



RA Work Documentation

DATA REPLICATION & SYNTHETIC WORK LOAD GENERATION
M KARAM SHEHZAD & JAM MUHAMMAD YOUSIF
SUPERVISOR: DR. MUHAMMAD SAQIB ILYAS

NAMAL COLLEGE, MIANWALI, PAKISTAN | karam.shehzad@hotmail.com,
jammuhammadyusif@gmail.com, saqib.ilyas@namal.edu.pk

Table of Contents

1	Introduction	5
1.1	Bootable Steps	5
1.2	Installation of Ubuntu	8
1.2.1	Ubuntu Installation Steps.....	8
1.2.2	Welcome to Ubuntu!	11
2	What is MySQL Replication?	12
2.1	Replication Scheme	12
3	Setting up Replication Slaves	12
3.1	Replication-Test	13
4	PostgreSQL-Replication.....	14
5	Cassandra DB Replication on Ubuntu 14.04	14
5.1	Login to Cassandra using Username & Password	17
5.1.1	Example.....	17
6	Experiments	19
6.1	Experiments of PostgreSQL.....	19
6.2	MySQL	21
6.3	Cassandra	23
6.3.1	24
7	Automation	27
7.1	Automation of MySQL.....	27
7.2	Automation of PostgreSQL	28
8	Files and Folders.....	30
8.1	Faban.....	30
8.2	Replication	30
9	Installation of Software's	32
9.1	How to install SSH-Server.....	32
9.1.1	Optional Step -Disable the Password for Root Login	34
9.2	How to install Wire-Shark	35
9.3	How to install VirtualBox - Virt Manager	36
9.4	How to install VirtualBox	37
10	Solutions to Problems	38
10.1	How to Transfer File/Folder from Host P.C to Remote.....	38

10.1.1	Example.....	38
10.1.2	Example.....	38
10.2	How to Check IP & MAC address of a Machine	39
10.2.1	Example.....	40
10.3	Determining free memory (RAM) on Linux.....	40
10.4	Determining free memory (RAM) on Windows.....	40
10.5	Determining free memory (Hard Disk) on Windows	41
10.6	Determining free memory (Hard Disk) on Linux.....	42
10.7	Convert KVM File to VirtualBox	43
10.8	MySQL Replication Errors	43
10.8.1	Create dumpfile error	43
10.8.2	Replication not working	44
10.8.3	Replication error	44
10.8.4	Replication Error	45
10.8.5	If MySQL Log file is not at correct position	46
10.8.6	If nothing works or Slave IO running=No and SQL running=No.....	46
10.9	How to set compression approach in MySQL Master-Slave Replication.....	47
10.10	Stop Wire-Shark using Shell Script	48
10.11	Start Wire-Shark using Shell Script	48
10.12	Start/Stop MySQL Slave using Shell Script	48
10.13	Measure CPU Performance using Shell Script	49
10.14	Completely Remove MySQL.....	49
10.15	C-Code for MySQL to send data to remote host.....	50
10.16	Completely Remove PostgreSQL using Shell Script	51
10.17	How to Send Images to remote PostgreSQL server using C-Code.....	51
10.17.1	Example.....	54
10.18	Solution to Encrypted Data in Wire-shark	55
10.18.1	Solution	55
10.18.2	Decrypted data.....	55
10.18.3	Encrypted Data.....	55
10.19	Conversion of Image to Hex Values	56
10.20	How to Find difference between two csv files.....	56
10.21	How to Convert Hex File to Text Format	56

10.22	Merge two rows into one row	56
10.23	Use of Wire-Shark	57
10.24	Terminal Commands	57
10.25	MySQL Commands	57
10.26	PostgreSQL Commands	58
10.27	Resizing Virtual Machines	59
10.27.1	Assumptions	59
10.28	Assign Static IP on Linux	59
10.29	Installation of Dukto on Linux	61
10.30	Installation of MongoDB on Ubuntu 14.04	61
10.31	Compression Approach in Cassandra DB	61
11	Note	62

Table of Figures

Fig. 1. Basic Model	5
Fig. 2. How to use Rufus Step 1.....	6
Fig. 3. How to use Rufus Step 2.....	7
Fig. 4. How to use Rufus Step 3.....	7
Fig. 5. How to install Ubuntu Step 1	9
Fig. 6. How to install Ubuntu Step 2	10
Fig. 7. How to install Ubuntu Step 3	10
Fig. 8. MySQL Replication Test.....	13
Fig. 9. Python Code for Tweets Data.....	20
Fig. 10. Command to run Python Code.....	21
Fig. 11. Python Code for Flickr Images.....	22
Fig. 12. Inputs for MySQL Experiment	28
Fig. 13. Inputs for PostgreSQL Experiment	29
Fig. 14. Steps to Perform SSH Login without using Password.....	33
Fig. 15. Output of SSH copy ID	34
Fig. 16. Remote Login using SSH	34
Fig. 17. How to start Wire-shark	36
Fig. 18. Output window of Wire-shark.....	36
Fig. 19. Copy data to Remote Machine.....	38
Fig. 20. Copy data to Remote Machine 2.....	39
Fig. 21. Output of ifconfig	39
Fig. 22. Output of free Command	40
Fig. 23. RAM on Windows OS.....	41
Fig. 24. Hard Disk on Windows	42
Fig. 25. Hard Disk on Ubuntu	43
Fig. 26. KVM to VBox.....	43
Fig. 27. Solution to IO Connecting.....	45
Fig. 28. Solution to SQL Running No	46
Fig. 29. How to run C Code	54
Fig. 30. Static IP Step 1	59
Fig. 31. Static IP Step 2	60
Fig. 32. Static IP Step 3	60
Fig. 33. Dukto Installation Commands	61

1 Introduction

There are five machines namely Client, *Master* and 3-*Slaves*. Each of them is running on different physical hosts.

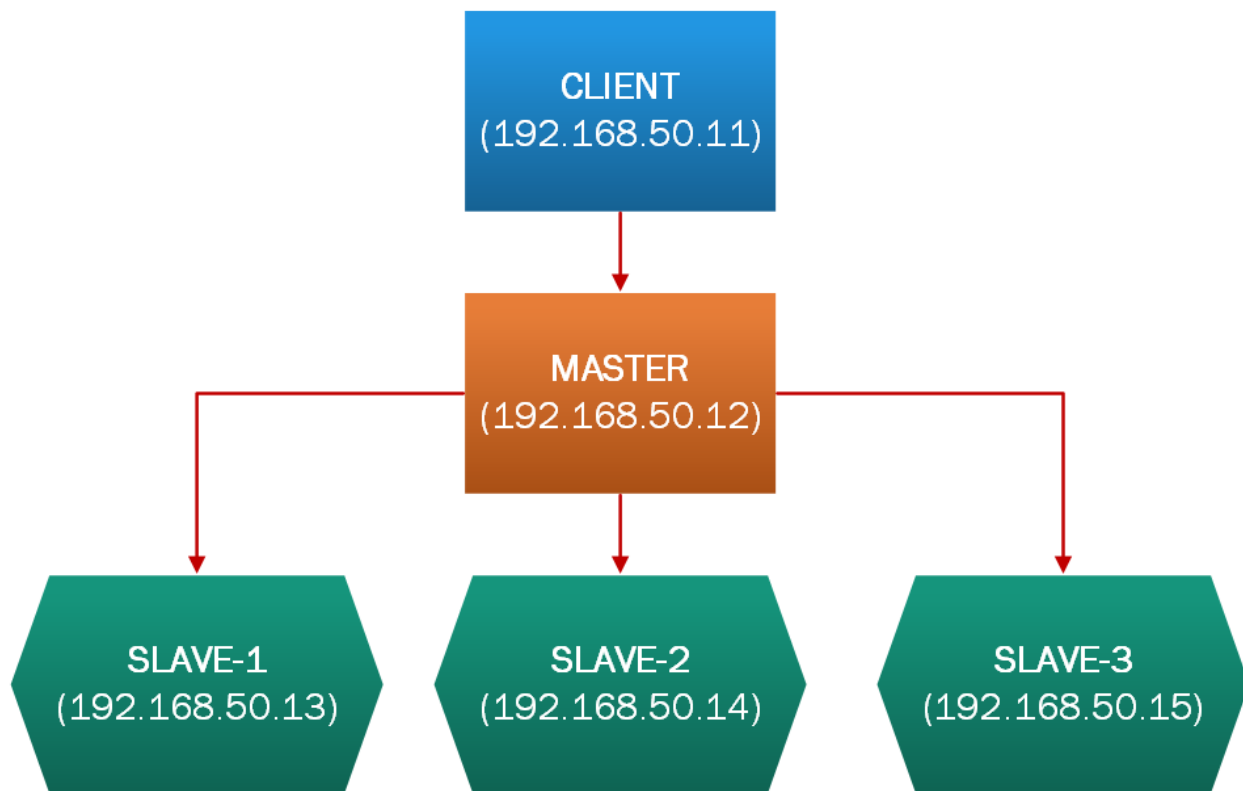


Fig. 1. Basic Model

First of all, we will need Ubuntu 14.04. The software of Ubuntu is available in our package named “Software & Documents”. Or can be downloaded from:

<http://www.ubuntu.com/download/desktop/>

Please ensure that software is in bootable form. We would recommend to use <https://rufus.akeo.ie/> to make your USB bootable. Or follow the steps given below:

1.1 Bootable Steps

Install Rufus Software from the above link. Place Ubuntu 14.04 Software in a directory. Now you can start by opening up Rufus. Don’t worry about any of the settings here as Rufus will automatically do this for you depending on the type of ISO you are using. Run Rufus:

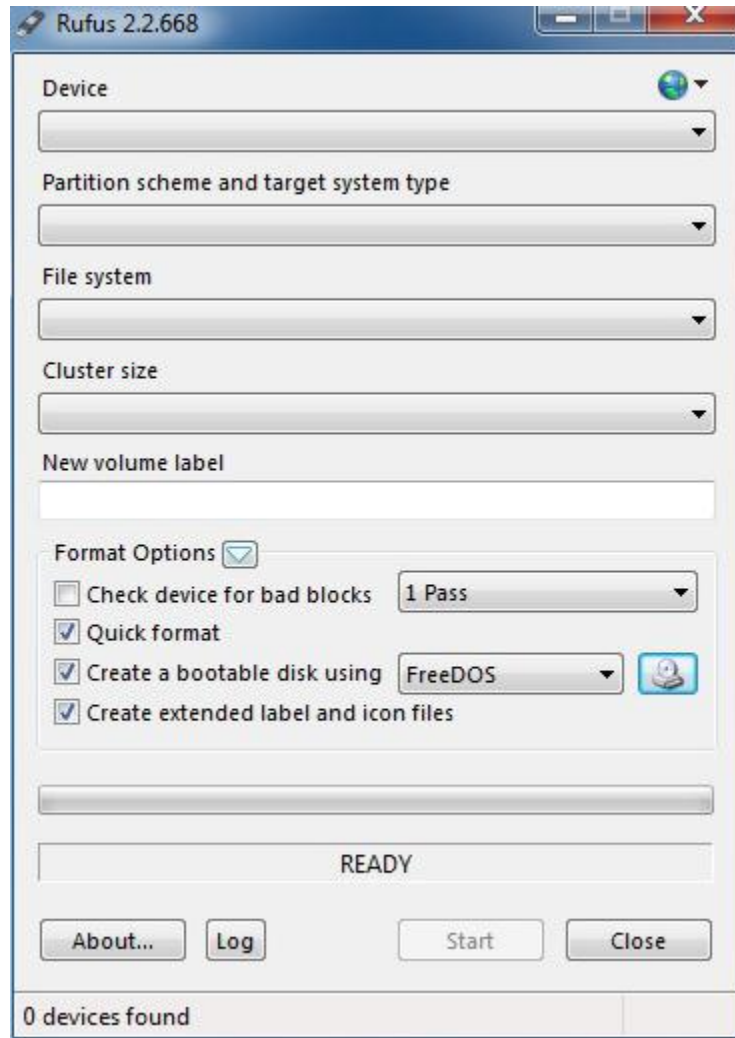


Fig. 2. How to use Rufus Step 1

By default, it will appear to have many of the options blank until you connect your USB. Once you connect the USB, these fields will be automatically filled for you. Next, connect your USB to your computer. Now we need to select our ISO (Ubuntu Software, in this case the Ubuntu 14.04 ISO). Select the icon highlighted as shown below, which will let you browse for the required ISO file.

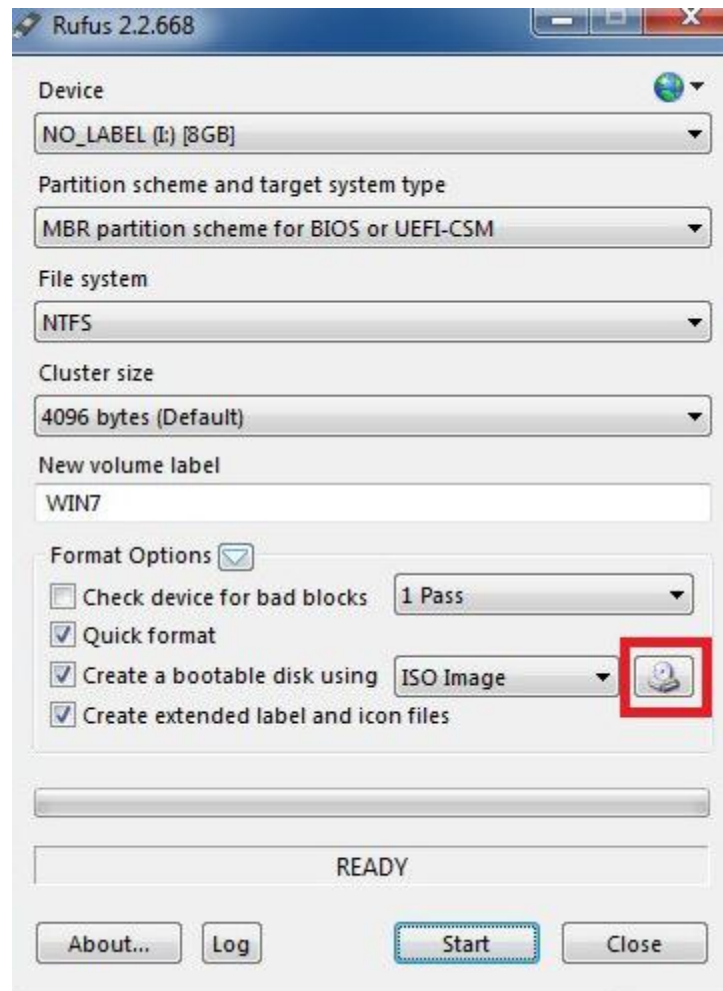


Fig. 3. How to use Rufus Step 2

Rufus will now have filled out the fields for you. All you have to do now is to hit Start button. If your USB is correctly interfaced with the software then you will get a standard warning telling you that any data/partitions that was on the USB before starting the process will be permanently deleted.

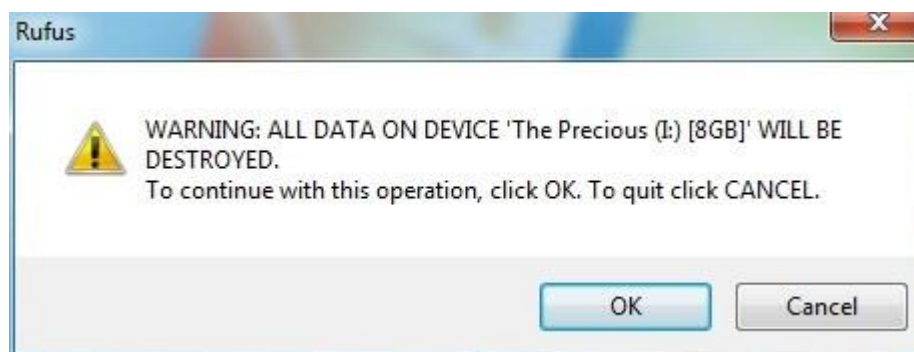


Fig. 4. How to use Rufus Step 3

Click OK, to start.

How long the process will take to complete will depend on the specs of your computer, the USB (read/write speeds) and the size of the ISO you are using. That's it. Now you have a bootable USB.

1.2 Installation of Ubuntu

Read the instruction below or go to link (if available) and install Ubuntu.

<https://ivanblagojevic.com/how-to-install-ubuntu-14-04-lts-on-an-empty-hard-disk-tutorial/>

Before we do the installation proper, we will focus on all the important things that you need to take into consideration. By itself, Linux in general, and Ubuntu in particular, is not any more difficult to master than any which operating system you choose. But it is foreign to most Windows users, who have never tried Linux before. Thus, it is not so much the installation experience that is difficult, it's the whole new world of concepts.

Still, in order to successfully install ANY operating system, including Ubuntu, you should be familiar and comfortable with several basic and intermediate concepts.

- You need to be able to back-up your data.
- You need to be able to download the installation image (say ISO file). (Done!)
- You need to be able to burn the installation image to a media (DVD, USB). (Done!)
- You need to be able to boot your system from DVD or USB. (Done!)
- You need to be familiar with disk management and partitioning.
- You need to understand basic Linux concepts.

1.2.1 Ubuntu Installation Steps

Here are some pre-required items that must be met:

- 6.3 GB available drive space
- internet connection

Start your computer, connect bootable USB and on the first screen you will see a button that allows you to adjust the BIOS. Often it is **F2**, **F12**, **ESC** etc. Press the button accordingly. Once you are in BIOS settings, look for BOOT options (like boot Device Priority). Select the removable device in the first place. Save the settings. The computer will restart and boot from the disc that we have prepared.

If you have successfully completed the first step it will lead you to your new operating system.

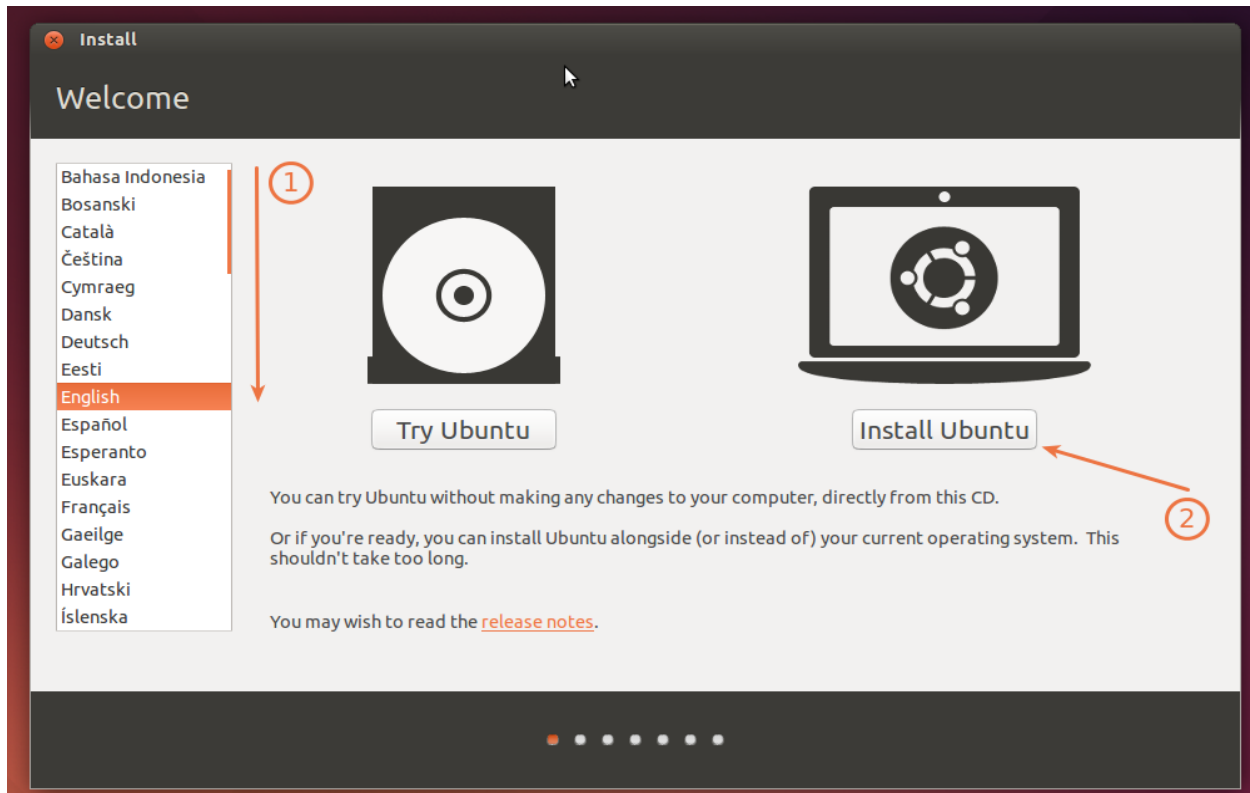


Fig. 5. How to install Ubuntu Step 1

1. Select the language you want to use in Ubuntu
2. Click on Install Ubuntu button.

After that click on Continue.

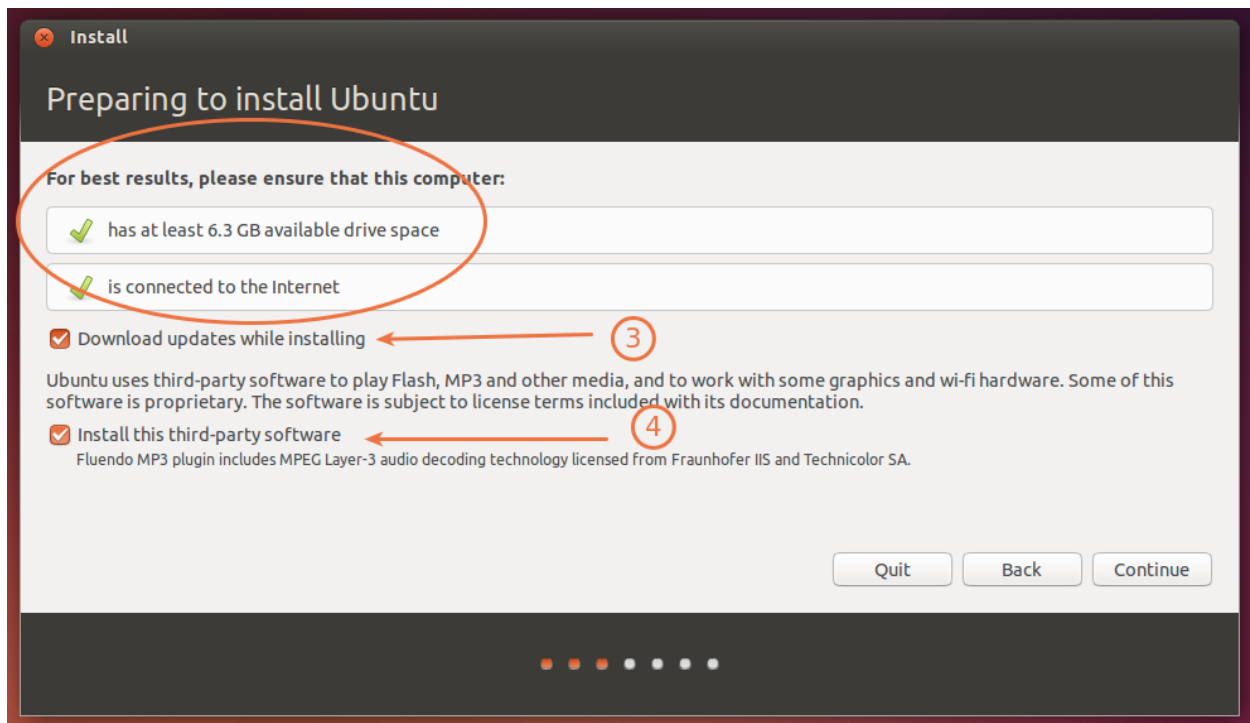


Fig. 6. How to install Ubuntu Step 2

Now, click on Erase disk and install Ubuntu.

3. Set your location.
4. Select your language.
5. Select your keyboard layout.

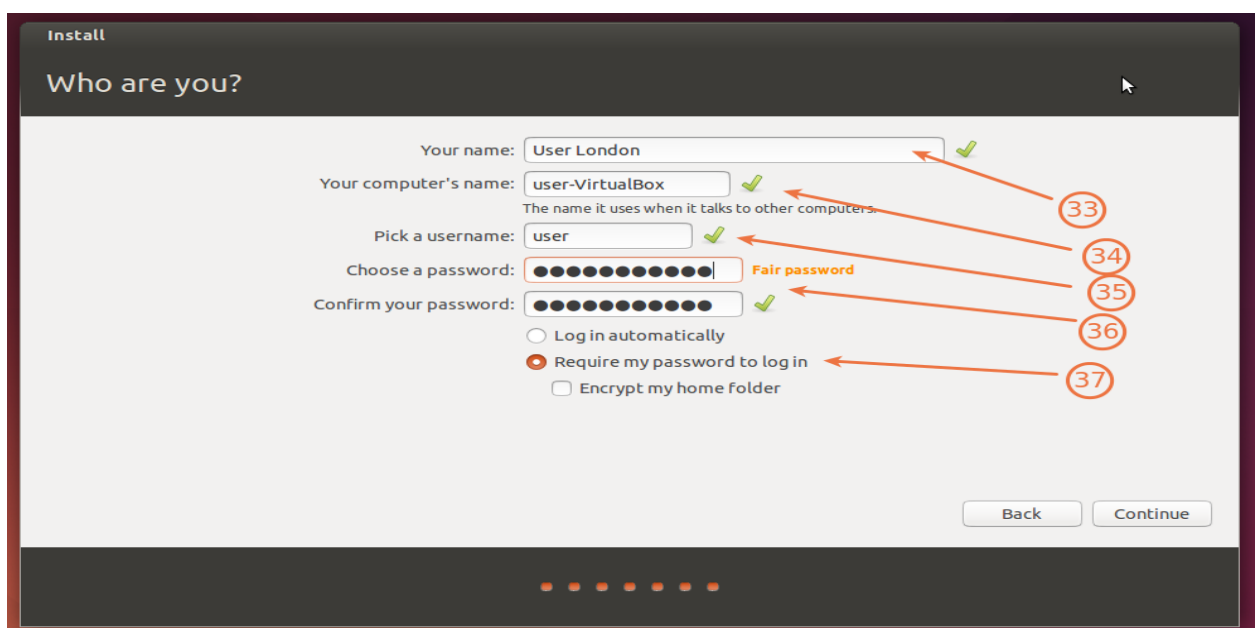


Fig. 7. How to install Ubuntu Step 3

1.2.2 Welcome to Ubuntu!

Wait 15-25 minutes to complete installation and that's it!

This is the procedure to install the Ubuntu operating system on completely empty, new, or deleted hard disk. Also, this is the answer to beginners' question: How to install Ubuntu (any edition) on an empty hard disc? The procedure to install Ubuntu on the disk with existing partitions is part of a brand new story.

By using above procedure, install Ubuntu 14.04 on all P.Cs.

MySQL-Replication

2 What is MySQL Replication?

Replication enables data from one MySQL server (the master) to be replicated to one or more MySQL servers (the slaves). MySQL Replication is very easy to setup, and is used to scale out read workloads, provide high availability and geographic redundancy, and offload backups and analytic jobs.

2.1 Replication Scheme

There are currently two replication schemes supported by MySQL Replication:

- Asynchronous replication
- Semi-synchronous replication

There is no restriction in mixing replication schemes in the same topology. Both have their pros and cons.

3 Setting up Replication Slaves

We'll now deploy a MySQL replication topology consisting of one master and three slaves. Our architecture is illustrated below:

First of all, we will install MySQL on Master & Slave by using:

```
sudo apt-get install -y mysql-server
```

Do this on Master:

Edit **my.cnf** file, add ip of P.C, server id (e.g, 1) and enable binary log

```
sudo service mysql restart
```

```
sudo mysql -uroot -p
```

```
create user 'repl'@'%' identified by 'slavepassword';
```

```
grant replication slave on *.* to 'repl'@'%';
```

```
exit --This will exit mysql shell
```

On home terminal pass the following commands:

```
Sudo mysqldump -uroot -p --all-databases --master-data > masterdump.sql
```

```
grep CHANGE *sql | head -1
```

Copy **masterdump.sql** to slave (to pass the following command)

scp masterdump.sql 192.168.50.13:

Where, 192.168.50.13 is the IP address of Slave. Now, go to Slave:

Edit **my.cnf** file, add ip of P.C, server id (e.g, 2)

sudo service mysql restart

sudo mysql -uroot -p

CHANGE MASTER TO MASTER_HOST='Master Machine IP',

MASTER_USER='repl',MASTER_PASSWORD='slavepassword';

exit --This will exit mysql shell

sudo mysql -uroot -p < masterdump.sql (If masterdump.sql is on home directory otherwise specify)

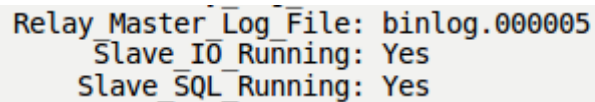
sudo mysql -uroot -p

start slave;

The replication status can only be checked from a replicating slave by using the following statement:

show slave status\G;

If slave status show that sql and I/O running yes then good. Your replication is working.



```
Relay Master Log File: binlog.000005
Slave IO Running: Yes
Slave SQL Running: Yes
```

Fig. 8. MySQL Replication Test

3.1 Replication-Test

Go to master:

sudo mysql -uroot -p

create database test;

show databases; (You can see database test is created)

Go to Slave:

sudo mysql -uroot -p

show databases; (You can see database test is created)

Excellent, Replication is working.

If you want to replicate on multiple slaves then create new user on master grant privileges then do rest of the steps as above.

If you want to send data from Client to master MySQL remotely.

Do this on Master MySQL:

*grant all privileges on *.* to 'root'@'%' identified by 'password_here' with grant option;*

Do this on Client:

Sudo mysql -uroot -p -h 192.168.50.12

If you are able to login then you can send data to master MySQL remotely.

The link below is also useful for configuration when there is only one master and one slave. If you want to use multiple slaves then you will have to create users in Master's database for each slave and grant privileges. To create users, follow the same method as used in case of one slave.

<https://www.youtube.com/watch?v=JXDuVypcHNA>

As outcome, master slave replication will be working successfully.

4 PostgreSQL-Replication

Link for replication of data using PostgreSQL database:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-master-slave-replication-on-postgresql-on-an-ubuntu-12-04-vps>

5 Cassandra DB Replication on Ubuntu 14.04

Apache Cassandra is a highly scalable open source database (NoSQL) system, achieving great performance on multi-node setups. Because you're about to build a multi-node Cassandra cluster, you must determine how many servers you'd like to have in your cluster and configure each of them. It is recommended, but not required, that they have the same or similar specifications.

In my case, I will install three node (one client, one master & one slave) Cassandra Cluster. IPs are given below:

Client: 192.168.50.11

Master: 192.168.50.12

Slave: 192.168.50.13

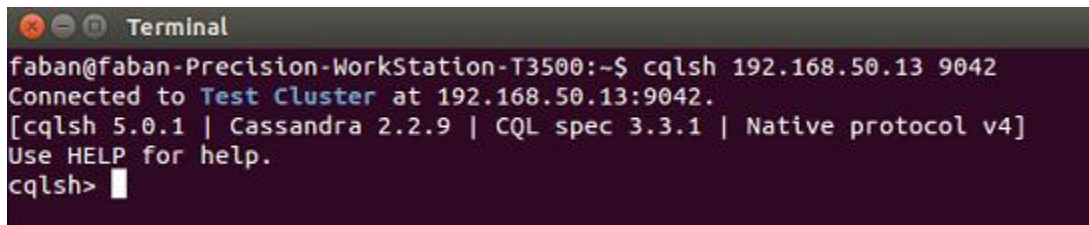
To complete this tutorial, you'll need the following:

- At least two Ubuntu 14.04 servers configured using <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04>
- Each server must be secured with a firewall using <https://www.digitalocean.com/community/tutorials/how-to-implement-a-basic-firewall-template-with-iptables-on-ubuntu-14-04>
- Each server (only Master & Slave in our case) must also have Cassandra installed by following <https://www.digitalocean.com/community/tutorials/how-to-install-cassandra-and-run-a-single-node-cluster-on-ubuntu-14-04>

Once, you have installed Cassandra on all servers. It's time to setup multi node Cassandra server for data replication. Follow the link given below to setup multimode Cassandra Cluster.

<https://www.digitalocean.com/community/tutorials/how-to-run-a-multi-node-cluster-database-with-cassandra-on-ubuntu-14-04>

After configuration of multi node Cassandra, you can login to any server by providing the IP of required server. For example, if you are at .12 the you can login to .13 by using command



```

faban@faban-Precision-WorkStation-T3500:~$ cqlsh 192.168.50.13 9042
Connected to Test Cluster at 192.168.50.13:9042.
[cqlsh 5.0.1 | Cassandra 2.2.9 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cqlsh>

```

Where, 9042 is port number. Now, you are at .13 server. Similarly, if you want to login to .12 then on .12, simply pass 192.168.50.12 instead of .13 in above command.

Now, login to 192.168.50.12 to setup replication.

Cqlsh 192.168.50.12 9042

On cqlsh shell, promote following commands.

Create keyspace keyspace_name_here with replication = {'class' : 'SimpleStrategy' , 'replication_factor' : 2};

Set **replication_factor** to 2 if you want to replicate data to one slave.

After completing above step, you replication configuration is complete. Now, I will show you an example to test replication.

Example

You are at 192.168.50.12 (Master).

Step 1.


```
[cqlsh 5.0.1 | Cassandra 2.2.9 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE mydb WITH REPLICATION = { 'class' : 'SimpleStrategy', '
replication_factor' : 2 };■
```

Step 2.

```
cqlsh> use mydb;
cqlsh:mydb> create table test (id int PRIMARY KEY, data blob);■
```

Step 3.

```
cqlsh:mydb> describe table test;

CREATE TABLE mydb.test (
  id int PRIMARY KEY,
  data blob
) WITH bloom_filter_fp_chance = 0.01
AND caching = '{"keys":"ALL", "rows_per_partition":"NONE"}'
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCom
pactionStrategy'}
AND compression = {'sstable_compression': 'org.apache.cassandra.io.compress.
LZ4Compressor'}
AND dclocal_read_repair_chance = 0.1
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.0
AND speculative_retry = '99.0PERCENTILE';

cqlsh:mydb> ■
```

You can see that table **test** has been created at .12. Now, login to Slave Server (192.168.50.13).

```

faban@faban-Precision-WorkStation-T3500:~$ cqlsh 192.168.50.13 9042
Connected to Test Cluster at 192.168.50.13:9042.
[cqlsh 5.0.1 | Cassandra 2.2.9 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cqlsh> use mydb;
cqlsh:mydb> describe tables;

students test images2 dogtypes books images crime

cqlsh:mydb> describe table test;

CREATE TABLE mydb.test (
  id int PRIMARY KEY,
  data blob
) WITH bloom_filter_fp_chance = 0.01
  AND caching = '{"keys":"ALL", "rows_per_partition":"NONE"}'
  AND comment = ''
  AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}
  AND compression = {'sstable_compression': 'org.apache.cassandra.io.compress.LZ4Compressor'}
  AND dclocal_read_repair_chance = 0.1
  AND default_time_to_live = 0
  AND gc_grace_seconds = 864000

```

You can see that table **test** has also been created on slave side as well. Congratulations! Replication is working now.

5.1 Login to Cassandra using Username & Password

http://docs.datastax.com/en/cassandra/2.1/cassandra/security/secure_login_cqlsh_t.html

5.1.1 Example

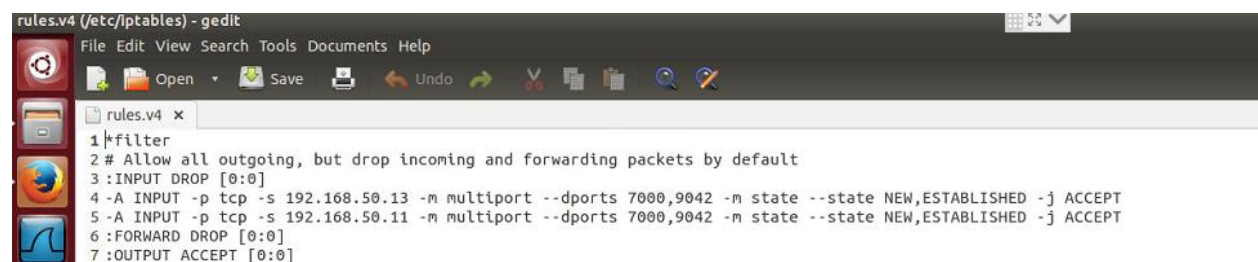
```

faban@faban-Precision-WorkStation-T3500:~$ cqlsh -u cassandra -p cassandra 192.168.50.12 9042
Connected to Test Cluster at 192.168.50.12:9042.
[cqlsh 5.0.1 | Cassandra 2.2.9 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cassandra@cqlsh>

```

Till now, we are done with replication setup. Now, it's time to do experiment. In order to do experiment of Cassandra database, go to Master and edit following line under `sudo gedit /etc/iptables/rules.v4`

`-A INPUT -p tcp -s your_source_server_ip -m multiport --dports 7000,9042 -m state --state NEW,ESTABLISHED -j ACCEPT`



After adding the rule, save and close the file, then restart IPTables & Cassandra.

```
sudo service iptables-persistent restart
```

```
sudo service cassandra restart
```

Now, you will be able to send queries to Cassandra DB from remote host (client). In order to verify, promote following command on home terminal of Client's PC:

```
cqlsh -u cassandra -p cassandra 192.168.50.12 9042
```

6 Experiments

First of all, **twitter's archive file** of a particular user (@Karam_Shehzad) was downloaded. To download and view your Twitter archive:

1. Login to your twitter account, go to your **account settings** by clicking on the **profile icon** at the top right of the page and selecting **Settings** from the drop-down menu.
2. Click **Request your archive**.
3. When your download is ready, twitter's team will send an email with a download link to the confirmed email address associated with your Twitter account.
4. Once you receive the email, click the **Go now** button to log in to your Twitter account and download a .zip file of your Twitter archive.

Twitter's data was in .csv format which was consisted of Tweets & Retweets of a particular user. Finally, this data was inserted in Master's database which was replicated on slave as well & traffic was captured during the replication by using Wire-shark.

Secondly, a .csv file that was consisted of crime reports in USA was replicated. Thirdly, an image set (500 items) was replicated.

6.1 Experiments of PostgreSQL

You are at Client's P.C

```

1 import psycopg2
2 import csv
3 filename='/home/faban/Downloads/Python/Python-Postgre/tweets.csv'
4 with open (filename,'rU') as csvfile:
5     csv_data=csv.reader(csvfile)
6
7 #csv_data = csv.reader(file('/home/faban/Python-Postgresql/Crime.csv'))
8
9 database = psycopg2.connect (host= "192.168.50.12",database = "postgres", user="postgres", password="faban")
10 cursor = database.cursor()
11 count_row=0
12 cursor.execute("Drop table if exists Tweets")
13 cursor.execute("create table Tweets(a text,b text, c text, d text, e text, f text, g text, h text, i text, j text)")
14 print('Table Created Successfully')
15 for row in csv_data:
16     name=row[0]
17     city=row[1]
18     country=row[2]
19     d=row[3]
20     e=row[4]
21     tweet=row[5]
22     g=row[6]
23     h=row[7]
24     i=row[8]
25     j=row[9]
26     print("Inserted row number:",count_row)
27     count_row+=1
28 # print(count)
29 cursor.execute("INSERT INTO Tweets VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", (name,city,country,d,e,tweet,g,h,i,j))
30 # csv_data.close()
31 # cursor.close()
32 database.commit()
33 database.close()
34
35 print "CSV data imported"

```

Fig. 9. Python Code for Tweets Data

In order to run above code, first of all you need to create a database at Master's PostgreSQL. To do that, go to Master's P.C:

su – postgres

psql

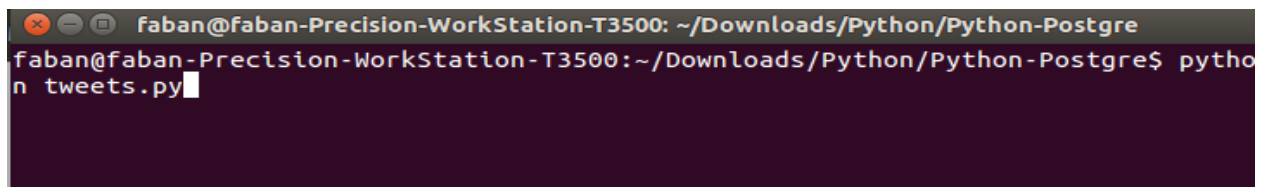
create database postgres;

Now, go to client's P.C and place **tweets.csv** file in your desired directory. For example, above code, file is in:

/home/faban/Downloads/Python-Postgre

After that, copy above code and save in a file. For example "tweets.py". Now promote following command in the terminal.

Python tweets.py

A terminal window with a dark background. The title bar shows 'faban@faban-Precision-WorkStation-T3500: ~/Downloads/Python/Python-Postgre'. The command prompt shows 'faban@faban-Precision-WorkStation-T3500:~/Downloads/Python/Python-Postgre\$ python tweets.py' with a cursor at the end of the line.

```
faban@faban-Precision-WorkStation-T3500: ~/Downloads/Python/Python-Postgre
faban@faban-Precision-WorkStation-T3500:~/Downloads/Python/Python-Postgre$ python tweets.py
```

Fig. 10. Command to run Python Code

As a result, above code will start inserting tweets into PostgreSQL Master's database (192.168.50.12). When the above code will be executed then you can see all of your queries in the master's database. Go to Master's database and do following steps:

su – postgres

psql

\d

Now, you can see table **"Tweets"**. Now, type:

select id from Tweets;

*select * from Tweets;*

You can see that all the tweets are inserted into database.

Now, in order to see whether replication is working or not, go to respective slave and do above steps. You can see that all the tweets are also been replicated to slave as well. Congratulations!

Similarly, you can insert Crime's & Images' data in PostgreSQL database. Code is available in the directory:

/RA Work-Replication/Replication/Python/Python-Postgre

6.2 MySQL

You are at Client's P.C

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 import MySQLdb as mdb
4 import psycopg2
5 import sys
6 import MySQLdb
7
8 def read_image(i):
9
10     filename="/home/faban/Downloads/Python/Python-MySQL/images/im"
11     filename=filename+str(i)+".jpg"
12     print(filename)
13     fin = open(filename)
14     img = fin.read()
15
16     return img
17
18 |
19
20 con = MySQLdb.connect("192.168.50.12","root","faban","experiments" )
21 with con:
22     print('connecting to database')
23     range_from=input('Enter range from:')
24     range_till=input('Enter range till:')
25     for i in range(range_from,range_till):
26         cur = con.cursor()
27         data = read_image(i)
28         cur.execute("INSERT INTO images VALUES(%s, %s)", (i,data, ))
29         cur.close()
30         con.commit()
31 con.close()

```

Fig. 11. Python Code for Flicker Images

In order to run above code, you need to create database at Master's MySQL database. To do that, go to Master's database:

sudo mysql -uroot -p

Enter Paswword: *faban*

create database experiments;

Now, you will have to create a table as well. To do that:

use experiments;

create table Images (id int, data MEDIUMBLOB)

show tables;

You can see the table Images. Excellent!

Now go to Client's P.C and place **images' folder** in your desired directory. For example, in above code, file is in:

/home/faban/Downloads/Python-MySQL

After that, copy above code and save in a file. For example “images.py”. Now promote following command in the terminal.

Python images.py

Now, insert range of images. For example: 1-500

As a result, the above code will start inserting images into MySQL Master's database (192.168.50.12)

When the above code will be executed, then you can see all of your queries in the master's database. Go to Master's database and do following steps:

sudo mysql -uroot -p

Enter Paswword: faban

use experiments;

show tables;

Now, you can see table “**Images**”. Now, type:

select id from Images;

*select * from Images;*

You can see that all the images are inserted into table.

In order to see whether replication is working or not, go to respective slave and do above steps. You can see that all the tweets have also been replicated to slave as well. Congratulations!

Similarly, you can insert Tweets' & Crimes' data in MySQL database. Code is available in the directory:

/RA Work-Replication/Replication/Python/Python-Mysql

6.3 Cassandra

In order to do experiments for Cassandra DB, first of all, we need to install few drivers on Client machine (192.168.50.11). Follow the link to install Cassandra-drivers.

sudo apt-get install libcurl4-gnutls-dev librtmp-dev

sudo pip install --install-option="--no-cython"


```
sudo pip install cassandra-driver
```

```
sudo pip install cql
```

```
sudo apt-get install -y python-pip
```

```
sudo easy_install -U setuptools
```

```
sudo pip install setuptools-cython
```

<https://datastax.github.io/python-driver/installation.html>

https://stackoverflow.com/questions/44154551/populating-cassandra-database-using-python/44163441?noredirect=1#comment75583288_44163441

You might face few errors, while installing Cassandra-Drivers. I faced many problems during the installation, Googling helped me. At the end, `sudo pip install --install-option="--no-cython"` saved my day.

6.3.1

You are at Client's P.C

```
cassandra_images.py x
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 from cassandra.cluster import Cluster
4 from cassandra.auth import PlainTextAuthProvider
5 from cassandra.cluster import SimpleStatement
6 import sys
7 #reading my hostname, username, and password from the command line; defining my Cassandra keyspace as as variable.
8 hostname=sys.argv[1]
9 username=sys.argv[2]
10 password=sys.argv[3]
11 keyspace="excelsior"
12 #adding my hostname to an array, setting up auth, and connecting to Cassandra
13 nodes = []
14 nodes.append(hostname)
15 auth_provider = PlainTextAuthProvider(username=username, password=password)
16 ssl_opts = {}
17 cluster = Cluster(nodes,auth_provider=auth_provider,ssl_options=ssl_opts)
18 session = cluster.connect(keyspace)
19 def read_image(i):
20
21     filename="/home/faban/Downloads/Python/Python-Mysql/images/im"
22     filename=filename+str(i)+".jpg"
23     print(filename)
24     fin = open(filename)
25     img = fin.read()
26     return img
27 with session:
28     print('connecting to database')
29     range_from=input('Enter range from:')
30     range_till=input('Enter range till:')
31     for i in range(range_from,range_till):
32         data = read_image(i)
33         strCQL = "INSERT INTO images (id,data) VALUES (?,?)"
34         pStatement = session.prepare(strCQL)
35         session.execute(pStatement,[i,data])
36 session.shutdown()
```

In order to run above code, you need to create keyspace “excelsior” at Master's Cassandra database. To do that, go to Master's database:

```
Terminal
faban@faban-Precision-WorkStation-T3500:~$ cqlsh -u cassandra -p cassandra 192.168.50.12 9042
Connected to Test Cluster at 192.168.50.12:9042.
[cqlsh 5.0.1 | Cassandra 2.2.9 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cassandra@cqlsh> CREATE KEYSPACE excelsior WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 2 };
```

create table images (id int PRIMARY KEY, data blob);

Now go to Client's P.C and place **images' folder** in your desired directory. For example, in above code, file is in:

/home/faban/Downloads/Python-MySQL

After that, copy above code and save in a file. For example "images.py". Now promote following command in the terminal.

```
faban@faban-Precision-WorkStation-T3500: ~/Desktop
faban@faban-Precision-WorkStation-T3500:~/Desktop$ python cassandra_images.py 192.168.50.12 cassandra cassandra
```

Now, insert range of images. For example: 1-500

As a result, the above code will start inserting images into Cassandra Master's database (192.168.50.12)

When the above code will be executed, then you can see all of your queries in the master's database. Go to Master's database and do following steps:

```
Terminal
faban@faban-Precision-WorkStation-T3500:~$ cqlsh -u cassandra -p cassandra 192.168.50.12 9042
Connected to Test Cluster at 192.168.50.12:9042.
[cqlsh 5.0.1 | Cassandra 2.2.9 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cassandra@cqlsh> use excelsior ;
cassandra@cqlsh:excelsior> describe tables;

tweets  crime2  images3  images2  test  books  images

cassandra@cqlsh:excelsior> SELECT COUNT(*) FROM images;

count
-----
499

(1 rows)
```

You can see that all 499 images have been inserted to Cassandra DB. You can do same steps on Slave & make sure that dataset also replicated as well.

Similarly, you can insert Tweets' & Crimes' data in Cassandra database. Code is available in the directory:

[/RA Work-Replication/Replication/Python/Python-Cassandra](#)

Note: Don't use `select * from images;` command under Cassandra shell though it works but it is very expensive for Cassandra DB.

You can also check properties of a table by promoting following command on home terminal:

```
nodetool cfstats -- keyspace_name_here.table_name_here
```

```
nodetool cfstats -- excelsior.images
```

7 Automation

Till now, we have done all the experiments using manual approach which is very hectic and time consuming. That's why we have written a code using shell script to do all the above experiments automatically. The code for MySQL and PostgreSQL's experiments is available in [/RA Work-Replication/Replication/Script](#) directory.

To do experiments, make sure all of your P.Cs are running, connected to internet and replication is working. Secondly, all the files (given in Scripts' folder) that are mentioned in Script's code are at required directories. Thirdly, Wire-Shark is installed on all P.Cs. I will explain later in this tutorial that how you can install Wire-shark on Ubuntu.

7.1 Automation of MySQL

To start the experiment for MySQL. First of all, do following steps:

1. Place **s_display.sh** within Script folder (Script folder should be at Client's P.C)
2. [Python/Python-Mysql](#) (All files are inside this directory, this folder should be at Client's side)
3. Place **cpu_performance.sh** file in the home directory of Master's Machine
4. Place **stop_tcpdump.sh** in the Home directory of Client, Master and Slave1,2,3
5. Place **stoptop.sh** in the home directory of Master's machine

Note: All above files are available in Script folder.

Go to Script folder (you can see MySQL.sh file here) and promote following command at terminal:

```
./MySQL.sh
```

```

faban@faban-Precision-WorkStation-T3500:~/Script$ ./Mysql.sh
Lets start...
Which dataset you want to use?
For Tweets--> insert 1
For Crime--> insert 2
For Images--> insert 3
: 1
Enter number of iterations
Please enter any number between 1 to N
: 2
Enter number of slaves to replicate data
Please enter any number between 0 to 3
: 2
2_MySQL_slave_tweets

```

Fig. 12. Inputs for MySQL Experiment

As a result, you will be asked to input the required values. For example, how many times you want to repeat the experiment? Which dataset you want to use? Number of Slaves to replicate data? In above figure, I have selected 1 which means, I want to insert Tweets data to MySQL database. I have selected 2 number of iterations which means that I want to repeat experiment two times. And number of slaves 2 which means I want to repeat data to two slaves as well. You can see at the end of above figure **2_MySQL_slave_tweets**. This is the name of Wire-shark's file according to your inputs, which will be created at Client, Master and slaves respectively.

When this program will be completed then you can see all the Wire-shark files are created at the specified directories. We wish you good luck for the collected Results!

7.2 Automation of PostgreSQL

In PostgreSQL shell script, there is also some features that automatically set replication if somehow it is not working. A positive point!

To start the experiment for MySQL. First of all, do following steps:

1. Place **s_display.sh** within Script folder (Script folder should be at Client's P.C)
2. **Python/Python-Postgre** (All files are inside this directory, this folder should be at Client's side)
3. Place **cpu_performance_postgres.sh** file in the home directory of Master's Machine.
4. Place **stop_tcpdump.sh** in the Home directory of Client, Master and Slave1, 2, 3.
5. Place **stoptop.sh** in the home directory of Master's machine.

Note: All above file are available in Script folder.

To start the experiment for PostgreSQL. Go to Script folder (you can see Postgresql.sh file here) and promote following command at terminal:

`./Postgresql.sh`

```
faban@faban-Precision-WorkStation-T3500:~/Script$ ./Postgresql.sh
Enter How Many Times You Want To Repeat Experiment:
3
Which dataset you want to use?
For Tweets--> insert 1
For Crime--> insert 2
For Images--> insert 3
:3
With how many slaves?
3
3_slave_images
Number of slaves: 3
```

Fig. 13. Inputs for PostgreSQL Experiment

As a result, you will be asked to input the required values. For example, how many times you want to repeat the experiment? Which dataset you want to use? With how many slaves? In above figure, I have selected 3 which means, I want to insert Tweets data to PostgreSQL database. I have selected 3 number of iterations which means that I want to repeat experiment two times. And number of slaves 3 which means I want to repeat data to two slaves as well.

When this program will be completed then you can see all the Wire-shark files are created at the specified directories. We wish you good luck for collected Results!

8 Files and Folders

All data of RA Work is under the folder name “RA Work by Karam & Jam”. The size of folder is 234 GB. There are two folders inside the parent folder. One is “Faban” in which all the data related to Faban’s work is stored. Other one is “Replication” in which all the data related to replication traffic is stored.

8.1 Faban

Faban’s tutorial is available in file name “Tutorial”. Performed experiments of Faban

[/RA Work by Karam & Jam/Faban/Faban & Replication Experiments.](#)

JMeter’s (another synthetic workload generator) data is available in

[/RA Work by Karam & Jam/Faban/JMeter-Workload Generator](#)

All the software that are required to install Faban are available in

[/RA Work by Karam & Jam/Faban/Softwares](#)

Faban’s virtual machines are located in

[/RA Work by Karam & Jam/Faban/VMs](#)

KVM images are also inside the VMs folder. All the documents are available inside:

[/RA Work by Karam & Jam/Faban/Documents](#)

All the results of Faban’s experiments are available in “Faban-Reults” and “Faban-Results-Graph” file.

8.2 Replication

All the Miscellaneous files (practice codes etc)

[/RA Work by Karam & Jam/Replication/Misc.](#)

All the code and Wire-shark files related to MySQL and PostgreSQL

[/RA Work by Karam & Jam/Replication/MySQL Experiments](#)

[/RA Work by Karam & Jam/Replication/PostgreSQL Experiments](#)

Python Code, tweets, crime and flicker images’ data for MySQL, PostgreSQL & Cassandra

[/RA Work by Karam & Jam/Replication/Python](#)

All the results (new as well as old)

[/RA Work by Karam & Jam/Replication/Python](#)

“New Results” file contains all the results of new experiments. “Research Q-Ans” file contains all the explanation of new results.

Automation Scripts for MySQL and PostgreSQL are available in

[/RA Work by Karam & Jam/Replication/Script](#)

“MySQL.sh” file contains automation script to run MySQL experiments. “Postgresql.sh” file contains automation script to run PostgreSQL experiments. All the files required to run MySQL and PostgreSQL experiments are also inside **Script** folder.

All the software and Documents (papers, RA work Documentation etc.) related to Replication are available in

[/RA Work by Karam & Jam/Replication/Softwares & Documents](#)

9 Installation of Software's

9.1 How to install SSH-Server

Sudo apt-get install openssh-server

How to login to remote server using SSH

ssh user_name@ip_address

For example,

ssh faban@192.168.50.12

OR

ssh 192.168.50.12

If you want to login to remote server without using password then you will have to copy id to remote server first.

Ssh-keygen -t rsa

This creates a new ssh key. Press “Enter Button” on every phrase.

The entire key generation process looks like this:

```

ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/demo/.ssh/id_rsa.
Your public key has been saved in /home/demo/.ssh/id_rsa.pub.
The key fingerprint is:
4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 demo@a
The key's randomart image is:
+--[ RSA 2048]-----+
|           .oo.      |
|          . o.E      |
|         + . o       |
|        . = = .      |
|       = S = .       |
|      o + = +        |
|     . o + o .       |
|          . o        |
|                   |
+-----+

```

Fig. 14. Steps to Perform SSH Login without using Password

The public key is now located in `/home/demo/.ssh/id_rsa.pub`. The private key (identification) is now located in `/home/demo/.ssh/id_rsa`

Once the key pair is generated, it's time to place the public key on the virtual server that we want to use.

You can copy the public key into the new machine's `authorized_keys` file with the `ssh-copy-id` command. Make sure to replace the example username and IP address below.

`ssh-copy-id faban@192.168.50.12` or `ssh-copy-id 192.168.50.12`

No matter which command you chose, you should see something like:

```
The authenticity of host '12.34.56.78 (12.34.56.78)' can't be established.  
RSA key fingerprint is b1:2d:33:67:ce:35:4d:5f:f3:a8:cd:c0:c4:48:86:12.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '12.34.56.78' (RSA) to the list of known hosts.
```

Fig. 15. Output of SSH copy ID

```
faban@faban-Precision-WorkStation-T3500:~$ ssh 192.168.50.11  
faban@192.168.50.11's password:  
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
New release '16.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Fri Mar 17 11:23:51 2017 from 192.168.50.13  
faban@faban-Precision-WorkStation-T3500:~$
```

Fig. 16. Remote Login using SSH

Now try logging into the machine, with

ssh 192.168.50.12

I've used .11 just for my own use you can specify .12. Congratulations! You are now at the home directory of .11.

It should be noted that, at first, you will have to provide password but afterwards you will no need to provide password.

9.1.1 Optional Step -Disable the Password for Root Login

Once you have copied your SSH keys unto your server and **ensured that you can log in with the SSH keys alone**, you can go ahead and restrict the root login to only be permitted via SSH keys.

In order to do this, open up the SSH config file:

sudo nano /etc/ssh/sshd_config

Within that file, find the line that includes `PermitRootLogin` and modify it to ensure that users can only connect with their SSH key:

`PermitRootLogin without-password`

Put the changes into effect:

reload ssh

If you want to uninstall your ssh-server then pass following command on terminal:

Sudo apt-get purge openssh-server

9.2 How to install Wire-Shark

Sudo apt-get install -y wireshark

Sudo addgroup -quiet -system wireshark

Sudo chown root:wireshark /usr/bin/dumpcap

Sudo setcap cap_net_raw, cap_net_admin= eip /usr/bin/dumpcap

Sudo usermod -a -G wireshark username_of_your_PC_here (For example faban)

Now, start Wire-shark. If Wire-shark doesn't start capturing packets then do following steps.

Sudo dpkg-reconfigure wireshark-common

Sudo usermod -a -G wireshark username_of_your_P.C_here (For example faban)

Sudo reboot (This command will restart your P.C)

Now, you can start Wire-shark and can capture packets.

Start [Wire-shark](#) and then click on the network interface you want to use to capture the data.

On a wired network, it will likely be eth0. Now click Start.

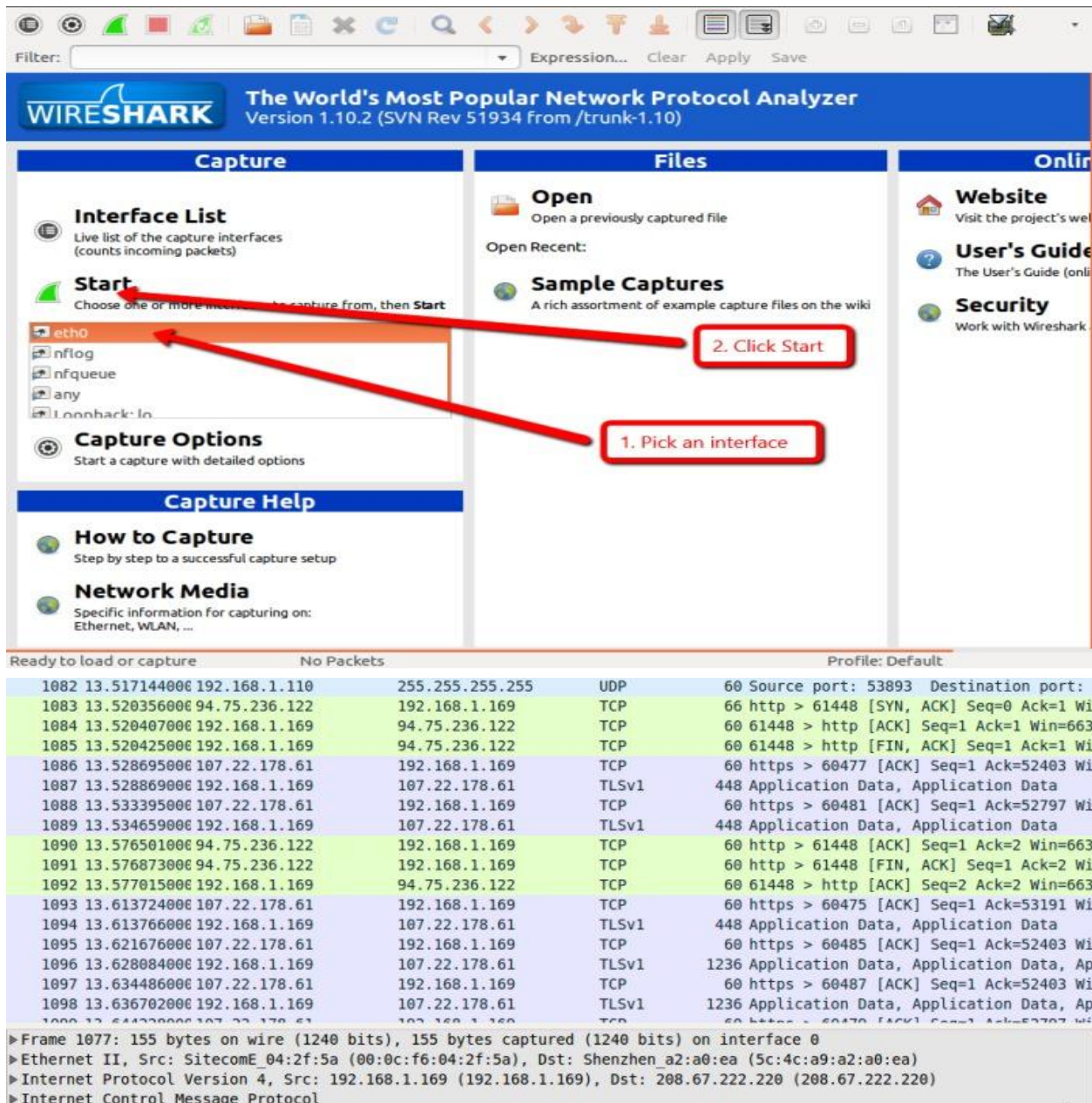


Fig. 18. Output window of Wire-shark

As a result, Wire-shark will start capturing packets like shown in fig.18.

9.3 How to install VirtualBox - Virt Manager

Sudo apt-get install virt-manager

Sudo apt-get install qemu-kvm

Run:

Sudo virt-manager

yes->Install->Continue

9.4 How to install VirtualBox

Sudo apt-get install virtualbox

10 Solutions to Problems

10.1 How to Transfer File/Folder from Host P.C to Remote

*If you want to transfer a **file** then use following command

scp file_path_here user_name_of_remote_here@IP_of_remote_here:

Or

scp file_path_here IP_of_remote_here:

The “:” at the end indicates that this files will be copied to home directory of remote machine.

You can also specify path after “:”. For example,

scp file_path_here IP_of_remote_here:/home/faban/Downloads/Code

Now, your file will be copied to **/home/faban/Downloads/Code directory**.

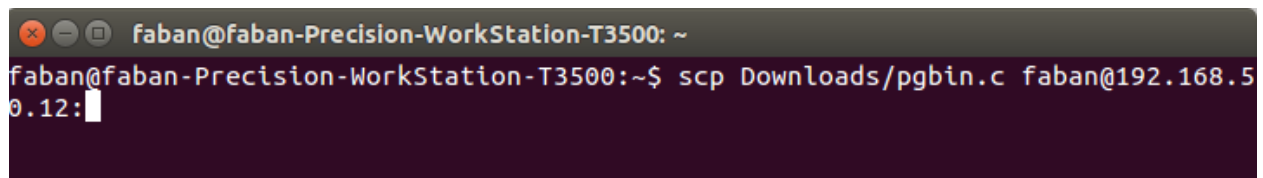


Fig. 19. Copy data to Remote Machine

10.1.1 Example

In above example, pgbin.c is my file that I wanna transfer and Downloads is the directory in which that file is located.

*If you want to transfer a **folder** then use following command

scp -r file_path_here user_name_of_remote_here@IP_of_remote_here:

Or

scp -r file_path_here IP_of_remote_here:

The “:” at the end indicates that this files will be copied to home directory of remote machine.

10.1.2 Example

```
faban@faban-Precision-WorkStation-T3500: ~  
faban@faban-Precision-WorkStation-T3500:~$ scp -r Script/ 192.168.50.15:
```

Fig. 20. Copy data to Remote Machine 2

In above example, Script is my folder that I want to transfer, which is located in the home directory of my Host machine.

10.2 How to Check IP & MAC address of a Machine

For, IP Address. Type “`ifconfig`”

```
faban@faban-Precision-WorkStation-T3500: ~  
faban@faban-Precision-WorkStation-T3500:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr bc:30:5b:e4:26:d3  
          inet addr:192.168.50.11  Bcast:192.168.50.255  Mask:255.255.255.0  
          inet6 addr: fe80::be30:5bff:fee4:26d3/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:160018 errors:0 dropped:1426 overruns:0 frame:0  
          TX packets:289847 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:30605485 (30.6 MB)  TX bytes:152414343 (152.4 MB)  
          Interrupt:17  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:204389 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:204389 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:199752874 (199.7 MB)  TX bytes:199752874 (199.7 MB)  
  
virbr0    Link encap:Ethernet  HWaddr 1e:80:76:be:f2:24  
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

Fig. 21. Output of `ifconfig`

In above picture, **inet addr: 192.168.50.11** is IP address of my P.C.

For MAC address,

`ifconfig -a`

Locate the number next to the **HWaddr**. This is your MAC address.

The MAC Address will be displayed in the form of

00:08:C7:1B:8C:02.

10.2.1 Example

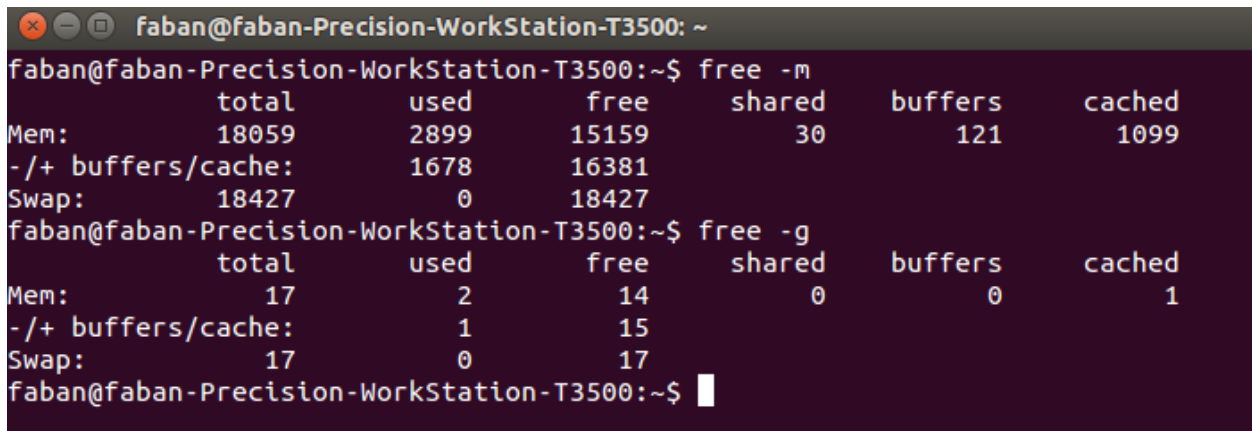
"*ifconfig -a*" output:

eth0 Link encap:Ethernet HWaddr **00:08:C7:1B:8C:02**

inet addr:192.168.50.11 Bcast:192.168.111.255 Mask:255.255.255.0

10.3 Determining free memory (RAM) on Linux

When checking the amount of free memory on a Linux server, it's easy to think you're running out of memory when you're not.



```
faban@faban-Precision-WorkStation-T3500: ~  
faban@faban-Precision-WorkStation-T3500:~$ free -m  
              total        used        free      shared    buffers     cached  
Mem:          18059         2899        15159          30         121        1099  
-/+ buffers/cache:         1678        16381  
Swap:         18427           0        18427  
faban@faban-Precision-WorkStation-T3500:~$ free -g  
              total        used        free      shared    buffers     cached  
Mem:              17           2           14           0           0           1  
-/+ buffers/cache:              1           15  
Swap:              17           0           17  
faban@faban-Precision-WorkStation-T3500:~$
```

Fig. 22. Output of *free* Command

For example, here's the output of *free -m*

free -g outputs memory in Gigabytes. *Free -m* in Megabytes.

10.4 Determining free memory (RAM) on Windows

Computer-> Properties

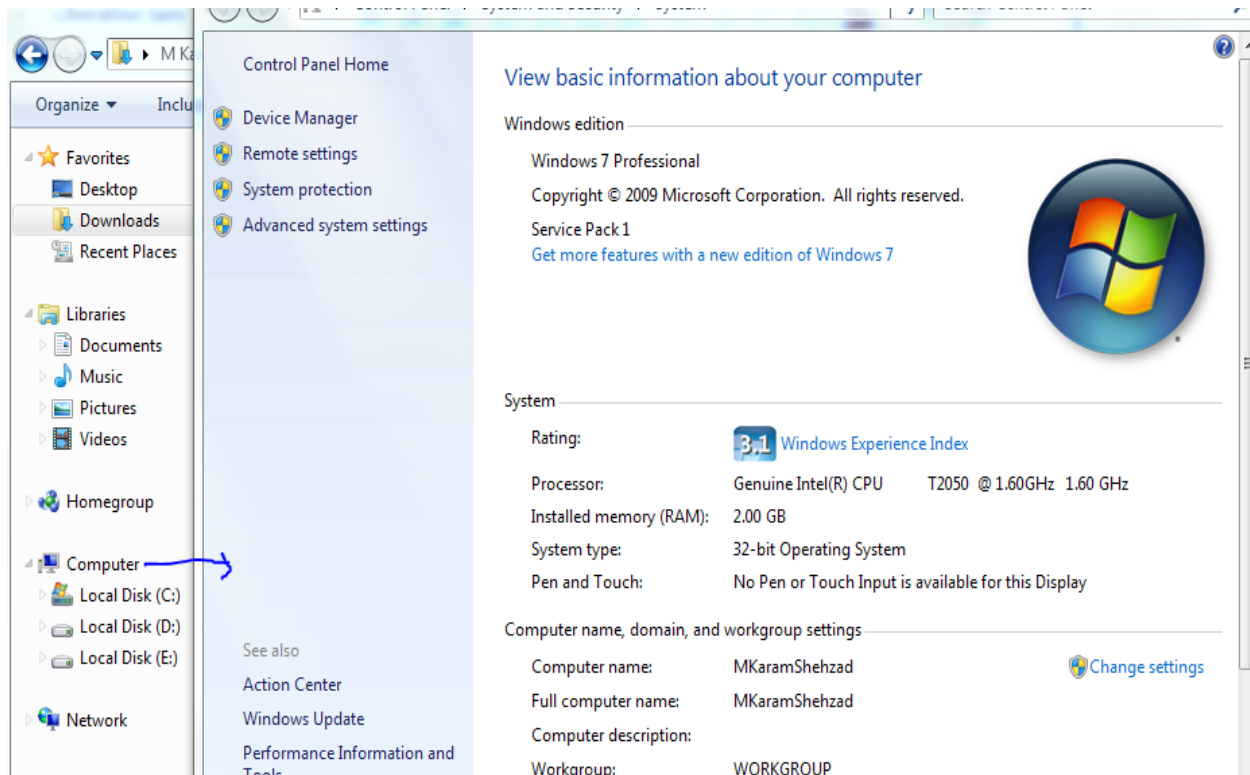


Fig. 23. RAM on Windows OS

10.5 Determining free memory (Hard Disk) on Windows

Go to run, type Create and Format-> Click on Create and Format hard disk partition.

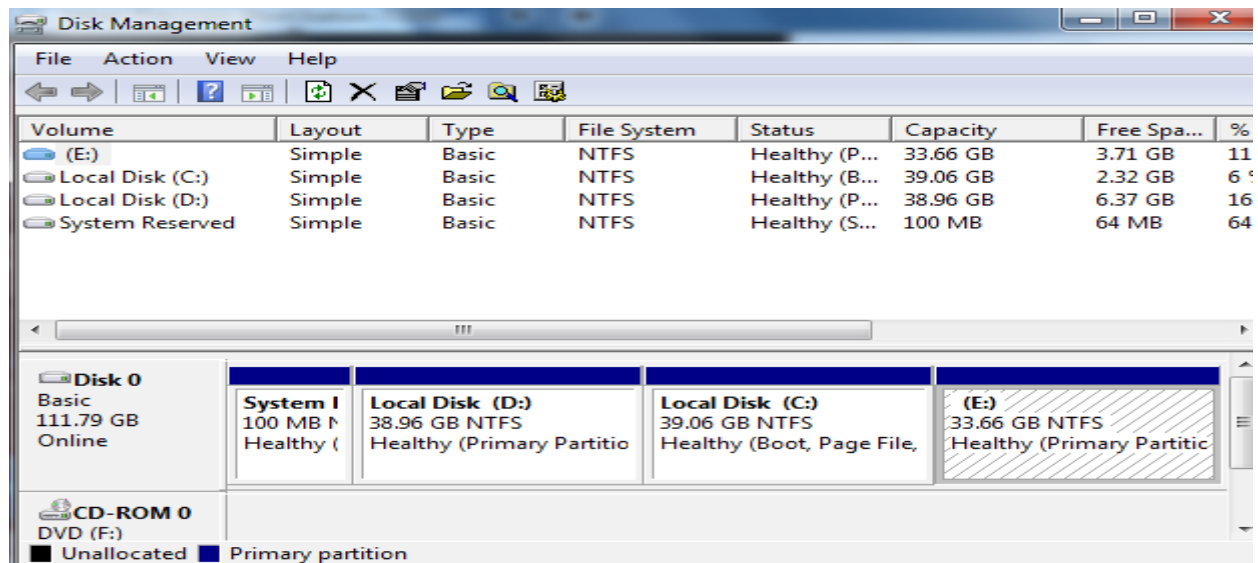


Fig. 24. Hard Disk on Windows

10.6 Determining free memory (Hard Disk) on Linux

Computer-> Properties

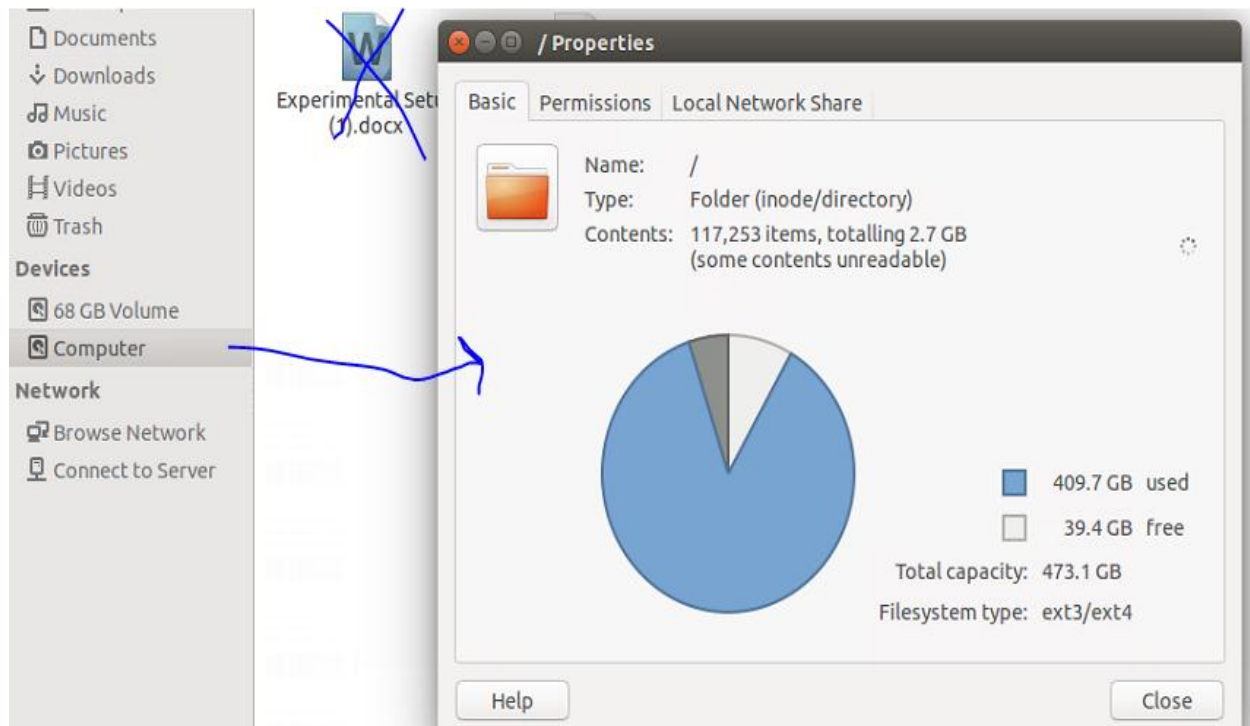


Fig. 25. Hard Disk on Ubuntu

10.7 Convert KVM File to VirtualBox

`VBoxManage convertfromraw --format VDI file_name_ Vbox.vdi`

```
faban@faban-Precision-WorkStation-T3500: ~
faban@faban-Precision-WorkStation-T3500:~$ VBoxManage convertfromraw --format VDI lv2 Vbox.vdi
```

Fig. 26. KVM to VBox

Where, lv2 is the KVM file that will be converted.

10.8 MySQL Replication Errors

10.8.1 Create dumpfile error

If by passing following commands you face the error:

`sudo mysql -uroot -p < masterdump.sql`

```
Sudo mysqldump -uroot -p --all-databases --master-data > masterdump.sql
```

Error:

```
mysqldump: error 'show create table 'proc': Table:'mysql/proc is crashed.....
```

10.8.1.1 Solution

Type following Command on terminal:

```
mysqlcheck --auto-repair -A -u username_here -p password_here
```

Inside mySQL shell type:

```
repair table 'table_name_here';
```

10.8.2 Replication not working

What if MySQL Replication is not working?

```
Show slave status\G; gives error
```

Error

```
when Slave_IO_Running: No
```

10.8.2.1 Solution

On slave, inside MySQL shell, type

```
CHANGE MASTER TO MASTER_HOST = '***192.168.50.12***', MASTER_USER = 'replicator',  
MASTER_PASSWORD = 'passwd', MASTER_LOG_FILE = 'mysql-bin.000002', MASTER_LOG_POS =  
107;
```

Where, 'replicator' is your user and 'passwd' is your password against that user that you passed on Master.

10.8.3 Replication error

What if MySQL Replication not working?

```
Show slave status\G; gives error
```

Error

```
when Slave_IO_Running: Connecting
```

10.8.3.1 Solution

add a comment



I faced the same issue with the MySQL replication.
I had given the `GRANT ALL` privileges, which solved my issue, as follows:

0

```
GRANT ALL ON *.* TO 'slave_user'@'%' IDENTIFIED BY 'password';
```



share improve this answer

edited Dec 13 '16 at 5:10

answered Dec 12 '16 at 11:45



Roopa

1 1

add a comment

Fig. 27. Solution to IO Connecting

10.8.4 Replication Error

What if MySQL Replication not working?

`Show slave status\G;` gives error

Error

When Slave_SQL_Running: No

10.8.4.1 Solution

On Slave

Just to go sure, we stop the slave:

```
mysql> STOP SLAVE;
```

Fixing the problem is actually quite easy. We tell the slave to simply skip the invalid SQL query:

```
mysql> SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 1;
```

This tells the slave to skip one query (which is the invalid one that caused the replication to stop). If you'd like to skip two queries, you'd use `SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 2;` instead and so on.

That's it already. Now we can start the slave again...

```
mysql> START SLAVE;
```

... and check if replication is working again:

```
mysql> SHOW SLAVE STATUS \G
```

```
mysql> SHOW SLAVE STATUS \G
```

Fig. 28. Solution to SQL Running No

10.8.5 If MySQL Log file is not at correct position

Pass following command on slave,

```
CHANGE MASTER TO MASTER_HOST = 'Master IP', MASTER_USER = 'user_name_here',  
MASTER_PASSWORD = 'password_here', MASTER_LOG_FILE = 'mysql-bin.numberehere',  
MASTER_LOG_POS = number here;
```

10.8.6 If nothing works or Slave IO running=No and SQL running=No.

DO all steps of replication again.

10.9 How to set compression approach in MySQL Master-Slave Replication

If we want to compress the transmission between the master and slave databases, luckily MySQL provides a convenient parameter, which is dynamic and is to be set on the slave: `slave_compressed_protocol=1`

Here is how to implement it:

```
mysql> show global variables like 'slave_compressed_protocol';
```

Output:

```
| Variable_name | Value |
+-----+-----+
| slave_compressed_protocol | OFF |
+-----+-----+

1 row in set (0.00 sec)

mysql> stop slave;

Query OK, 0 rows affected (0.25 sec)

mysql> set global slave_compressed_protocol=1;

Query OK, 0 rows affected (0.00 sec)

mysql> start slave;

Query OK, 0 rows affected (0.00 sec)
mysql> show global variables like 'slave_compressed_protocol';

+-----+-----+
| Variable_name | Value |
+-----+-----+
| slave_compressed_protocol | ON |
+-----+-----+
```

Congratulation! Now, all the traffic that will be transmitted from Master to slave will be in compressed mode.

You can also visit:

<https://www.pythian.com/blog/masterslave-replication-in-limited-bandwidth-scenarios/#comment-9490507>

10.10 Stop Wire-Shark using Shell Script

```
#!/bin/bash

# This program finds PID of tcpdump and kill it. or stops tcpdump

pid=$(ps -e | grep tcpdump)
echo $pid
kill -9 $pid
```

Save above code in file_name.sh and run following command on terminal:

[./file_name.sh](#)

10.11 Start Wire-Shark using Shell Script

```
#!/bin/bash
tmux &
ssh faban@192.168.50.12
tcpdump -w test.pcap &
```

Save above code in file_name.sh and run following command on terminal:

[./file_name.sh](#)

10.12 Start/Stop MySQL Slave using Shell Script

```

#!/bin/bash
#ssh faban@192.168.50.12 scp masterdump.sql 192.168.50.11: &
#ssh 192.168.50.12 scp masterdump.sql 192.168.50.13:
echo namal | sudo -S service mysql start
mysql -u root -pfaban -e "stop slave"
mysql -uroot -pfaban < masterdump.sql
sleep 1
echo namal | sudo -S service mysql restart
#mysql -u root -pfaban -e "start slave"
#####
#ssh faban@192.168.50.12 scp masterdump.sql 192.168.50.11: &
#echo namal | sudo -S service mysql stop
#mysql -u root -pfaban -e "stop slave"

```

10.13 Measure CPU Performance using Shell Script

```

cpu_performance.sh x
1 #!/bin/bash
2
3 #echo 'Enter the name you want to save with:'
4 #read filename
5 #echo $filename
6
7 for ((i=1; i<=3;i++))   ### Outer for loop-Monte Carlo Realizations ###
8 do
9     #echo " ", " ", " ", " " >> /home/faban/a.csv
10    for((j=1; j<=10;j++))   ### Inner for loop-Time of Execution ###
11    do
12        # echo $i
13        pid=$(ps -ef | awk '/mysqld/{print $2}' | head -n 1)
14        ps -p $pid -o %cpu,%mem --noheader >>/home/faban/d.csv
15        echo $j >> a.csv
16        top -b -n 1 | grep Cpu | cut -f 4 -d " " | cut -f 2 -d " " >> /home/faban/b.csv
17        top -b -n 1 | grep "KiB Mem" | cut -f 2 -d ":" | cut -f 1,3 -d " " | sed 's/[^0-9,]*/g' >> /home/faban/c.csv
18        paste -d , a.csv b.csv c.csv d.csv > TopOutput.csv
19        sleep 1
20    done
21    echo "" >> /home/faban/d.csv
22    echo "" >> /home/faban/a.csv
23    echo "" >> /home/faban/b.csv
24    echo "" >> /home/faban/c.csv
25    echo "" >> /home/faban/TopOutput.csv
26 done
27 #rm /home/faban/a.csv
28 #rm /home/faban/b.csv
29 #rm /home/faban/c.csv
30 #rm /home/faban/d.csv

```

10.14 Completely Remove MySQL

<http://stackoverflow.com/questions/25244606/completely-remove-mysql-ubuntu-14-04-lts>

<http://stackoverflow.com/questions/10853004/removing-mysql-5-5-completely>

After removing completely.

sudo apt-get update

sudo apt-get install -y mysql-server

10.15 C-Code for MySQL to send data to remote host

```
10k_Quries.c x
1 #include <my_global.h>
2 #include <mysql.h>
3 #include <time.h>
4
5 void finish_with_error(MYSQL *con)
6 {
7     fprintf(stderr, "%s\n", mysql_error(con));
8     mysql_close(con);
9     exit(1);
10 }
11
12 int main(int argc, char **argv)
13 {
14
15     MYSQL *con = mysql_init(NULL);
16
17     if (con == NULL)
18     {
19         fprintf(stderr, "mysql_init() failed\n");
20         exit(1);
21     }
22
23     if (mysql_real_connect(con, "192.168.40.12", "root", "faban",
24         "experiments", 0, NULL, 0) == NULL)
25     {
26         finish_with_error(con);
27     }
28
29
30
31     mysql_close(con);
32     exit(0);
33 }
```

How to run C-Code?

Do following step on Client's terminal

```
mysql_config - - cflags
```

```
mysql_config -libs
```

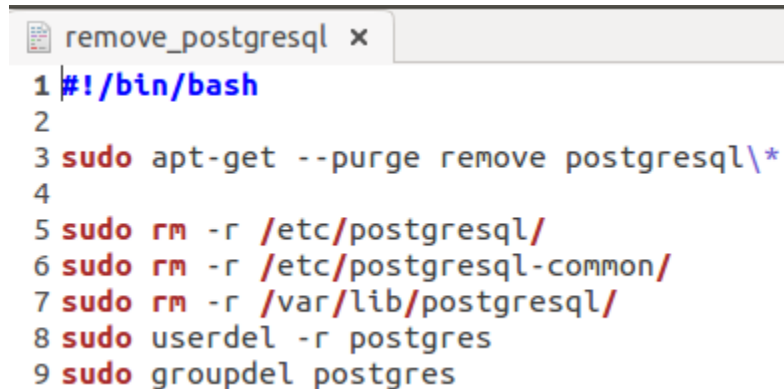
```
gcc -o output_file_name_here $(mysql_config - - cflags) input_file_name_here $(mysql_config -  
- libs)
```

Now,

```
./output_file_name_here
```

As a result, you will be connected to remote MySQL server to run queries.

10.16 Completely Remove PostgreSQL using Shell Script



```
1 #!/bin/bash  
2  
3 sudo apt-get --purge remove postgresql\*  
4  
5 sudo rm -r /etc/postgresql/  
6 sudo rm -r /etc/postgresql-common/  
7 sudo rm -r /var/lib/postgresql/  
8 sudo userdel -r postgres  
9 sudo groupdel postgres
```

Install PostgreSQL using Shell Script.

```
1 #!/bin/bash  
2  
3 sudo apt-get install postgresql-9.3
```

10.17 How to Send Images to remote PostgreSQL server using C-Code

pgbin.c x

```
1 #include <libpq-fe.h>
2 #include <stdlib.h>
3 #include <arpa/inet.h>
4
5 static void
6 exit_nicely(PGconn *conn)
7 {
8     PQfinish(conn);
9     exit(1);
10 }
11
12 PGresult* put_data_to_Images(
13     PGconn* conn,
14     int id,
15     int data_size,
16     const char* const data
17 ) {
18     PGresult* result;
19     const unsigned int id_big_endian = htonl((unsigned int)id);
20     const char* const paramValues[] = { &id_big_endian, data };
21     const int nParams = sizeof(paramValues) / sizeof(paramValues[0]);
22     const int paramLengths[] = { sizeof(id_big_endian), data_size };
23     const int paramFormats[] = { 1, 1 }; /* binary */
24     const int resultFormat = 0; /* text */
25
26     result = PQexecParams(
27         conn,
28         "insert into Images (id, data) values ($1::integer, $2::bytea)",
29         nParams,
30         NULL, /* Types of parameters, unused as casts will define types */
31         paramValues,
32         paramLengths,
33         paramFormats,
34         resultFormat
35     );
36     return result;
```

```

pgbin.c x
37 }
38
39 int main(int argc, char** argv)
40 {
41     char conninfo[30];
42     /* if(argc < 3)
43     {
44         printf("Usage: ./pgbin 192.168.50.12 postgres im1.png\n");
45         return -1;
46     }*/
47     sprintf(conninfo, "dbname=postgres hostaddr=192.168.50.12 user=postgres password=faban sslmode=disable"); //, argv[2], argv[1]);
48     PGconn* conn = PQconnectdb(conninfo);
49     char *bytes;
50     if (PQstatus(conn) != CONNECTION_OK)
51     {
52         printf("Connection to database failed: %s",
53             PQerrorMessage(conn));
54         exit_nicely(conn);
55     }
56     // FILE* fptr = fopen(argv[3], "rb");
57     FILE* fptr = fopen("im1.png", "rb");
58     if(!fptr)
59     {
60         printf("Couldn't open image file %s.\n", argv[3]);
61         return -2;
62     }
63     //Get file length
64     fseek(fptr, 0, SEEK_END);
65     int fileLen=ftell(fptr);
66     fseek(fptr, 0, SEEK_SET);
67
68     //Allocate memory
69     bytes=(char *)malloc(fileLen+10);
70     if (!bytes)
71     {
72         printf("Memory error!\n");

```

```

72         printf("Memory error!\n");
73         fclose(fptr);
74         return -3;
75     }
76
77     //Read file contents into buffer
78     fread(bytes, fileLen, 1, fptr);
79     fclose(fptr);
80
81
82     PGresult* res = put_data_to_Images(conn, 0, fileLen, bytes);
83     PQclear(res);
84     PQfinish(conn);
85     free(bytes);
86     return 0;
87 }

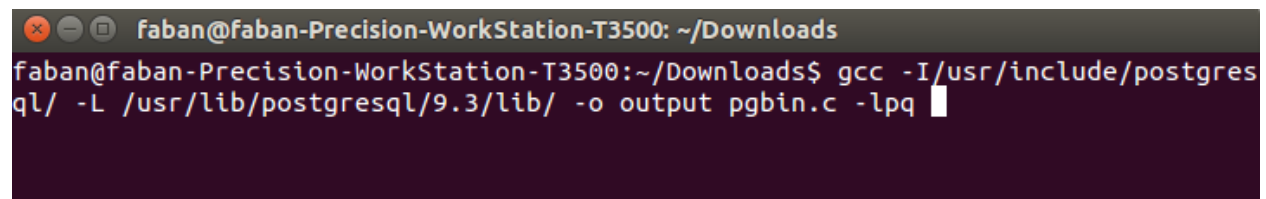
```

The above code will insert the image to a remote PostgreSQL server. In order to run above code, copy above code and save in a file like “insert-image.c” and run the following commands on terminal:

```
gcc -I/usr/include/postgresql/ -L /usr/lib/postgresql/9.3/lib/ -o output_file_name_here
input_file_name_here -lpq
```

```
./output_file_name_here
```

10.17.1 Example



```

faban@faban-Precision-WorkStation-T3500: ~/Downloads
faban@faban-Precision-WorkStation-T3500:~/Downloads$ gcc -I/usr/include/postgres
ql/ -L /usr/lib/postgresql/9.3/lib/ -o output pgbin.c -lpq

```

Fig. 29. How to run C Code

After that, type:

```
./output
```

10.18 Solution to Encrypted Data in Wire-shark

I was trying to monitor all traffic related to PostgreSQL between Master and Slave. To that end, I used Wire-shark to monitor the traffic. Then, I started PostgreSQL and ran three queries (Create table Hello, Create table Bye & inserted an image to PostgreSQL database). During queries, I ran Wire-shark on Master just to capture the traffic between Master and Slave.

But there was one problem with PostgreSQL traffic captured using Wireshark, all the traffic was sent/received in TCP packets and that traffic was in encrypted form. That means I was unable to read the data. I wanted to find out all those three queries from Wire-shark that I inserted in PostgreSQL database. What is the best way to go about finding queries of PostgreSQL?

On the other hand, I ran same queries on MySQL database and repeated above mentioned experiment. I can easily read all those three queries in Wire-shark dump because they are not in encrypted form.

10.18.1 Solution

I used Python code to insert queries into remote PostgreSQL database. I used following line in PostgreSQL to connect with database.

```
con = psycopg2.connect(host="192.168.50.12", database="postgres", user="postgres", password="faban")
```

If you use above approach then all the data will be sent in encrypted form. If you use the approach given below in python code then all the data will be sent in decrypted form. You can easily read all queries in Wire-shark.

```
con = psycopg2.connect("host=192.168.50.12 dbname=postgres user=postgres password=faban sslmode=disable")
```

Same is the case in C-Code as well.

10.18.2 Decrypted data

```
sprintf(conninfo, "dbname=postgres hostaddr=192.168.50.12 user=postgres password=faban sslmode=disable");
```

10.18.3 Encrypted Data

```
sprintf(conninfo, "dbname=postgres hostaddr=192.168.50.12 user=postgres password=faban");
```


10.19 Conversion of Image to Hex Values

run this in terminal:

```
xxd image_name_here
```

In order to find specific number in Hex values:

```
xxd Image_name_here | grep 'Number you wanna search'
```

10.20 How to Find difference between two csv files

There are four methods.

```
Sudo apt-get install diffuse
```

```
diffuse file_1_here file_2_here
```

or

```
sudo apt-get install meld
```

```
meld file_1_here file_2_here
```

or

```
diff file_1_here file_2_here
```

or

```
sudo apt-get install vim
```

```
vim file_1_here file_2_here
```

10.21 How to Convert Hex File to Text Format

```
xxd -r file_name_here
```

10.22 Merge two rows into one row

```
awk 'BEGIN {RS=" "; FS="\n";OFS=",";} {print $1,$2;}'
```

Output:

Hello, Good Bye

If you don't put comma between \$1 and \$2 then rows will merge together

Output:

HelloGood Bye

10.23 Use of Wire-Shark

Filters to search Traffic

MySQL Traffic

```
ip.src==192.168.50.11 && ip.dst==192.168.50.12 && tcp.port==3306  
ip.addr==192.168.50.12
```

PostgreSQL Traffic

```
ip.src==192.168.50.11 && ip.dst==192.168.50.12 && tcp.port==5432
```

Search a string in Wireshark file

```
frame contains "string_here"
```

How to convert a captured file into csv file

Open File-> apply filter-> File-> export as-> csv

10.24 Terminal Commands

Restart P.C

```
sudo reboot
```

or

```
sudo shutdown -r now
```

Shutdown

```
sudo shutdown -h
```

```
clear
```

Simply put, this clears the current window. (Within Terminal in OS X, you can still scroll up to see what was there. This command simply clears the current view).

Learn more on:

<https://computers.tutsplus.com/tutorials/40-terminal-tips-and-tricks-you-never-thought-you-needed--mac-51192>

10.25 MySQL Commands

Login

```
sudo mysql -uroot -p
```

create database name_here;

show databases;

use database_name_here;

drop database name_here;

create table in a database for example, exp.

Use exp;

create table images (id int, data MEDIUMBLOB);

select id from images;

*select * from images;*

INSERT INTO table_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

truncate table images;

drop table images;

DROP USER 'repl'@'%';

10.26 PostgreSQL Commands

Login

su - postgres

create database name_here;

show databases;

use database_name_here;

drop database name_here;

create table in a database for example, exp.

Use exp;

create table images (id int, data MEDIUMBLOB);

select id from images;

*select * from images;*

INSERT INTO table_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

delete from images;

drop table images;

10.27 Resizing Virtual Machines

The following link points to a page that, I think, it shows the right way to resize the VM's disk. It uses a GParted Live CD so that the VM's disk partitions can be resized without booting the VM.

<https://www.stevejenkins.com/blog/2014/07/resize-a-fedora-or-windows-dynamic-guest-virtual-disk-vdi-in-virtualbox-in-3-easy-steps/>

OR

Follow the procedure given below:

10.27.1 Assumptions

First, we're assuming that you've backed up your data, if it's important. Remember, there are two types of data: data that's backed up, and data that's waiting to be lost. Backup your .vdi file before going any further.

We're assuming that your Guest OS is LUBUNTU. For any other setup, these instructions will probably still be helpful, but this is the Guest systems that this specifically applies to.

We're assuming that you set up your VDI with Dynamically Allocated Storage. This will not work with fixed drives.

We're assuming that if you hose this up royally, you don't know where I live... and that even if you do hose it up royally, you've followed Assumption #1. Seriously. Back it up, yo!

All the steps to resize VMS are given in [/RA Work by Karam & Jam/Faban/Documents/Lubutu VMs Size Increase Guide](#)

10.28 Assign Static IP on Linux

Click on Edit
Connections



Fig. 30. Static IP Step 1

Click on Edit

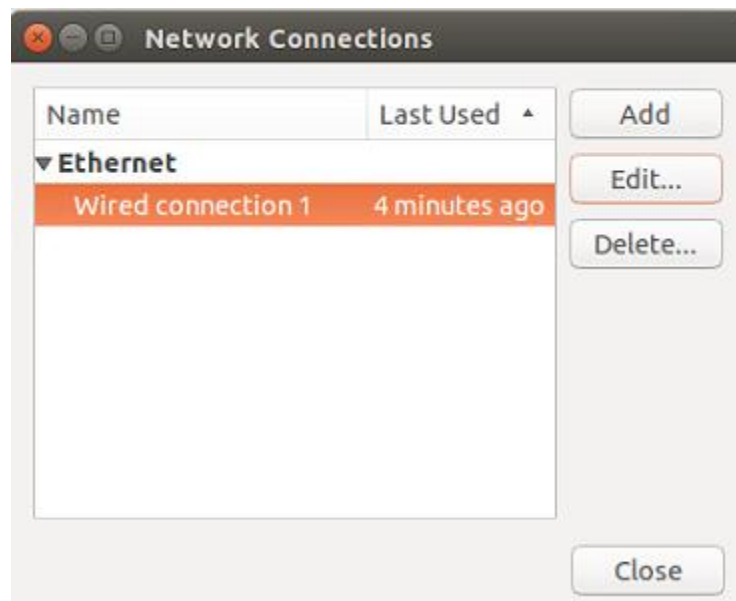


Fig. 31. Static IP Step 2

Click on IPV4 settings, select Manual from Method list and fill the required fields shown below.

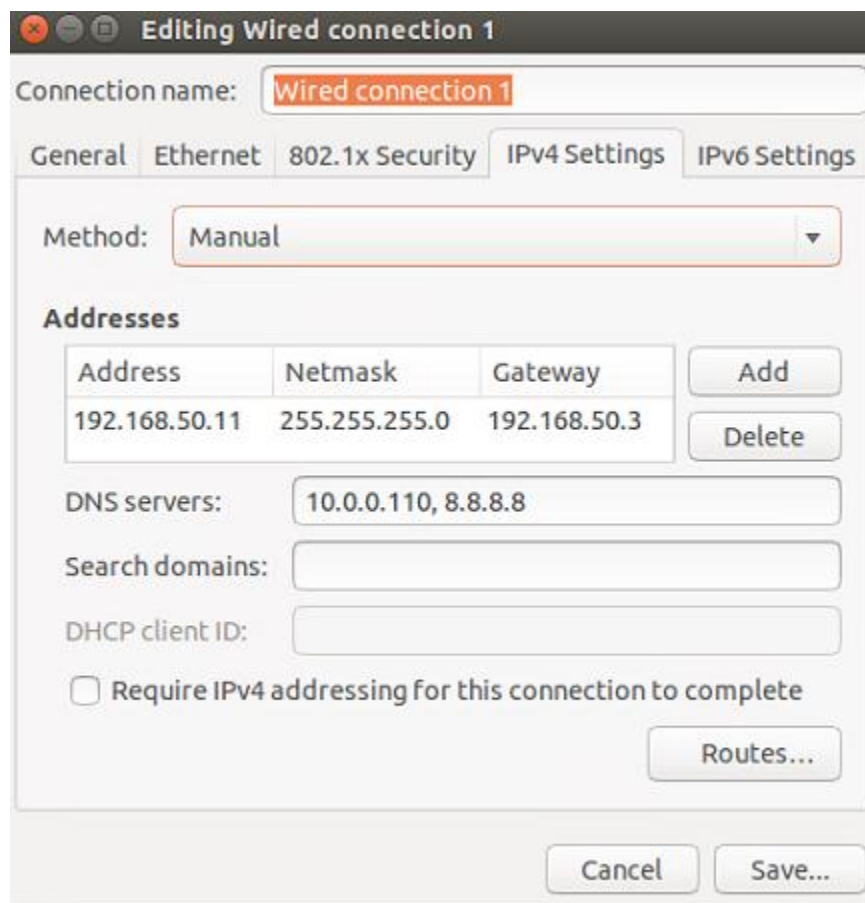


Fig. 32. Static IP Step 3

For example, in this case, my static IP is 192.168.50.11.

10.29 Installation of Dukto on Linux

A program to transfer all kind of files to and texts to several devices, regardless of their OS.

Instructions for 32 bit systems:

```
$ sudo apt-get install gdebi  
$ wget download.opensuse.org/repositories/home:/colomboem/xUbuntu_12.04/i386/dukto_6.0-1_i386.deb  
$ sudo gdebi dukto_6.0-1_i386.deb
```

Instructions for 64 bit systems:

```
$ sudo apt-get install gdebi  
$ wget download.opensuse.org/repositories/home:/colomboem/xUbuntu_12.04/amd64/dukto_6.0-1_amd64.deb  
$ sudo gdebi dukto_6.0-1_amd64.deb
```

Fig. 33. Dukto Installation Commands

10.30 Installation of MongoDB on Ubuntu 14.04

Installation:

<https://www.liquidweb.com/kb/how-to-install-mongodb-on-ubuntu-14-04/>

Replication:

<https://blog.ajduke.in/2013/05/31/setup-mongodb-replica-set-in-4-steps/>

<http://stackoverflow.com/questions/15124610/multiple-instances-of-mongo-db-on-same-server>

<http://www.alphadevx.com/a/491-Running-two-MongoDB-instances-on-one-server>

<http://www.thegeekstuff.com/2014/02/mongodb-replication/>

<https://www.youtube.com/watch?v=vrpum7kpXzo>

10.31 Compression Approach in Cassandra DB

Compression is enabled by default in Cassandra. In order to disable or change compression approach:

http://docs.datastax.com/en/cassandra/2.1/cassandra/operations/ops_config_compress_t.html

11 Note

If you face any problem while installation or during experiments then most of the problems and their solutions are written above.

For queries please feel free to contact:

Muhammad Karam Shehzad

Karam.shehzad@hotmail.com

(+923155758422)

Jam Muhammad Yousif

jammuhammadyusif@gmail.com

(+923046421856)