Commands to be used for replication in multiple machines

Every system has

1. **Local host** which recognizes the system as "This system" with a specific name of that system.

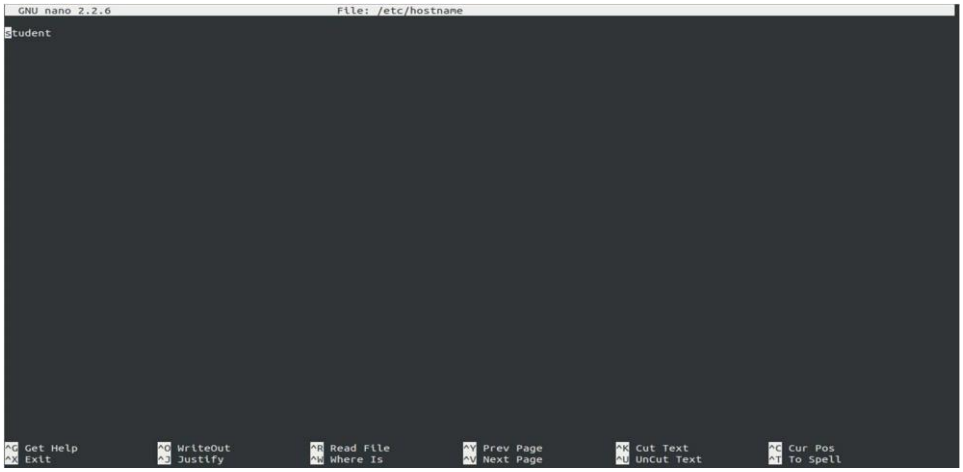
   Example "student@student" (<username>@<systemname or local-host>) shows that name of the local host or that specific computer is "student".

   Process

   - ✓ The local host name or the system name on the ubuntu can be found using the command

     nano /etc/hostname

   - ✓ The above command specifies that the "hostname" is the file present in the directory "etc".
   - ✓ "nano" text editor is used to open the file in the terminal itself where it can be edited, it can also be opened through the command "gedit /etc/hostname" where "gedit" is the editor same as notepad in windows.
   - ✓ The output of the above command when executed on the terminal will be as follows:



   ⇒ The above terminal shows that the name of that particular system is "student"

   ⇒ To come out of the prompt press "**ctrl x**".

The name of the system is set during the installation of the OS (ubuntu) on that particular computer.

- ✓ By default, the file "hostname" is write protected and permissions on that file has to be changed to make any changes.

  ⇒ The permissions for the file can be changed using the following command.

  sudo chmod -R 777 /etc/hostname

- ✓ To change the host name

⇒ Set the permissions as shown above and open the file hostname in editor mode by executing the following code

nano /etc/hostname

⇒ The prompt appears with the host name, replace the existing name with the new name.

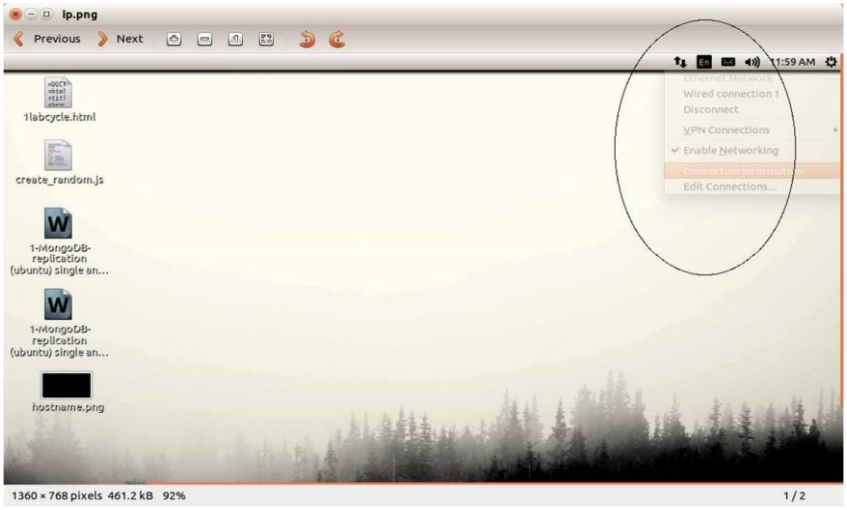
⇒ Save the changes by pressing "**ctrl x**" which asks for the permission to change, press **Y** (for yes) and then press enter.

The hostname is used for the communicating purposes between different systems, if different systems have different names then there is no need to change the host name.

2. **IP address** that is used for communication on different systems over the internet (every system has unique IP addresses which may be static or dynamic).

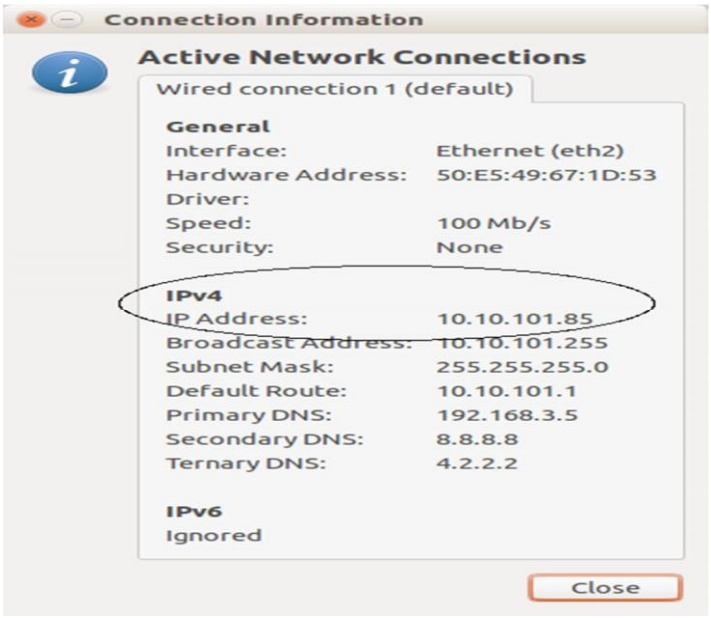
Process

✓ The IP address of the system can be found using the connection information present on the tool bar.



✓ After selecting connection information, the below prompt will appear.

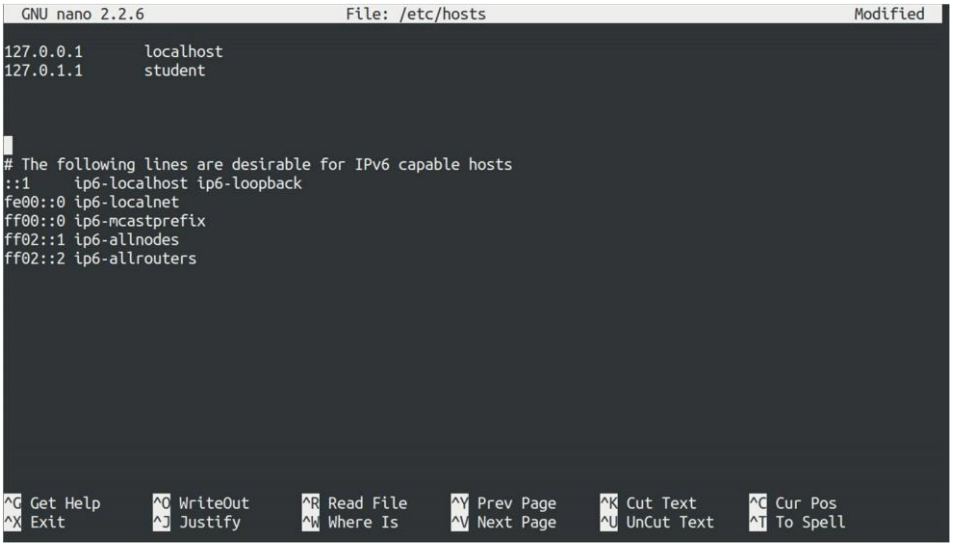
✓ The highlighted area specifies the IP address of the system.



- ✓ When there is a need for connection between different systems it is better to specify the IP addresses of different machines with their system names, by which it will not be required to remember the IP addresses of all the system when communication has to be established.
- ✓ All the information about the different host machines has to be specified in the file called "hosts" which is present in the "etc" directory.
- ✓ By default, the "hosts" file will contain only the address of the local host and name of the local host machine, this file can be viewed in the terminal by executing the following command.

nano /etc/hosts

- ✓ The output after executing the command will be as follows



- ✓ By default, the hosts file is write protected, to change the permission to make it editable execute the following command on the terminal.

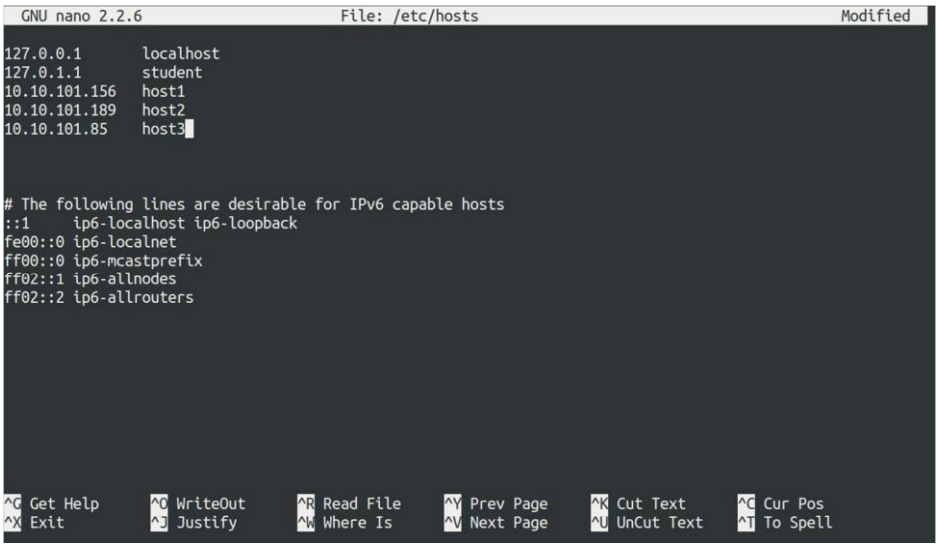
⇒ The permissions for the file can be changed using the following command.

sudo chmod -R 777 /etc/hosts

- ✓ To add the hosts to any particular system
  - ⇒ Note down the IP addresses and their respective names of the systems which has to be added
  - ⇒ Write the same information in the hosts file after changing the permissions of the hosts file as shown in the previous command.
- ✓ After changing the permissions of hosts file open the file in the editor mode by executing the command shown below:

nano /etc/hosts

- ✓ The IP addresses and their respective names is written in the hosts file
- ✓ To save the file
  - ⇒ press "ctrl x"
  - ⇒ press y (for yes)
  - ⇒ press enter



- ✓ Check the new hosts and hostname files by executing following commands

nano /etc/hosts
nano /etc/hostname

- ✓ After changing the name both in hostname and hosts file the new name will be shown on the terminal.

For configuring the multiple systems for replication follow the below procedure by starting with changing the directory permissions as follows.

Process

1. Set permission to etc directory

sudo chmod -R 777 /etc/hosts
sudo chmod -R 777 /etc/hostname

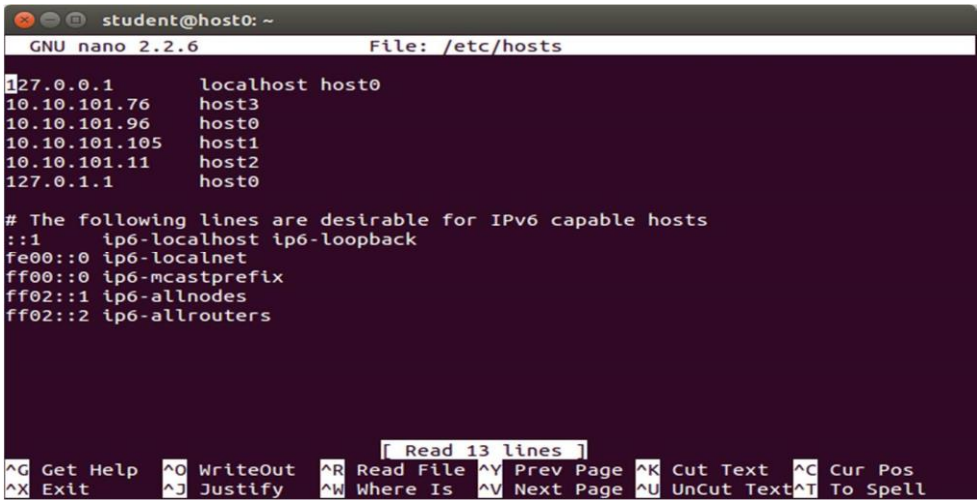
2. Change the hostname in 3 machines following the given procedure

nano /etc/hostname

3. Add all the hosts name in each machine

nano /etc/hosts

✓ etc/hosts of host0 will look like the following window:



Note: The new name of the machine has to be written after the string '**localhost**'.
Do not change the address of the string '**localhost**'
Remove the address of the old hostname.
Add the IP address taken and add the new hostname.

4. Set permissions to the directory in which data directories for the replication set is being created in **each machine**.

Note: This may not work as it may give the error as host name unreachable.
Restart the machine before continuing.

sudo chmod -R 777 /var

5. Create data directory in **each system**. These directories are created to store the operational logs, database files and any other information necessary for the replica set to perform efficiently.
6. As the directories are being created in **different systems**, the name can either be same for all the systems or unique names can be given for the **respective systems**.
This command is to be written in each system individually.

mkdir -p /var/lib/mongodb/host0

```
mkdir -p /var/lib/mongodb/host1
mkdir -p /var/lib/mongodb/host2
```

Note: This will create directory called "host0", "host1", "host2" in three different systems which will contain the database files.

7. Start mongod server by creating an mongod instance executing the code on each system.
   Note - Make sure that no other mongod instance is running.
   If any of the instances is running then shutdown the server then restart the server with new configuration for the replica set as shown below.

Terminal in host0

```
mongod --port 27040 --dbpath /var/lib/mongodb/host0 --replSet test
                    --smallfiles --oplogSize 128
```

Terminal in host1

```
mongod --port 27041 --dbpath /var/lib/mongodb/host1 --replSet test
                    --smallfiles --oplogSize 128
```

Terminal in host2/

```
mongod --port 27042 --dbpath /var/lib/mongodb/host2 --replSet test
                    --smallfiles --oplogSize 128
```

Note:

- ⇒ The --smallfiles and --oplogSize settings reduce the disk space that each mongod instance uses. This is ideal for testing and development deployments as it prevents overloading your machine.
- ⇒ While starting the server the port number specified by --port can be same for all the systems or it can be different as per convenience.
- ⇒ Make sure the database path specified with --dbpath is same as that of the data directories created in the previous step.

(test is name of replica set in above commands in step-7)

8. Any one system (e.g. host0:27040) which will behave as primary, execute following commands to make primary and to add 2 secondary nodes.

Terminal 2 in host0

```
mongo --port 27020
rs.initiate()
rs.add("host1:27041")
rs.add("host2:27042")
```

- ⇒ Use rs.initiate() in only one of the system which is preferred to be primary (until rs.initiate() is not executed in any one of the systems none of the systems will be primary).

⇒ Note If rs.initiate() is used on multiple systems then those systems will not become part of replica set.

9. Execute following commands at remaining 2 systems which are secondary

Terminal 2 in host1

mongo --port 27021
db.getMongo().setSlaveOk()

Terminal 2 in host2

mongo --port 27022
db.getMongo().setSlaveOk()

⇒ Now test replica set is ready, primary at 27040 port and 2 secondary at 27041 and 27042

⇒ As the replication method in mongodb follows master-slave configuration it is necessary to execute "db.getMongo().setSlaveOk()" on the systems which acts as secondary without which the data cannot be accessed on the secondary systems.

10. Create the database, collection and document in the primary member and view the same in the secondary members.

⇒ **Note** Make sure that every terminal is open which was opened from the start of the process, closing of any one terminal will result in the loss of the connection between the members of the replication set and the operations on the members will not be executed.

1. To set the priority of any of the member of the replication set test to "0"
   example: member="_id:2"
   cfg=rs.conf()
   cfg.members[2].priority=0
   rs.reconfig[cfg]

2. To set the hidden property of any of the member of the replication set to "true"
   example: member="_id:2"
   cfg=rs.conf()
   cfg.members[2].priority=0
   cfg.members[2].hidden=true
   rs.reconfig[cfg]

3. To set the slaveDelay property of any of the member of the replication set to a specific time duration.

example: member="_id:2", time=60seconds

cfg=rs.conf()

cfg.members[2].priority=0

cfg.members[2].hidden=true

cfg.members[2].slaveDelay=60

rs.reconfig[cfg]

Common errors

⇒ If permissions are not set to 777 (read, write, execute) for the hosts file and hostname file, it cannot be edited.

⇒ If permissions are not set to 777 (read, write, execute) for any of the directories (var,srv,etc..) then the new data directory cannot be created.

⇒ While inserting the IP address and the host name of the other systems in the hostname file make sure every IP address is assigned to their respective system names without which the connections cannot be established.

⇒ Make sure that while creating the server with mongod instance it has to use the unused port number and has the specific db path in which the directories are created.

⇒ Mongod and mongo are two different entities.

- ✓ mongod is used to start the server.
- ✓ mongo is used for creating an instance to perform different operations on database.
  - χ Without executing mongod first, the mongo instance cannot not be connected.

⇒ If there is any problem in starting the mongod server it maybe because of the following reasons:

- ✓ Server is already running with that port number.(stop the server)
- ✓ The port number being used is assigned to any other replicaset.(use different port number)
- ✓ The permissions may not be set for the data directories (set the permissions to 777)
- ✓ The db path specified by --dbpath may be different from that of the data directories created for that particular member.

⇒ If the connection between the systems is not being established then:

- ✓ Check for the internet connection.
- ✓ Restart all the servers.

⇒ If the database or the collections is not being shown on the instances of secondary member then set the secondary members to db.getMongo().setSlaveOk() .

Extra Information

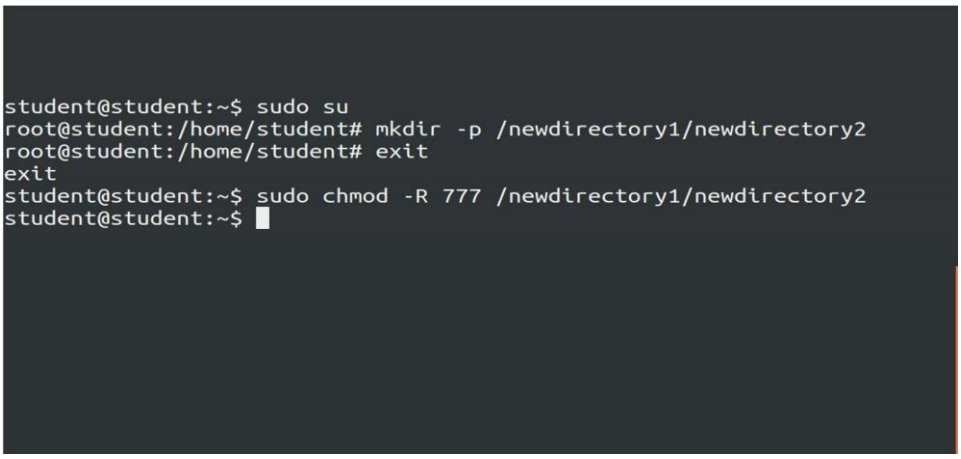
⇒ Data directories for the replication set can be created in any of the directories such as var, srv, usr, data, etc...

⇒ If there is a need for a new directory instead of using the existing directories, then the new directories can be created by executing the following code on the terminal:

sudo su

⇒ The above command gives the permission to create the directory/folder in the root directory as the super user.

⇒ The following snap shot shows the transition from home directory to root directory and creation of new directories by setting the permissions.



Note: var is the folder in which data directories are created.