.</p>	 <pre> botienoh@botienoh: /etc/mysql botienoh@botienoh:~\$ cd /etc/mysql/ botienoh@botienoh:/etc/mysql\$ ll total 24 drwxr-xr-x 3 root root 4096 Aug 3 2021 ./ drwxr-xr-x 110 root root 4096 Feb 2 10:30 ../ drwxr-xr-x 2 root root 4096 Aug 3 2021 conf.d/ -rw----- 1 root root 333 Aug 3 2021 debian.cnf -rwxr-xr-x 1 root root 1220 Mar 17 2014 debian-start* lrwxrwxrwx 1 root root 24 Aug 3 2021 my.cnf -> /etc/alternatives/my.cnf -rw-r--r-- 1 root root 839 Aug 3 2016 my.cnf.fallback botienoh@botienoh:/etc/mysql\$ </pre>
<p>ii. Backup a copy of my.cnf file. i.e. sudo cp my.cnf my_back.cnf</p> <p>Check to make sure the backup is available</p>	 <pre> botienoh@botienoh: /etc/mysql botienoh@botienoh:/etc/mysql\$ sudo cp my.cnf my_back.cnf [sudo] password for botienoh: botienoh@botienoh:/etc/mysql\$ ll total 28 drwxr-xr-x 3 root root 4096 Feb 11 11:15 ./ drwxr-xr-x 110 root root 4096 Feb 2 10:30 ../ drwxr-xr-x 2 root root 4096 Aug 3 2021 conf.d/ -rw----- 1 root root 333 Aug 3 2021 debian.cnf -rwxr-xr-x 1 root root 1220 Mar 17 2014 debian-start* -rw-r--r-- 1 root root 839 Feb 11 11:15 my_back.cnf lrwxrwxrwx 1 root root 24 Aug 3 2021 my.cnf -> /etc/alternatives/my.cnf -rw-r--r-- 1 root root 839 Aug 3 2016 my.cnf.fallback botienoh@botienoh:/etc/mysql\$ </pre>

- iii. Delete the existing my.cnf file. This will ensure the symbolic link is also dropped.

```

botienoh@botienoh: /etc/mysql
botienoh@botienoh:/etc/mysql$ sudo rm my.cnf
botienoh@botienoh:/etc/mysql$ ll
total 32
drwxr-xr-x  3 root root 4096 Feb 11 11:24 ./
drwxr-xr-x 110 root root 4096 Feb  2 10:30 ../
drwxr-xr-x  2 root root 4096 Aug  3 2021 conf.d/
-rw-----  1 root root  333 Aug  3 2021 debian.cnf
-rwxr-xr-x  1 root root 1220 Mar 17 2014 debian-start*
-rw-r--r--  1 root root  839 Feb 11 11:15 my_back.cnf
-rw-r--r--  1 root root 4638 Feb 11 11:22 my.cnf.fallback
botienoh@botienoh:/etc/mysql$ █
  
```

- iv. Restore a copy of my.cnf file as follows:
 sudo cp my_back.cnf
 my.cnf

See Index for a sample config variable. Save the settings once done. Check to make sure **my.cnf** file is now available. Note that the symbolic link is not created at this time until MySQL service is restarted.

```

botienoh@botienoh: /etc/mysql
botienoh@botienoh:/etc/mysql$ ll
total 36
drwxr-xr-x  3 root root 4096 Feb 11 11:31 ./
drwxr-xr-x 110 root root 4096 Feb  2 10:30 ../
drwxr-xr-x  2 root root 4096 Aug  3 2021 conf.d/
-rw-----  1 root root  333 Aug  3 2021 debian.cnf
-rwxr-xr-x  1 root root 1220 Mar 17 2014 debian-start*
-rw-r--r--  1 root root  839 Feb 11 11:15 my_back.cnf
-rw-r--r--  1 root root 3823 Feb 11 11:31 my.cnf
-rw-r--r--  1 root root 4638 Feb 11 11:22 my.cnf.fallback
botienoh@botienoh:/etc/mysql$ █
  
```

<p>v. If configs already set, verify, and optimize the following variables based on the available system resources (i.e. Disk space and Physical memory)</p> <p>NB: Make the settings under InnoDB section</p>	<pre>[mysqld] innodb_flush_method =O_DIRECT innodb_file_per_table=1 innodb_buffer_pool_size=12G innodb_buffer_pool_instances=12 key_buffer_size=32M innodb_log_buffer_size=8M tmp_table_size=256M max_heap_table_size=256M table_definition_cache=2400 open_files_limit=65535 max_connections=500 max_allowed_packet=16M</pre>
<p>vi. Key settings to look at under InnoDB section.</p>	<ul style="list-style-type: none"> - max_connection: Refers to the number of simultaneous connections allowed to the database. You may want to adjust this if you get errors citing "<i>Too many connections</i>". - innodb_buffer_pool_size: This setting allocates system memory as a data cache for your database. If you have large chunks of data, increase this value (can take up-to 75% of RAM). Take note of the RAM required to run other system resources. - innodb_io_capacity: This variable sets the rate for input/output from your storage device. This is directly related to the type and speed of your storage drive. A 5400-rpm HDD will have a much lower capacity than a high-end SSD or Intel Optane. You can adjust this value to better match your hardware.

vii. To confirm that the settings have been picked, log into mysql and execute the following queries:

```
show variables like
'innodb_buffer%';
```

Note:

The **innodb_buffer_pool_instances** depend on the GB size allocated to **innodb_buffer_pool_size**. For instance, if the **innodb_pool_size** is allocated 1GB then the **innodb_pool_instances** is set to 1. 2 instances for 2GB, 3 instances for 3GB etc

```
mysql> show variables like 'innodb_buffer%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_buffer_pool_dump_at_shutdown | OFF |
| innodb_buffer_pool_dump_now | OFF |
| innodb_buffer_pool_filename | ib_buffer_pool |
| innodb_buffer_pool_instances | 8 |
| innodb_buffer_pool_load_abort | OFF |
| innodb_buffer_pool_load_at_startup | OFF |
| innodb_buffer_pool_load_now | OFF |
| innodb_buffer_pool_size | 4294967296 |
+-----+-----+
8 rows in set (0.00 sec)
```

innodb_buffer_pool_size is set to 4GB

SECTION 2: JAVA HEAP-SPACE SETTINGS

There are 5 important parameters to be set for Java Heap Space variable as follows:

- Xmx
- Xms
- XX: PermSize
- XX: MaxPermSize
- XX: NewSize

The memory allocation for these parameters depends on the available physical memory. A typical allocation for a 8GB server could be as follows:

Steps: Open terminal (Ctrl+T)
 sudo gedit /etc/default/tomcat9

```
# The home directory of the Java development kit (JDK). You need at least
# JDK version 8. If JAVA_HOME is not set, some common directories for
# OpenJDK and the Oracle JDK are tried.
#JAVA_HOME=/usr/lib/jvm/java-8-openjdk
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
# You may pass JVM startup parameters to Java here. If you run Tomcat with
# Java 8 instead of 9 or newer, add "-XX:+UseG1GC" to select a suitable GC.
# If unset, the default options will be: -Djava.awt.headless=true

JAVA_OPTS="${JAVA_OPTS} -Xmx2048m -Xms2048m -XX:PermSize=1024m -XX:MaxPermSize=1024m -XX:NewSize=512m"

# To enable remote debugging uncomment the following line.
# You will then be able to use a Java debugger on port 8000.
#JAVA_OPTS="${JAVA_OPTS} -agentlib:jdwp=transport=dt_socket,address=8000,server=y,suspend=n"

# Java compiler to use for translating JavaServer Pages (JSPs). You can use all
# compilers that are accepted by Ant's build.compiler property.
#JSP_COMPILER=javac
```

Xmx: 2048m Xms: 1024m -XX: PermSize = 512m -XX: MaxPermSize =512m -XX: NewSize = 512m

SECTION 3: BACKUP CONFIGURATIONS

Background

KenyaEMR database has been configured to self-backup automatically twice a day (11am and 4pm). However, after upgrading the system to Ubuntu 20.04LTS, users may have noticed that the backup frequency increased tremendously. This often creates numerous copies of the database per day that takes up the hard disk space and makes the server performance to degenerate. To correct this anomaly, System Admins need to update the backup frequency to the default twice a day.

The following manual provides a guide on how to correct this. Click [HERE](#) to open

SECTION 4: STOP ALL UNUSED SCHEDULERS

Schedulers are procedures that are set to execute a particular event at set intervals. System resources are committed to use during scheduler execution and depending on the type of scheduler, this can take up significant amount of resources in your server processing power. To ensure optimal use of server resources, it is recommended that you disable Schedulers that are not in use at your facility. For example, if your site is not using VDOT module, there is no reason to keep it running.

Steps:

- On the Developers Panel, click Legacy UI
- On the Admin Panel, click Manage Scheduler
- Under the list of Schedulers, identify and stop schedulers that the facility doesn't need.

Observations

[Manage Observations](#)

Scheduler

[Manage Scheduler](#)

Programs

[Manage Programs](#)

[Manage Triggered State Conversions](#)

NB: Ensure you also disable them from automatically starting when the system reboots as shown below.

Scheduler

[Task Configuration](#)

[Schedule](#)

[Properties](#)

Schedule

PLEASE NOTE: Changes to the schedule below are NOT passed onto task instances that are already running. After saving your changes, you MUST stop/start the desired task on the Task List page in order to pass these schedule changes onto that task.

Started: true

Start on startup: *Uncheck this*

Start time pattern:

Start time:

Repeat interval:

Last Execution Time: 14 February 2022 11:56:59 EAT

ANNEX: SAMPLE MYSQL CONFIG

```

# SAMPLE MYSQL CONFIG FILE
# =====
# Copyright (c) 2015, 2016, Oracle and/or its affiliates. All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License, version 2.0,
# as published by the Free Software Foundation.
#
# This program is also distributed with certain software (including
# but not limited to OpenSSL) that is licensed under separate terms,
# as designated in a particular file or component or in included license
# documentation. The authors of MySQL hereby grant you an additional
# permission to link the program and your derivative works with the
# separately licensed software that they have included with MySQL.
  
```

```
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License, version 2.0, for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
#
# The MySQL Server configuration file.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# * IMPORTANT: Additional settings that can override those from this file!
# The files must end with '.cnf', otherwise they'll be ignored.
#
[mysqld]
innodb_flush_method =O_DIRECT
innodb_file_per_table=1
innodb_buffer_pool_size=12G
innodb_buffer_pool_instances=12
thread_cache_size=50
query_cache_size=0
key_buffer_size=32M
innodb_log_buffer_size=8M
tmp_table_size=256M
max_heap_table_size=256M
table_definition_cache=2400
table_open_cache=2000
query_cache_type=0
open_files_limit=65535
max_connections=500
max_allowed_packet=16M

# adding bind address
bind-address = 0.0.0.0
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
```


*****THE END*****