

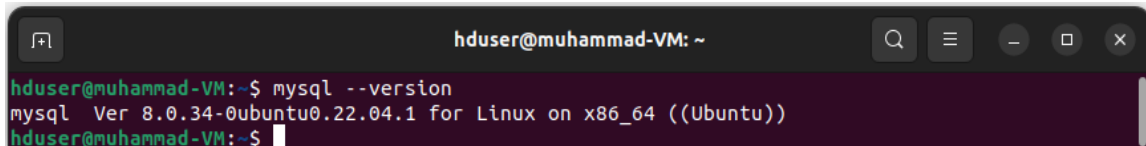
Tutorial 9 YCSB (MongoDB and MySQL)

Database Benchmarking

For benchmarking, we will use two databases (MySQL and MongoDB) in this tutorial and use a benchmarking tool (YCSB) to check their performance.

1) You can check installed version of MySQL by using the following command as

```
$mysql --version
```



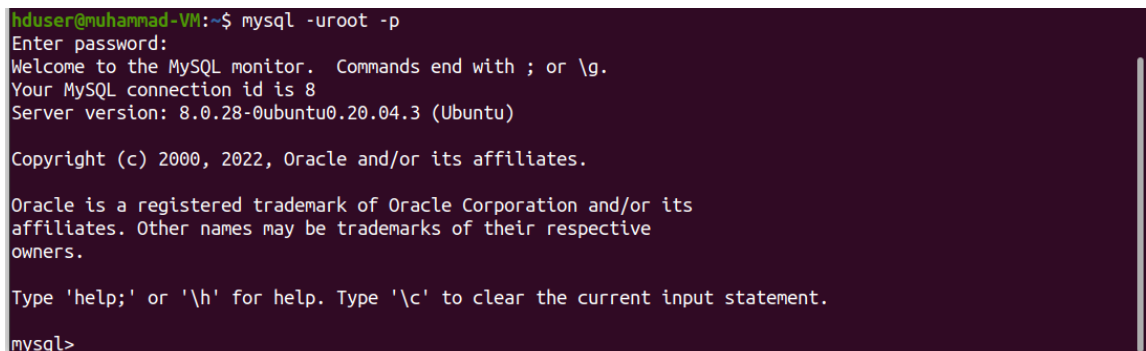
```
hduser@muhammad-VM: ~  
hduser@muhammad-VM:~$ mysql --version  
mysql Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))  
hduser@muhammad-VM:~$
```

If you could not find MySQL on your VM, then complete Tutorial 5 (part I) before step 2 starting. Otherwise, you can start from step 2.

2) Use password as **password** as we set in Tutorial 5 (part I).

```
$mysql -uroot -p
```

3) Create a database “**BenchTest**” and “**usertable**” table in MySQL. Create the database named as “BenchTest”. Then create the table named as “usertable”.



```
hduser@muhammad-VM:~$ mysql -uroot -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 8  
Server version: 8.0.28-0ubuntu0.20.04.3 (Ubuntu)  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

At the `mysql>` command prompt, enter the following command as mentioned as well as in the screen shot below

```
mysql>create database BenchTest;  
mysql>use BenchTest;  
mysql>CREATE TABLE usertable (YCSB_KEY VARCHAR(255) PRIMARY KEY,  
                                FIELD0 VARCHAR(255), FIELD1 VARCHAR(255),  
                                FIELD2 VARCHAR(255), FIELD3 VARCHAR(255),  
                                FIELD4 VARCHAR(255), FIELD5 VARCHAR(255),  
                                FIELD6 VARCHAR(255), FIELD7 VARCHAR(255),  
                                FIELD8 VARCHAR(255), FIELD9 VARCHAR(255));
```

```
hduser@muhammad-VM: ~  
mysql> create database BenchTest;  
Query OK, 1 row affected (0.02 sec)  
  
mysql> use BenchTest;  
Database changed  
mysql> CREATE TABLE usertable (YCSB_KEY VARCHAR(255) PRIMARY KEY,  
-> FIELD0 VARCHAR(255), FIELD1 VARCHAR(255),  
-> FIELD2 VARCHAR(255), FIELD3 VARCHAR(255),  
-> FIELD4 VARCHAR(255), FIELD5 VARCHAR(255),  
-> FIELD6 VARCHAR(255), FIELD7 VARCHAR(255),  
-> FIELD8 VARCHAR(255), FIELD9 VARCHAR(255));  
Query OK, 0 rows affected (0.04 sec)  
  
mysql> exit  
Bye  
hduser@muhammad-VM:~$
```

4) Download and install YCSB as mentioned below using the commands on Ubuntu shell as shown.

```
$pwd
```

```
$cd /home/hduser
```

```
hduser@muhammad-VM:~$ pwd  
/home/hduser  
hduser@muhammad-VM:~$ cd /home/hduser  
hduser@muhammad-VM:~$
```

Download the ycsb-0.17.0.tar.gz from the below mentioned link by using the commands on the Ubuntu terminal. Scroll down the webpage and you will find the command to download the YCSB.

← → ↻ github.com/brianfrankcooper/YCSB

☰ README.md

Links

- To get here, use <https://ycsb.site>
- [Our project docs](#)
- [The original announcement from Yahoo!](#)

Getting Started

1. Download the [latest release of YCSB](#):

```
curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.17.0/ycsb-0.17.0.tar.gz  
tar xfvz ycsb-0.17.0.tar.gz  
cd ycsb-0.17.0
```

Unzip the folder using **tar** command on the Ubuntu terminal and write the following commands in sequence. (Install curl command if OS showed the error).

```
$curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.17.0/ycsb-0.17.0.tar.gz
$tar xfvz ycsb-0.17.0.tar.gz
$cd ycsb-0.17.0
```

```
hduser@muhammad-VM:~$ curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.17.0/ycsb-0.17.0.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 675M 100 675M    0     0 2116k      0  0:05:26 0:05:26 --:--:-- 1398k
hduser@muhammad-VM:~$ tar xfvz ycsb-0.17.0.tar.gz
ycsb-0.17.0/NOTICE.txt
ycsb-0.17.0/LICENSE.txt
ycsb-0.17.0/bin/ycsb.sh
ycsb-0.17.0/bin/ycsb.bat
ycsb-0.17.0/bin/ycsb
ycsb-0.17.0/bin/bindings.properties
ycsb-0.17.0/workloads/
ycsb-0.17.0/workloads/workloada
ycsb-0.17.0/workloads/workloadb
ycsb-0.17.0/workloads/workloadc
```

Now we will use YCSB later to check the performance of SQL and NoSQL databases.

5) Open a new Ubuntu terminal and download MongoDB by using the following commands.

```
$sudo wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-ubuntu1604-3.2.10.tgz
$sudo tar zxvf mongodb-linux-x86_64-ubuntu1604-3.2.10.tgz
$sudo mv ./mongodb-linux-x86_64-ubuntu1604-3.2.10 /usr/local
```

```
hduser@muhammad-VM:~$ sudo curl -O https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-ubuntu1604-3.2.10.tgz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 80.2M 100 80.2M    0     0 749k      0  0:01:49 0:01:49 --:--:-- 337k
hduser@muhammad-VM:~$ ls
Desktop      Downloads    Pictures      Templates    ycsb-0.17.0.tar.gz
docker-curriculum  mongodb-linux-x86_64-ubuntu1604-3.2.10.tgz  Public       Videos
Documents    Music       snap         ycsb-0.17.0
hduser@muhammad-VM:~$ sudo tar zxvf mongodb-linux-x86_64-ubuntu1604-3.2.10.tgz
mongodb-linux-x86_64-ubuntu1604-3.2.10/README
mongodb-linux-x86_64-ubuntu1604-3.2.10/THIRD-PARTY-NOTICES
mongodb-linux-x86_64-ubuntu1604-3.2.10/MPL-2
mongodb-linux-x86_64-ubuntu1604-3.2.10/GNU-AGPL-3.0
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongodump
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongorestore
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoexport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoimport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongostat
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongotop
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/bsondump
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoexport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoimport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongostat
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongotop
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/bsondump
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoexport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoimport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongostat
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongotop
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/bsondump
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoexport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongoimport
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongostat
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/mongotop
mongodb-linux-x86_64-ubuntu1604-3.2.10/bin/bsondump
hduser@muhammad-VM:~$ sudo mv ./mongodb-linux-x86_64-ubuntu1604-3.2.10 /usr/local
hduser@muhammad-VM:~$ cd /usr/local
hduser@muhammad-VM:/usr/local$
```

```
$cd /usr/local
```

```
$sudo chown -R hduser:hadoopgroup mongodb-linux-x86_64-ubuntu1604-3.2.10
```

```
$sudo ln -s ./mongodb-linux-x86_64-ubuntu1604-3.2.10 ./mongodb
```

```

hduser@muhammad-VM: /usr/local$ sudo chown -R hduser:hadoopgroup mongodb-linux-x86_64-ubuntu1604-3.2.10
hduser@muhammad-VM: /usr/local$ sudo ln -s mongodb-linux-x86_64-ubuntu1604-3.2.10 mongodb
hduser@muhammad-VM: /usr/local$ ls -l
lrwxrwxrwx 1 root root 38 Apr 13 00:37 mongodb -> mongodb-linux-x86_64-ubuntu1604-3.2.10
drwxr-xr-x 3 hduser hadoopgroup 4096 Apr 13 00:35 mongodb-linux-x86_64-ubuntu1604-3.2.10

```

Change the privileges for mongodb folder to hduser

```
$sudo chown -R hduser:hadoopgroup ./mongodb*
```

```
hduser@muhammad-VM: /usr/local$ sudo chown -R hduser:hadoopgroup mongodb*
```

```
$cd ..
```

```
$sudo mkdir mongodbddata
```

```
$sudo chown -R hduser:hadoopgroup mongodbddata
```

```
$cd /
```

```
$sudo ln -s /usr/mongodbddata data
```

```
$cd data
```

```
$mkdir -p db
```

```

hduser@muhammad-VM: /usr/local$ cd ..
hduser@muhammad-VM: /usr$ sudo mkdir mongodbddata
hduser@muhammad-VM: /usr$ sudo chown -R hduser:hadoopgroup mongodbddata
hduser@muhammad-VM: /usr$ ls -l
total 136
drwxr-xr-x  2 root root  49152 Apr 12 23:19 bin
drwxr-xr-x  2 root root   4096 Aug 19  2021 games
drwxr-xr-x 41 root root 16384 Apr  2 21:56 include
drwxr-xr-x 124 root root  4096 Feb 26 01:00 lib
drwxr-xr-x  2 root root  4096 Aug 19  2021 lib32
drwxr-xr-x  2 root root  4096 Mar  1 21:17 lib64
drwxr-xr-x 12 root root  4096 Jan 27 21:50 libexec
drwxr-xr-x  2 root root  4096 Aug 19  2021 libx32
drwxr-xr-x 18 root root  4096 Apr 13 00:37 local
drwxr-xr-x  2 hduser hadoopgroup 4096 Apr 13 00:37 mongodbddata
drwxr-xr-x  2 root root 20480 Apr 12 23:19 sbin
drwxr-xr-x 264 root root 12288 Apr  3 14:06 share
drwxr-xr-x  6 root root  4096 Apr  3 14:01 src
hduser@muhammad-VM: /usr$ cd /
hduser@muhammad-VM: /$ sudo ln -s /usr/mongodbddata data
hduser@muhammad-VM: /$ ls
bin  cdrom  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
boot  data  etc  lib  lib64  lost+found  mnt  proc  run  snap  swapfile  tmp  var
hduser@muhammad-VM: /$ cd data
hduser@muhammad-VM: /data$ mkdir -p db
hduser@muhammad-VM: /data$ cd
hduser@muhammad-VM: ~$

```

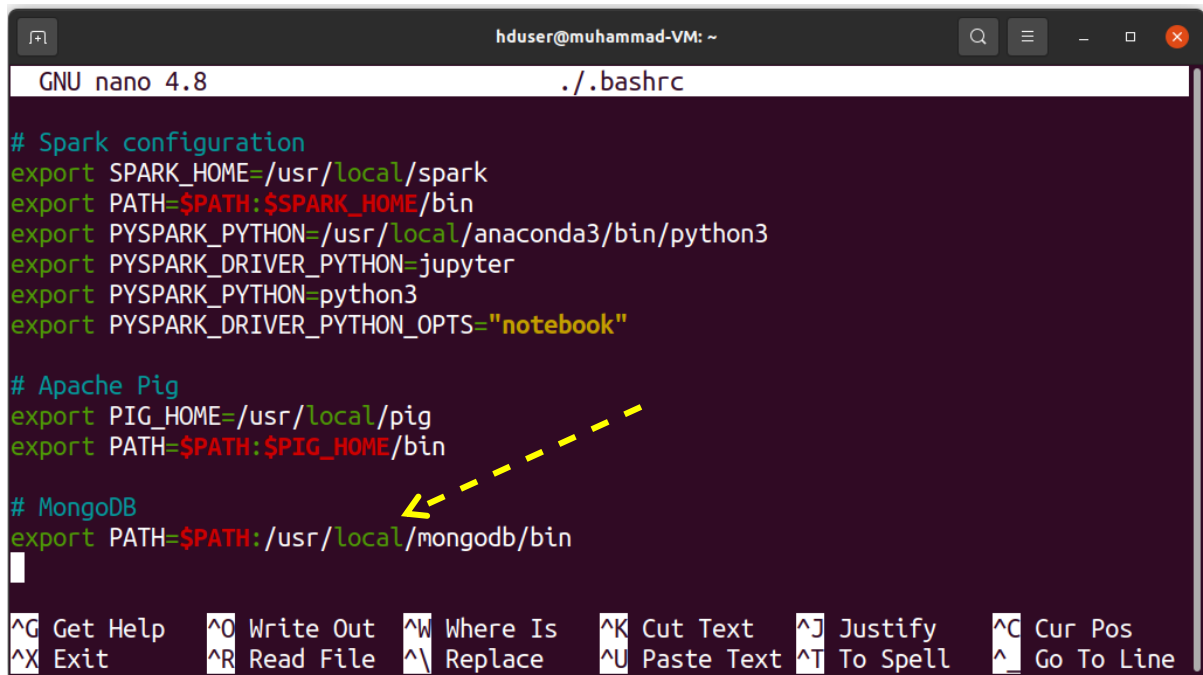
6) Edit the **./bashrc** file to update environment variables

```

hduser@muhammad-VM: ~$ pwd
/home/hduser
hduser@muhammad-VM: ~$ cd /home/hduser
hduser@muhammad-VM: ~$ nano ./bashrc

```

Update this file and add the line for the path of mongodb at the end of the file as mentioned below



```

GNU nano 4.8                               ~/.bashrc

# Spark configuration
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
export PYSARK_PYTHON=/usr/local/anaconda3/bin/python3
export PYSARK_DRIVER_PYTHON=jupyter
export PYSARK_PYTHON=python3
export PYSARK_DRIVER_PYTHON_OPTS="notebook"

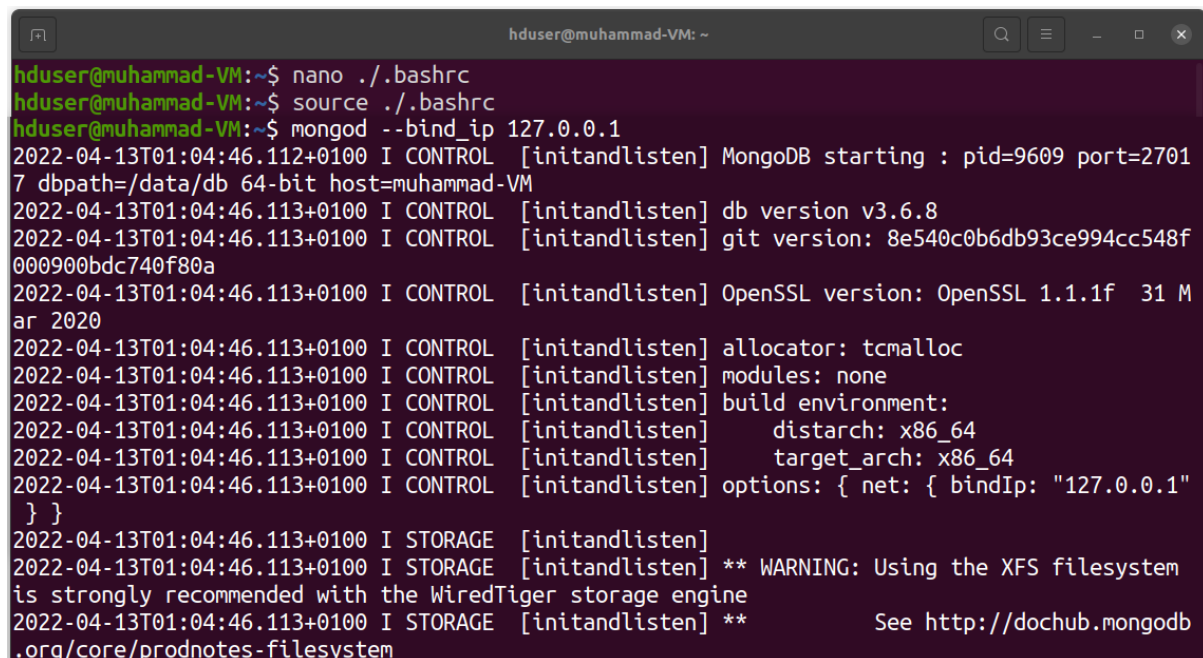
# Apache Pig
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin

# MongoDB
export PATH=$PATH:/usr/local/mongodb/bin

```

Save this file using nano editor (Ctrl + x, type y and hit the Enter key). If you want to allow the OS to see the updates, then use the following command.

```
$source ~/.bashrc
```



```

hduser@muhammad-VM:~$ nano ~/.bashrc
hduser@muhammad-VM:~$ source ~/.bashrc
hduser@muhammad-VM:~$ mongod --bind_ip 127.0.0.1
2022-04-13T01:04:46.112+0100 I CONTROL [initandlisten] MongoDB starting : pid=9609 port=2701
7 dbpath=/data/db 64-bit host=muhammad-VM
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] db version v3.6.8
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] git version: 8e540c0b6db93ce994cc548f
000900bdc740f80a
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.1.1f 31 M
ar 2020
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] allocator: tcmalloc
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] modules: none
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] build environment:
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten]     distarch: x86_64
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten]     target_arch: x86_64
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] options: { net: { bindIp: "127.0.0.1"
} }
2022-04-13T01:04:46.113+0100 I STORAGE [initandlisten]
2022-04-13T01:04:46.113+0100 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem
is strongly recommended with the WiredTiger storage engine
2022-04-13T01:04:46.113+0100 I STORAGE [initandlisten] ** See http://dochub.mongodb
.org/core/prodnotes-filesystem

```

Allow this session to run the **mongod** process. The mongoDB database is required in this mode on the terminal to execute the commands. Open a new terminal for the remaining steps of this tutorial. In case of any errors follow the step No. 7.

7) If you observe some kind error after the execution of `$mongod --bind_ip 127.0.0.1` command, then the following two commands should be executed in Ubuntu 22.04.* versions. **Otherwise you can ignore the step no. 7.**

(i)

```
$wget http://snapshot.debian.org/archive/debian/20190501T215844Z/pool/main/g/glibc/multiarch-support_2.28-10_amd64.deb
```


Type the above command and check the screenshot for further understanding. Then execute the below command

```
$sudo dpkg -i multiarch-support*.deb
```

(ii)

```
$wget http://snapshot.debian.org/archive/debian/20170705T160707Z/pool/main/o/openssl/libssl1.0.0_1.0.2l-1%7Ebpo8%2B1_amd64.deb
```

Then execute the below command

```
$sudo dpkg -i libssl1.0.0*.deb
```

You can check the execution of the commands in the screenshot as mentioned below

```
hduser@muhammad-Vm: ~
hduser@muhammad-Vm:~$ wget http://snapshot.debian.org/archive/debian/20190501T215844Z/pool/main/g/glibc/multiarch-support_2.28-10_amd64.deb
--2023-02-11 00:38:17-- http://snapshot.debian.org/archive/debian/20190501T215844Z/pool/main/g/glibc/multiarch-support_2.28-10_amd64.deb
Resolving snapshot.debian.org (snapshot.debian.org)... 193.62.202.27, 185.17.185.185, 2001:630:206:4000:1a1a:0:c13e:ca1b, ...
Connecting to snapshot.debian.org (snapshot.debian.org)|193.62.202.27|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 215080 (210K)
Saving to: 'multiarch-support_2.28-10_amd64.deb'

multiarch-support_2.28 100%[=====] 210.04K 47.8KB/s in 4.4s

2023-02-11 00:38:49 (47.8 KB/s) - 'multiarch-support_2.28-10_amd64.deb' saved [215080/215080]

hduser@muhammad-Vm:~$ sudo dpkg -i multiarch-support*.deb
Selecting previously unselected package multiarch-support.
(Reading database ... 242692 files and directories currently installed.)
Preparing to unpack multiarch-support_2.28-10_amd64.deb ...
Unpacking multiarch-support (2.28-10) ...
Setting up multiarch-support (2.28-10) ...

hduser@muhammad-Vm:~$ wget http://snapshot.debian.org/archive/debian/20170705T160707Z/pool/main/o/openssl/libssl1.0.0_1.0.2l-1%7Ebpo8%2B1_amd64.deb
--2023-02-11 00:39:37-- http://snapshot.debian.org/archive/debian/20170705T160707Z/pool/main/o/openssl/libssl1.0.0_1.0.2l-1%7Ebpo8%2B1_amd64.deb
Resolving snapshot.debian.org (snapshot.debian.org)... 185.17.185.185, 193.62.202.27, 2001:630:206:4000:1a1a:0:c13e:ca1b, ...
Connecting to snapshot.debian.org (snapshot.debian.org)|185.17.185.185|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1294986 (1.2M)
Saving to: 'libssl1.0.0_1.0.2l-1-bpo8+1_amd64.deb'

libssl1.0.0_1.0.2l-1-b 100%[=====] 1.23M 1.41MB/s in 0.9s

2023-02-11 00:39:39 (1.41 MB/s) - 'libssl1.0.0_1.0.2l-1-bpo8+1_amd64.deb' saved [1294986/1294986]

hduser@muhammad-Vm:~$ sudo dpkg -i libssl1.0.0*.deb
Selecting previously unselected package libssl1.0.0:amd64.
(Reading database ... 242696 files and directories currently installed.)
Preparing to unpack libssl1.0.0_1.0.2l-1-bpo8+1_amd64.deb ...
Unpacking libssl1.0.0:amd64 (1.0.2l-1-bpo8+1) ...
Setting up libssl1.0.0:amd64 (1.0.2l-1-bpo8+1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
```

Restart your VM after the installation of the above commands. Execute the command as mentioned below

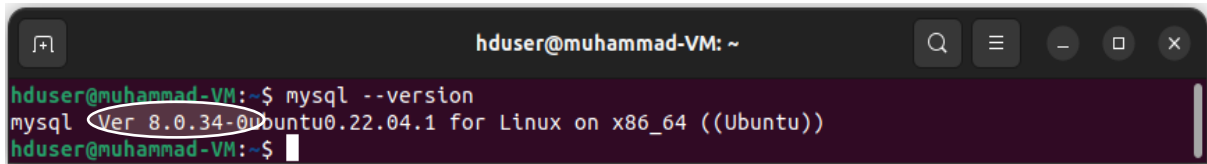
```
$mongod --bind_ip 127.0.0.1
```

```
hduser@muhammad-Vm:~$ mongod --bind_ip 127.0.0.1
2022-04-13T01:04:46.112+0100 I CONTROL [initandlisten] MongoDB starting : pid=9609 port=2701
7 dbpath=/data/db 64-bit host=muhammad-Vm
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] db version v3.6.8
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] git version: 8e540c0b6db93ce994cc548f
000900bdc740f80a
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.1.1f 31 Mar 2020
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] allocator: tcmalloc
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] modules: none
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] build environment:
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] distarch: x86_64
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] target_arch: x86_64
2022-04-13T01:04:46.113+0100 I CONTROL [initandlisten] options: { net: { bindIp: "127.0.0.1"
} }
2022-04-13T01:04:46.113+0100 I STORAGE [initandlisten]
2022-04-13T01:04:46.113+0100 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem
is strongly recommended with the WiredTiger storage engine
2022-04-13T01:04:46.113+0100 I STORAGE [initandlisten] ** See http://dochub.mongodb
.org/core/prodnotes-filesystem
```

Allow this session to run the **mongod** process. The mongoDB database is required in this mode on the terminal to execute the commands.

8) Use YCSB with MySQL and mongoDB

(i) Open up a new terminal session and check the version of MySQL as we did in step 1.



```
hduser@muhammad-VM: ~$ mysql --version
mysql Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
hduser@muhammad-VM: ~$
```

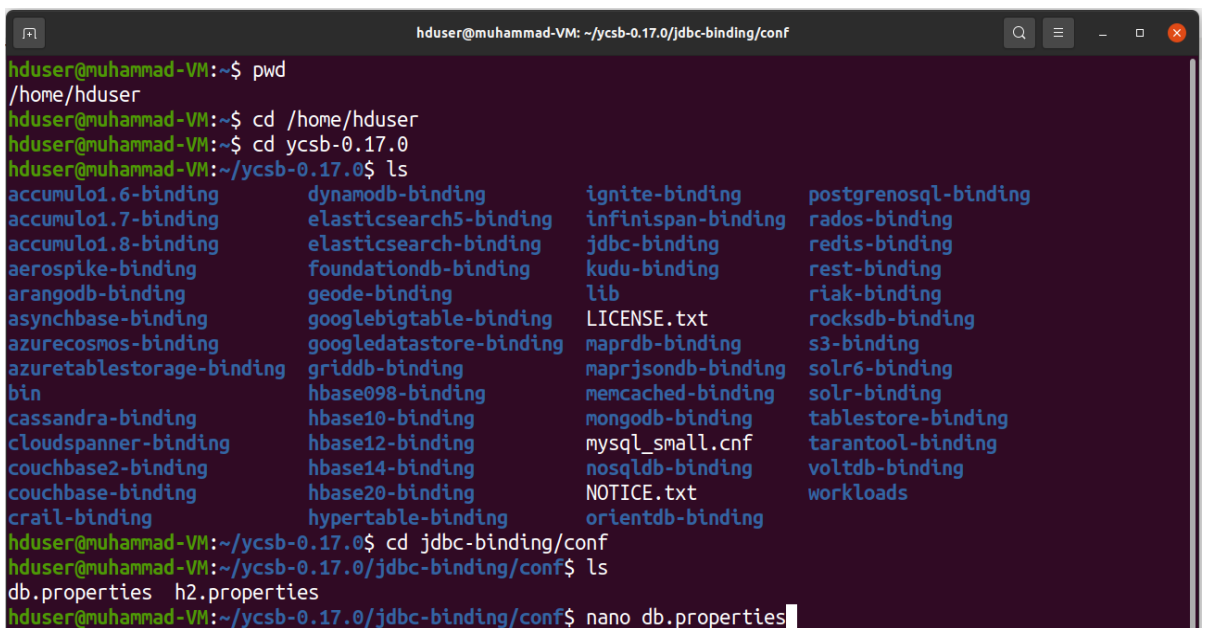
We are using 8.0.34 MySQL and if you are using some other version, then download the relevant connector from MySQL resources (<https://dev.mysql.com/downloads/connector/j/?os=26>).

(ii) Download mysql connector from Moodle. Copy the **mysql-connector-java-8.0.33.jar** file into the **/home/hduser/ycsb-0.17.0/jdbc-binding/lib** environment and shared folder (Download connector file from Moodle) as mentioned below in the screenshot.



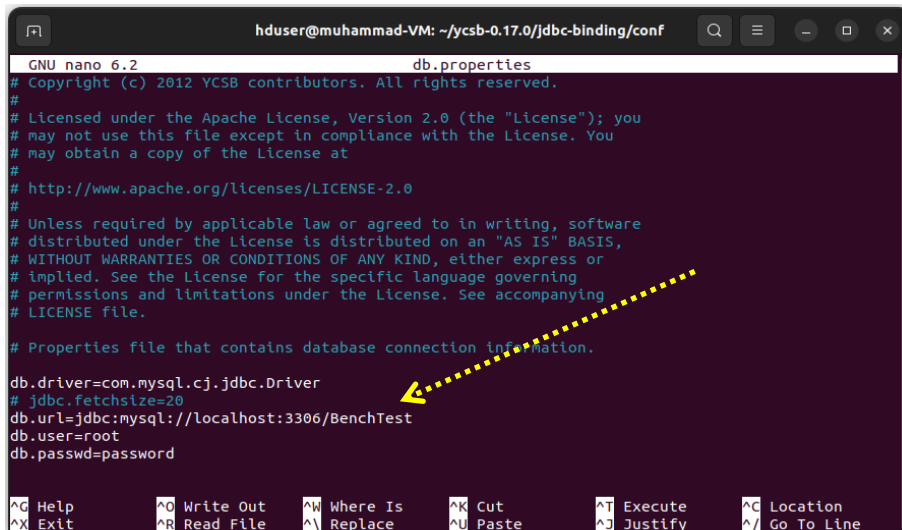
```
hduser@muhammad-VM: ~$ cd /home/hduser/Downloads
hduser@muhammad-VM: ~/Downloads$ ls mysql-connector*
mysql-connector-j-8.0.33.jar
hduser@muhammad-VM: ~/Downloads$ sudo cp ./mysql-connector-j-8.0.33.jar /home/hduser/ycsb-0.17.0/jdbc-binding/lib
[sudo] password for hduser:
hduser@muhammad-VM: ~/Downloads$ cd
hduser@muhammad-VM: ~$ pwd
/home/hduser
hduser@muhammad-VM: ~$
```

9) Edit the file named as **db.properties** in the **ycsb-0.17.0/jdbc-binding/conf** directory as follows:



```
hduser@muhammad-VM: ~$ pwd
/home/hduser
hduser@muhammad-VM: ~$ cd /home/hduser
hduser@muhammad-VM: ~$ cd ycsb-0.17.0
hduser@muhammad-VM: ~/ycsb-0.17.0$ ls
accumulo1.6-binding  dynamodb-binding  ignite-binding      postgresql-binding
accumulo1.7-binding  elasticsearch5-binding  infinispn-binding  rados-binding
accumulo1.8-binding  elasticsearch-binding  jdbc-binding        redis-binding
aerospike-binding    foundationdb-binding  kudu-binding        rest-binding
arangodb-binding      geode-binding         lib                 riak-binding
asynchbase-binding   googlebigtable-binding  LICENSE.txt          rocksdb-binding
azurecosmos-binding  googledatastore-binding  maprdb-binding      s3-binding
azuretablestorage-binding  griddb-binding         maprjsondb-binding  solr6-binding
bin                  hbase098-binding       memcached-binding   solr-binding
cassandra-binding     hbase10-binding        mongodb-binding     tablestore-binding
cloudspanner-binding  hbase12-binding        mysql_small.cnf     tarantool-binding
couchbase2-binding    hbase14-binding        nosqlldb-binding    vltodb-binding
couchbase-binding     hbase20-binding        NOTICE.txt          workloads
craill-binding         hypertable-binding      orientdb-binding
hduser@muhammad-VM: ~/ycsb-0.17.0$ cd jdbc-binding/conf
hduser@muhammad-VM: ~/ycsb-0.17.0/jdbc-binding/conf$ ls
db.properties  h2.properties
hduser@muhammad-VM: ~/ycsb-0.17.0/jdbc-binding/conf$ nano db.properties
```

Update the contents of the file as mentioned below



```

GNU nano 6.2 db.properties
# Copyright (c) 2012 YCSB contributors. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"); you
# may not use this file except in compliance with the License. You
# may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied. See the License for the specific language governing
# permissions and limitations under the License. See accompanying
# LICENSE file.
#
# Properties file that contains database connection information.

db.driver=com.mysql.cj.jdbc.Driver
# jdbc.fetchsize=20
db.url=jdbc:mysql://localhost:3306/BenchTest
db.user=root
db.passwd=password
  
```

Save this file using **nano** editor as you did in the previous tutorials and use the following command to move to ycsb-0.0.17.0 folder and moved to the folder ycsb-0.17.0 by using `$cd ../../`

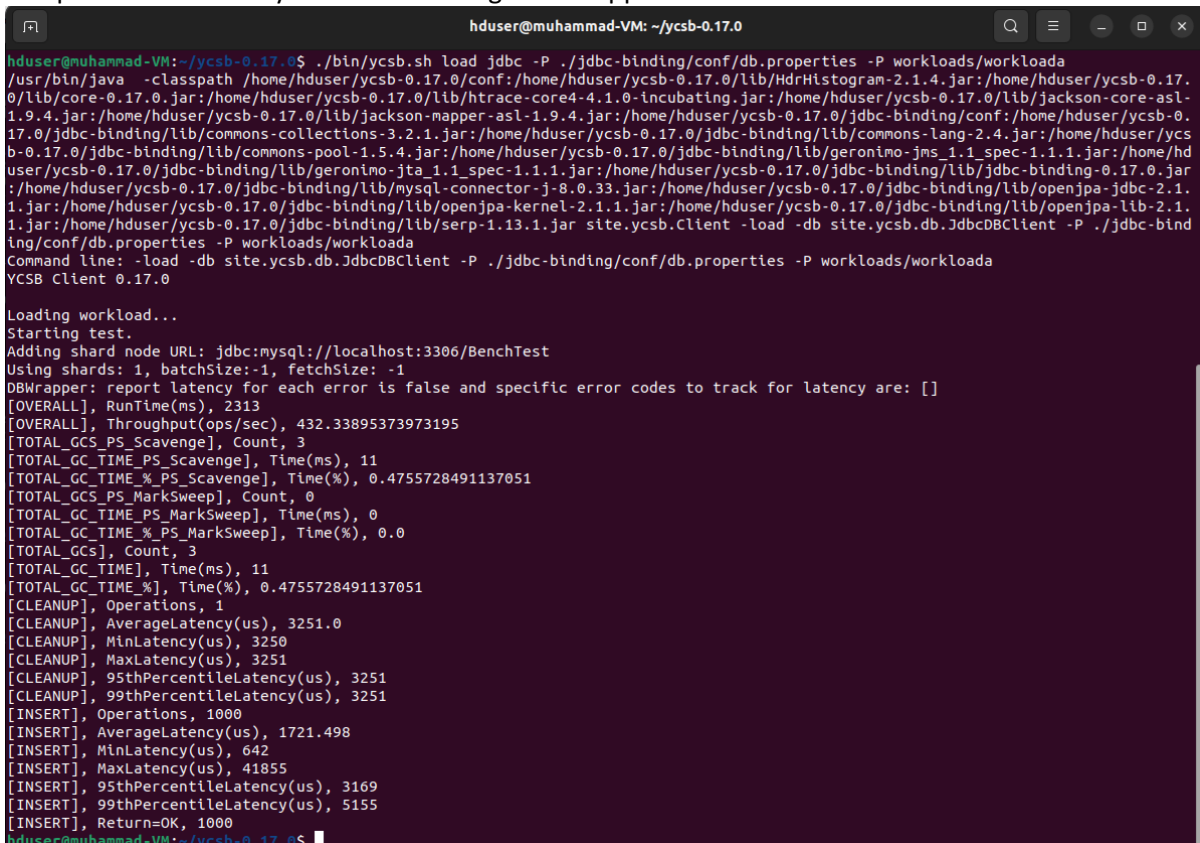
```

hduser@muhammad-VM:~/ycsb-0.17.0/jdbc-binding/conf$ cd ../../
hduser@muhammad-VM:~/ycsb-0.17.0$
  
```

10) The environment is set for **mongodb** and **mysql** and now we execute the workload with **ycsb-0.17.0** by using the following commands as

Execute a sample workload against mySQL as mentioned below

`hduser@muhammad-vm:~/ycsb-0.17.0$./bin/ycsb.sh load jdbc -P ./jdbc-binding/conf/db.properties -P workloads/workloada` and press the Enter Key and the following screen appears and it takes a little while



```

hduser@muhammad-VM:~/ycsb-0.17.0$ ./bin/ycsb.sh load jdbc -P ./jdbc-binding/conf/db.properties -P workloads/workloada
/usr/bin/java -classpath /home/hduser/ycsb-0.17.0/conf:/home/hduser/ycsb-0.17.0/lib/HdrHistogram-2.1.4.jar:/home/hduser/ycsb-0.17.0/lib/core-0.17.0.jar:/home/hduser/ycsb-0.17.0/lib/htrace-core4-4.1.0-incubating.jar:/home/hduser/ycsb-0.17.0/lib/jackson-core-asl-1.9.4.jar:/home/hduser/ycsb-0.17.0/lib/jackson-mapper-asl-1.9.4.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/conf:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/commons-collections-3.2.1.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/commons-lang-2.4.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/commons-pool-1.5.4.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/geronimo-jms_1.1_spec-1.1.1.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/geronimo-jta_1.1_spec-1.1.1.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/mysql-connector-j-8.0.33.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/openjpa-jdbc-2.1.1.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/openjpa-kernel-2.1.1.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/openjpa-lib-2.1.1.jar:/home/hduser/ycsb-0.17.0/jdbc-binding/lib/serp-1.13.1.jar site.ycsb.Client -load -db site.ycsb.db.JdbcDBClient -P ./jdbc-binding/conf/db.properties -P workloads/workloada
Command line: -load -db site.ycsb.db.JdbcDBClient -P ./jdbc-binding/conf/db.properties -P workloads/workloada
YCSB Client 0.17.0

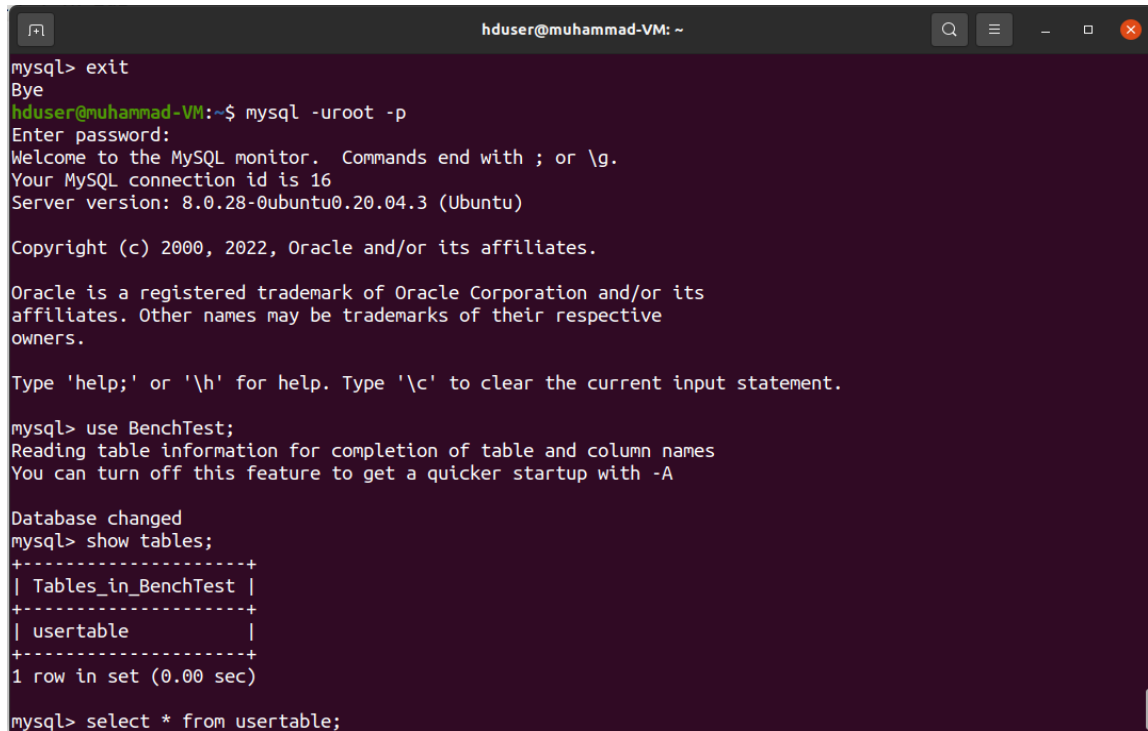
Loading workload...
Starting test.
Adding shard node URL: jdbc:mysql://localhost:3306/BenchTest
Using shards: 1, batchSize:-1, fetchSize:-1
DBWrapper: report latency for each error is false and specific error codes to track for latency are: []
[OVERALL], RunTime(ms), 2313
[OVERALL], Throughput(ops/sec), 432.33895373973195
[TOTAL_GCS_PS_Scavenge], Count, 3
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 11
[TOTAL_GC_TIME_PS_Scavenge], Time(%), 0.4755728491137051
[TOTAL_GCS_PS_MarkSweep], Count, 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(%), 0.0
[TOTAL_GCs], Count, 3
[TOTAL_GC_TIME], Time(ms), 11
[TOTAL_GC_TIME_%], Time(%), 0.4755728491137051
[CLEANUP], Operations, 1
[CLEANUP], AverageLatency(us), 3251.0
[CLEANUP], MinLatency(us), 3250
[CLEANUP], MaxLatency(us), 3251
[CLEANUP], 95thPercentileLatency(us), 3251
[CLEANUP], 99thPercentileLatency(us), 3251
[INSERT], Operations, 1000
[INSERT], AverageLatency(us), 1721.498
[INSERT], MinLatency(us), 642
[INSERT], MaxLatency(us), 41855
[INSERT], 95thPercentileLatency(us), 3169
[INSERT], 99thPercentileLatency(us), 5155
[INSERT], Return=OK, 1000
hduser@muhammad-VM:~/ycsb-0.17.0$
  
```

Create an excel sheet and record the parameters (AverageLatency, MinLatency, ..) as you obtained after the execution of the above command for workloada for mySQL. Execute the above command for

different workloads as mentioned in the lecture notes and ycsb web resource (Check for conceptual understanding). This command will store 1000 records in the usertable in MySQL database named as "BenchTest". You can check by using `mysql>select * from usertable;`

You can start a new session of MySQL on the new terminal. Perform the query in the **BenchTest** database's **usertable** to see if records have been successfully inserted as shown below in the screen-shots.

Note: Password for mysql is "**password**" as we set in the Tutorial 5 (part I).



```

mysql> exit
Bye
hduser@muhammad-VM:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.28-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

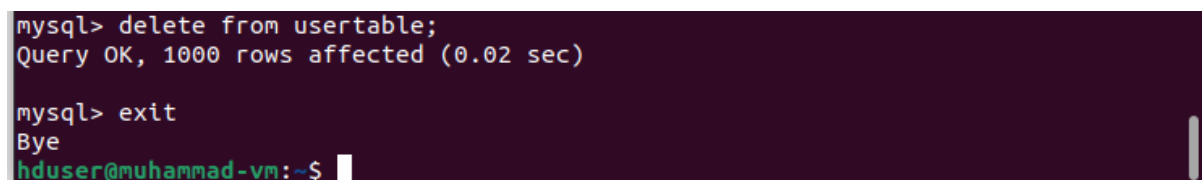
mysql> use BenchTest;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_BenchTest |
+-----+
| usertable            |
+-----+
1 row in set (0.00 sec)

mysql> select * from usertable;

```

Note: If you would like to execute the above command again, remove the records stored in MySQL table (usertable) by using the command. `mysql>delete from usertable;`



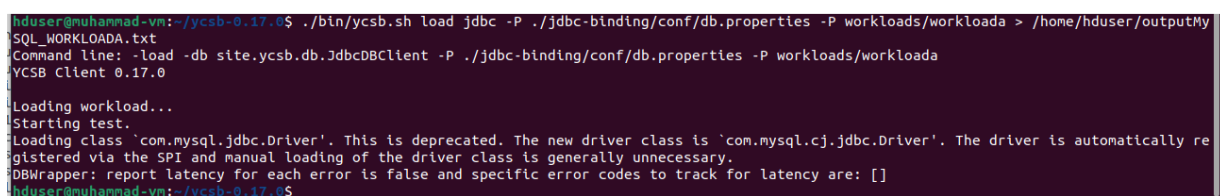
```

mysql> delete from usertable;
Query OK, 1000 rows affected (0.02 sec)

mysql> exit
Bye
hduser@muhammad-vm:~$

```

If you would like to store the YCSB generated output in the text file, then use the following command. (Make sure the records should be deleted from MySQL table before the execution of below mentioned command)



```

hduser@muhammad-vm:~/ycsb-0.17.0$ ./bin/ycsb.sh load jdbc -P ./jdbc-binding/conf/db.properties -P workloads/workloada > /home/hduser/outputMySQL_WORKLOADA.txt
Command line: -load -db site.ycsb.db.JdbcDBClient -P ./jdbc-binding/conf/db.properties -P workloads/workloada
YCSB Client 0.17.0

Loading workload...
Starting test.
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
DBWrapper: report latency for each error is false and specific error codes to track for latency are: []
hduser@muhammad-vm:~/ycsb-0.17.0$

```

Check the folder `/home/hduser/` in ubuntu and you will find the file named as "outputMySQL_WORKLOADA.txt"

11) For mongoDB, you can use the following command to execute a sample workload against mongoDB.

```
hduser@muhammad-vm:~/ycsb-0.17.0$ ./bin/ycsb.sh load mongodb -s -P workloads/workloada
```

The entire output after the execution of above command in this step is mentioned below.

```
hduser@muhammad-VM: ~/ycsb-0.17.0
hduser@muhammad-VM:~/ycsb-0.17.0$ ./bin/ycsb.sh load mongodb -s -P workloads/workloada
/usr/bin/java -classpath /home/hduser/ycsb-0.17.0/conf:/home/hduser/ycsb-0.17.0/lib/HdrHistogram-2.1.4.jar:/home/hduser/ycsb-0.17.0/lib/core-0.17.0.jar:/home/hduser/ycsb-0.17.0/lib/htrace-core4-4.1.0-incubating.jar:/home/hduser/ycsb-0.17.0/lib/jackson-core-asl-1.9.4.jar:/home/hduser/ycsb-0.17.0/lib/jackson-mapper-asl-1.9.4.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/logback-classic-1.1.2.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/logback-core-1.1.2.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/mongo-java-driver-3.8.0.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/mongodb-async-driver-2.0.1.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/mongodb-binding-0.17.0.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/slf4j-api-1.7.25.jar:/home/hduser/ycsb-0.17.0/mongodb-binding/lib/snappy-java-1.1.7.1.jar site.ycsb.Client -load -db site.ycsb.db.MongoDbClient -s -P workloads/workloada
Command line: -load -db site.ycsb.db.MongoDbClient -s -P workloads/workloada
YCSB Client 0.17.0

Loading workload...
Starting test.
2024-02-01 13:32:54:910 0 sec: 0 operations; est completion in 0 second
mongo client connection created with mongodb://localhost:27017/ycsb?w=1
DBWrapper: report latency for each error is false and specific error codes to track for latency are: []
2024-02-01 13:32:55:608 0 sec: 1000 operations; 1328.02 current ops/sec; [CLEANUP: Count=1, Max=1685, Min=1685, Avg=1685, 90=1685, 99=1685, 99.9=1685, 99.99=1685] [INSERT: Count=1000, Max=47327, Min=117, Avg=254.19, 90=276, 99=981, 99.9=3481, 99.99=47327]
[OVERALL], Runtime(ms), 753
[OVERALL], Throughput(ops/sec), 1328.0212483399735
[TOTAL_GC_PS_Scavenge], Count, 2
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 8
[TOTAL_GC_TIME_%_PS_Scavenge], Time(%), 1.0624169986719787
[TOTAL_GC_PS_MarkSweep], Count, 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 0
[TOTAL_GC_TIME_%_PS_MarkSweep], Time(%), 0.0
[TOTAL_GCs], Count, 2
[TOTAL_GC_TIME], Time(ms), 8
[TOTAL_GC_TIME_%], Time(%), 1.0624169986719787
[CLEANUP], Operations, 1
[CLEANUP], AverageLatency(us), 1685.0
[CLEANUP], MinLatency(us), 1685
[CLEANUP], MaxLatency(us), 1685
[CLEANUP], 95thPercentileLatency(us), 1685
[CLEANUP], 99thPercentileLatency(us), 1685
[INSERT], Operations, 1000
[INSERT], AverageLatency(us), 254.187
[INSERT], MinLatency(us), 117
[INSERT], MaxLatency(us), 47327
[INSERT], 95thPercentileLatency(us), 361
[INSERT], 99thPercentileLatency(us), 981
[INSERT], Return=OK, 1000
hduser@muhammad-VM:~/ycsb-0.17.0$
```

You can repeat the exercise for other workloads also and record the data in the excel sheet for comparison as mentioned for mySQL.

You can check a database called "ycsb" and a collection named "usertable" in the MongoDB database to see if the MongoDB workloada was successfully executed or not. Utilizing the MongoDB client, you can examine them.

```
hduser@muhammad-vm: ~
hduser@muhammad-vm:~$ mongo
MongoDB shell version: 3.2.10
connecting to: test
> show dbs
local 0.000GB
ycsb 0.001GB
> use ycsb
switched to db ycsb
> show collections
usertable
> db.usertable.find()
{ "_id" : "user6284781860667377211", "field1" : BinData(0,"Kj12KyM2My9kL01JfcrNUU/LkFj0z4yJF5vJLwx0j14LiQ2K0J5Lc9JkV/LkgLLCMuJT
kqNEgxJDA6PUQvIDNyKjN4M0wpLURtIiHoKF9lPzVsPlcLMSZgMDMyLDcy01ApIQ=="), "field0" : BinData(0,"Ikh1N15jKVQlMCQyLFh3P0JpJDRoMih0ILmzI
Vo/PEfxJLYxNFvlw2LFR5I0vrIF19014tISUqMcokPEUXJER30C4gMCI0MTd8L1BjJDD0P1Z1JVg3JEZ3IEghIDUoNSFoJg=="), "field7" : BinData(0,"LjVq
MiYuJDYvPsmLLU9MES1PEA9NTtkMVExLEMjLTQsJEX9IzR8JkjhLERlKk83NCR+OydiPjg4MEJhJcmFcl0UhtKitlL0ptPiv+KEs90SxQODV2M09tM0prJiJ80TskI
A=="), "field6" : BinData(0,"Myto01gvLz1oL0NpJVglLE9hLUSHp0Z/JkttJycw0FgvKSggJ5S+Lk9vLugnLTJsP0FzLkh5KDI40VJnK0onJFpnL0M/NEojJkV5
JyV8Jig4NdWkPCggNDZ0JyBqNiMqMh90w=="), "field9" : BinData(0,"KVE1MSA6N1ptL1hIDowKUKLJEL/PktLPipsL0ttMy8s0EKhLFFxMzpwPFE/KSNwn10
lLewjNyc4NUn0J187KE4jLjxsnLNVN1RxlKgtPyYuMS9u0TMMID540i58IKox0EMnIA=="), "field8" : BinData(0,"KS42MEk70iFmKlwjNDpwJzZwPy52Jj1mKC
BmTL101uvc0DT4mM30a0T70Nv40tCEv7E550zomYzcw7E1hM1E10E85MIE2DTv0TDR07vY0NvU17DvCv2M316TAx/0C8cTDCi7lctDA==") "field3" : BinData(0,"M
```

In order to execute the workload command again, remove the mongoDB database before the execution of the workload command for second workload as shown in the screenshot.

The image shows two terminal windows. The left window displays MongoDB logs for a WiredTiger engine, including messages about detecting an unclean shutdown, recovering data, and initializing full-time diagnostic data capture. The right window shows a MongoDB shell session where the user connects to a test database, shows the database size (0.000GB), switches to the 'ycsb' database, and drops it, resulting in a 'dropped' message.

```

e engine to 'WiredTiger'.
2023-04-23T21:45:04.004+0100 W - [initandlisten] Detected unclean shutdown
n - /data/db/mongod.lock is not empty.
2023-04-23T21:45:04.004+0100 W STORAGE [initandlisten] Recovering data from the
last clean checkpoint.
2023-04-23T21:45:04.004+0100 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=false,
statistics=(Fast),log=(enabled=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2023-04-23T21:45:04.209+0100 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory '/data/db/diagnostic.data'
2023-04-23T21:45:04.210+0100 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2023-04-23T21:45:04.210+0100 I NETWORK [initandlisten] waiting for connections
on port 27017
2023-04-23T21:45:05.011+0100 I FTDC [ftdc] Unclean full-time diagnostic data
capture shutdown detected, found interim file, some metrics may have been lost.
OK
2023-04-23T21:45:15.154+0100 I NETWORK [initandlisten] connection accepted from
127.0.0.1:56858 #1 (1 connection now open)
2023-04-23T21:46:44.526+0100 I COMMAND [conn1] dropDatabase ycsb starting
2023-04-23T21:46:44.534+0100 I COMMAND [conn1] dropDatabase ycsb finished

```

```

hduser@muhammad-vm: ~$ mongo
MongoDB shell version: 3.2.10
connecting to: test
> show dbs
local 0.000GB
ycsb 0.001GB
> use ycsb
switched to db ycsb
> db.dropDatabase()
{ "dropped" : "ycsb", "ok" : 1 }
> show dbs
local 0.000GB
>

```

You can execute for second workload after the deletion of the database in MongoDB or you can change the name of database for different workloads. Please explore yourself.

12) If you would like to store output generated by YCSB in a file, then use the following commands for the database (MongoDB) as mentioned below

```

$cd /home/hduser/ycsb-0.17.0
$./bin/ycsb load mongodb -s -P workloads/workloada > /home/hduser/output-Load.txt

```

13) Further exploration of workloads files is available in the below-mentioned folders. Different workloads are present in the folder

```

$cd ycsb-0.17.0/workloads

```

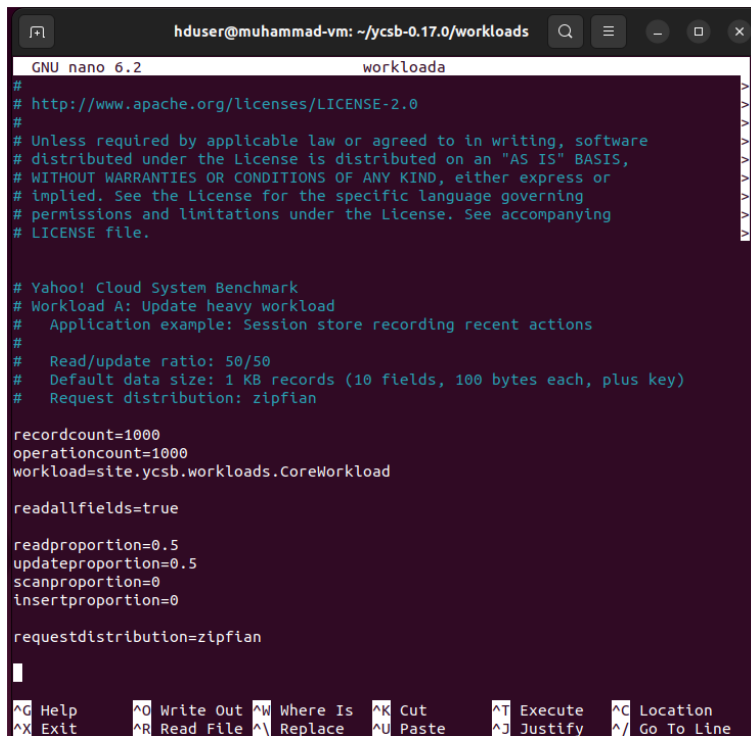
The image shows a terminal window where the user has navigated to the /ycsb-0.17.0/workloads directory. The 'ls' command lists various database bindings (e.g., accumulo, dynamodb, elasticsearch, mongodb, mysql, postgresql, redis, etc.) and workload files (e.g., workloada, workloadb, workloadc, workloadd, workloadf, workload_template). The user then enters 'cd workloads/' and 'ls' again, showing the same list of files and directories.

```

hduser@muhammad-VM: ~/ycsb-0.17.0/workloads
hduser@muhammad-VM:~$ cd ycsb-0.17.0/
hduser@muhammad-VM:~/ycsb-0.17.0$ ls
accumulo1.6-binding      dynamodb-binding      ignite-binding         postgresql-binding
accumulo1.7-binding      elasticsearch5-binding infinispn-binding      rados-binding
accumulo1.8-binding      elasticsearch-binding  jdbc-binding           redis-binding
aerospike-binding        foundationdb-binding   kudu-binding           rest-binding
arangodb-binding          geode-binding          LICENSE.txt             riak-binding
asynchbase-binding        googlebigtable-binding memcached-binding      rocksdb-binding
azurecosmos-binding       googledatastore-binding maprdb-binding         s3-binding
azuredatabasestorage-binding griddb-binding         maprjsondb-binding    solr6-binding
bin                       hbase098-binding      mongodb-binding        solr-binding
cassandra-binding         hbase10-binding       mysql_small.cnf        tablestore-binding
cloudspanner-binding      hbase12-binding       nosqlldb-binding      tarantool-binding
couchbase2-binding        hbase14-binding       NOTICE.txt            voldemort-binding
couchbase-binding         hbase20-binding        orientdb-binding       workloads
crail-binding             hypertable-binding
hduser@muhammad-VM:~/ycsb-0.17.0$ cd workloads/
hduser@muhammad-VM:~/ycsb-0.17.0/workloads$ ls
tsworkloada             workloada              workloadc              workloadf
tsworkload_template     workloadb              workloadd              workload_template
hduser@muhammad-VM:~/ycsb-0.17.0/workloads$ nano workloada
hduser@muhammad-VM:~/ycsb-0.17.0/workloads$ nano workloadb
hduser@muhammad-VM:~/ycsb-0.17.0/workloads$ nano workloadf
hduser@muhammad-VM:~/ycsb-0.17.0/workloads$

```

You can apply different workloads for **mysql** and **mongodb** to perform a comparative analysis for read, write and update operations. You can modify the recordcount also in these files if you are interested to explore further.



```
GNU nano 6.2 workloada
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied. See the License for the specific language governing
# permissions and limitations under the License. See accompanying
# LICENSE file.

# Yahoo! Cloud System Benchmark
# Workload A: Update heavy workload
# Application example: Session store recording recent actions
#
# Read/update ratio: 50/50
# Default data size: 1 KB records (10 fields, 100 bytes each, plus key)
# Request distribution: zipfian

recordcount=1000
operationcount=1000
workload=site.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=0.5
updateproportion=0.5
scanproportion=0
insertproportion=0

requestdistribution=zipfian
```

If you need more explanation for any above-mentioned command, please check the below mentioned references or ycsb/mongodb/SQL documentation.

References:

- <https://github.com/brianfrankcooper/YCSB/tree/master/mongodb>
- <https://github.com/brianfrankcooper/YCSB/tree/master/jdbc>
- <https://kiran4t4t.blogspot.com/2016/05/quick-start-guide-to-run-ycsb-on-mongodb.html>
- <https://learning.oreilly.com/library/view/mastering-mongodb-6-x/9781803243863/>