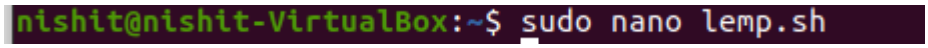
Name: Nishit Rajani

Day - 4

Task 1 - Learn MongoDB

◆ Install Lemp(Use Mongo DB instead of Mysql)

- Create shell script file using below command:
  sudo nano lemp.sh



- Write script command as below mentioned:

```
#!/bin/bash

#Retrieve new lists of packages bellow this command
sudo apt-get update

#install nginx bellow this command
apt install nginx

#Show runtime status of nginx bellow this command
systemctl status nginx.service

#Import the public key used by the package management system.
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -

#create a list file for MongoDb
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0
multiverse" | sudo tee /etc/apt/sources.list.d/mong>

#Reload local pakaage database
sudo apt-get update

#install nginx bellow this command
sudo apt-get install -y mongodb-org

#start mongodb
sudo systemctl start mongod

#Check mongodb status bellow this command
sudo systemctl status mongod

#install software-properties-common
sudo apt install software-properties-common

#add the ondrej/php PPA which provides different PHP versions for Ubuntu
sudo add-apt-repository ppa:ondrej/php

#install php version 7.4
sudo apt -y install php7.4

#show php version
php -v
echo "Install Lemp(Use Mongo DB instead of Mysql) successfully installed"
```

```
#!/bin/bash

#Retrieve new lists of packages bellow this command
sudo apt-get update

#install nginx bellow this command
apt install nginx

#Show runtime status of nginx bellow this command
systemctl status nginx.service

#Import the public key used by the package management system.
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -

#create a list file for MongoDb
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt>

#Reload local pakaage database
sudo apt-get update

#install nginx bellow this command
sudo apt-get install -y mongodb-org

#start mongodb
sudo systemctl start mongod

#Check mongodb status bellow this command
```

[Read 54 lines]

```
#install software-properties-common
sudo apt install software-properties-common

#add the ondrej/php PPA which provides different PHP versions for Ubuntu
sudo add-apt-repository ppa:ondrej/php

#install php version 7.4
sudo apt -y install php7.4

#show php version
php -v
```

- Give execute permission to script file.
  sudo chmod +x lemp.sh

```
nishit@nishit-VirtualBox:~$ sudo chmod +x lemp.sh
nishit@nishit-VirtualBox:~$ ls -l
```

```
-rwxrwxr-x 1 nishit nishit 1393 Feb 17 09:56 lemp.sh
```

- Run scrpit using below command:
  bash lemp.sh

```
nishit@nishit-VirtualBox:~$ bash lemp.sh
Ign:1 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:4 http://repo.mongodb.com/apt/ubuntu focal/mongodb-enterprise/6.0 InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:7 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 Release
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:9 http://repo.mongodb.com/apt/ubuntu focal/mongodb-enterprise/6.0 Release
Get:12 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist InRelease [731 kB]
Err:5 http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist InRelease
  Clearsigned file isn't valid, got 'NOSPLIT' (does the network require authentication?)
Reading package lists... Done
E: Failed to fetch http://downloads-distro.mongodb.org/repo/ubuntu-upstart/dists/dist/InRelease  Clearsigned file i
alid, got 'NOSPLIT' (does the network require authentication?)
```

- To enter in mongodb use below command.
  mongosh

```
nishit@nishit-VirtualBox:~$ mongosh
Current Mongosh Log ID: 63ef18e3b302e636f3c2820d
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1
Using MongoDB:          5.0.14
Using Mongosh:          1.7.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-02-17T11:23:06.223+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2023-02-17T11:23:07.004+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

test>
```

◆ Learn MongoDB Basic commands

1) **show dbs;** ==> It is used to view all databases.

```
test> show dbs;
admin   40.00 KiB
config  12.00 KiB
local   40.00 KiB
```

2) **use local;** ==> It is used to select database to work with.

```
test> use local;
switched to db local
local>
```

3) **show collections** ==> Using this we able to see collections (tables) in db.

```
local> show collections
startup_log
local>
```

4) **use Nishit** ==> To create new db.

```
local> use Nishit
switched to db Nishit
Nishit>
```

5) **db** ==> Check we are in which database.

```
Nishit> db
Nishit
```

6) **db.createCollection('Age')** ==> Here, create one collection(table) in Nishit db.

```
Nishit> db.createCollection('Age')
{ ok: 1 }
Nishit> show collections
Age
```

7) **db.collectionName.insert({“field” : “value”})** ==> This command creates field and value in collection.

db.Age.insert({“Name” : “20”})

```
Nishit> db.Age.insert({"Name" : "20"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63ef23f80a073416a5e6be63") }
}
```

8) **db.collectionName.find()** ==> It is used to print our field and value.

db.Age.find()

```
Nishit> db.Age.find()
[ { _id: ObjectId("63ef23f80a073416a5e6be63"), Name: '20' } ]
```

9) **db.dropDatabase()** ==> Used to delete database.

```
Nishit> db.dropDatabase()
{ ok: 1, dropped: 'Nishit' }
Nishit> show dbs
admin    40.00 KiB
config  108.00 KiB
local    40.00 KiB
```

10) **db.collectionName.drop()** ==> To delete collection

db.Age.drop()

```
Nishit> db.Age.drop()
true
```

- Now check the available collection in Nishit db.

db.Age.find()

```
Nishit> db.Age.find()

Nishit>
```

11) Create user

db.createUser({

user:"demo",

pwd:"demo123",

roles:[“readwrite

});

```
Nishit> db.createUser({
... user:"demo",
... pwd:"demo123",
... roles:["readWrite"]
... });
{ ok: 1 }
Nishit>
```

12) Delete User

db.dropUser(“demo”)

```
Nishit> db.dropUser("demo")
{ ok: 1 }
Nishit>
```

◆ Mongo Express installation and usage.

◆ Make Script for Mongo DB Password.

- First need to move to admin data base. Because other data base cannot create user.
- After that create user using following command:

  Use admin

  db.createUser({user: “demo”, pwd: “demo123”, roles: [“readWrite”] });

```
test> use admin
switched to db admin
admin> db.createUser({ user: "demo", pwd: "demo123", roles: ["readWrite"] });
{ ok: 1 }
```

- Login to user using below command:

  sudo mongosh -u demo -p

Here we simply run command sudo for root access, mongosh that we use database, -u for user and that user name and than -p for password.

```
nishit@nishit-VirtualBox:~$ sudo mongosh -u demo -p
Enter password: *******
Current Mongosh Log ID: 63ef498d6cd88c35880d06f6
Connecting to:          mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1
Using MongoDB:          5.0.14
Using Mongosh:          1.7.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2023-02-17T14:38:30.518+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2023-02-17T14:38:40.127+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

test>
```

Task 2 - Nginx Configuration.

- **How to point /fetch prefix to a specific directory.**

➢ Step1: Create two directories under *var*/www/html.
Command: **cd *var*/www/html**
mkdir site1 site2

```
root@nishit-VirtualBox:~# cd /var/www/html/
root@nishit-VirtualBox:/var/www/html# mkdir site1 site2
```

➢ Step 2: Create file in site1 directory and add some content in file.
Command: **cd site1/**
nano index.html

File content: **Hello WOrld site1**

```
root@nishit-VirtualBox:/var/www/html# cd site1/
root@nishit-VirtualBox:/var/www/html/site1# nano index.html
```

```
  GNU nano 4.8                                    index.html
Hello WOrld site1
```

- Create file in site2 directory and add some content in file.

Commnad: **cd ../site2**

nano index.com

File content: **Hello WOrld site2**

```
root@nishit-VirtualBox:/var/www/html/site1# cd ../site2
root@nishit-VirtualBox:/var/www/html/site2# nano index.html
```

```
  GNU nano 4.8                                          index.html
Hello WOrld site2
```

➢ Step 3: Change ownership of /var/www/html/site1 & /var/www/html/site2/.

Command: chown -R $USER:$USER /var/www/html/site1/

chown -R $USER:$USER /var/www/html/site2/

So, meaning of above command is changes the ownership of the /var/www/html/site1/ directory and all its contents to the currently logged-in user and the group that the user belongs to.

```
root@nishit-VirtualBox:~# chown -R $USER:$USER /var/www/html/site2/
root@nishit-VirtualBox:~# chown -R $USER:$USER /var/www/html/site1/
root@nishit-VirtualBox:~#
```

➢ Step 4: Using the sites-available default file create another copy file.

cp /etc/nginx/sites-available/default /etc/nginx/sites-available/demo

```
root@nishit-VirtualBox:~# cp /etc/nginx/sites-available/default /etc/nginx/sites-available/demo
```

➢ Step 5: Make the changes in newly created file to open two website in one virtual host.

server {

listen 80;

root /var/www/html/site1;

index index.html;

server_name www.site1.com site1.co;

location / {

try_files $uri $uri/ =404;

}

}

server {

listen 80;

root /var/www/html/site2;

index index.html;

server_name www.site2.com site2.co;

location / {

try_files $uri $uri/ =404;

}

```
        root /var/www/html/site1;
        index index.html;
        server_name www.site1.com       site1.co;
        location / {
        try_files $uri $uri/ =404;
    }
}

        server {
        listen 80;
        root /var/www/html/site2;
        index index.html;
        server_name www.site2.com       site2.co;
        location / {
        try_files $uri $uri/ =404;
    }
```

- Step 6: Now remove sites-enabled default file.

rm /etc/nginx/sites-enabled/default

```
root@nishit-VirtualBox:~# rm /etc/nginx/sites-enabled/default
```

- Step 7: Enabling Server Blocks

We must enable our server block files now that we have them. Symbolic links from these files to the sites-enabled directory, which Nginx reads from at startup, can be used to do this.

ln -s /etc/nginx/sites-available/demo /etc/nginx/sites-enabled/

```
root@nishit-VirtualBox:~# ln -s /etc/nginx/sites-available/demo /etc/nginx/sites-enabled/
```

- Step 8: Next, check all of your Nginx files to ensure there are no syntax errors:

  nginx -t

```
root@nishit-VirtualBox:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

- Step 9: In order to activate your modifications, restart Nginx if no issues were discovered:

  systemctl restart nginx

```
root@nishit-VirtualBox:~# systemctl restart nginx.service
```

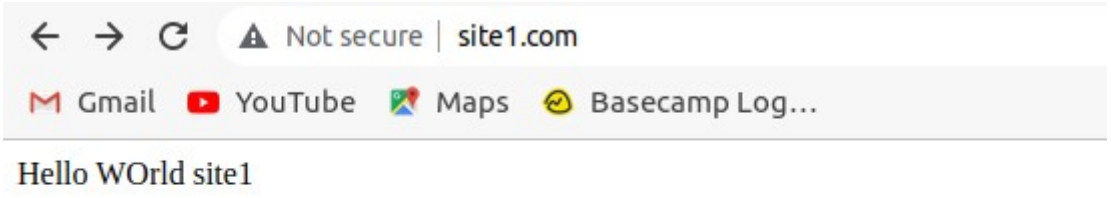
- Step 10: Modify etc/hosts file.

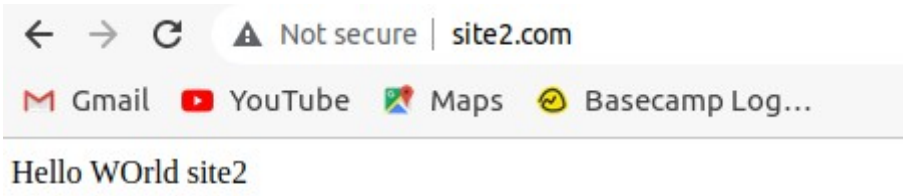
  nano etc/hosts

  Here add system ip and domain name.

```
192.168.10.133  www.site1.com   site1
192.168.10.133  www.site2.com   site2
```

- Step 11: Check the result.





- **Point a domain www.example.com to the Nginx Configuration.**

Step 1: Create one directories under *var*/www/html.

cd /*var*/www/html

mkdir example

```
root@nishit-VirtualBox:~# cd /var/www/html/
root@nishit-VirtualBox:/var/www/html# mkdir example
```

Step 2: Create html file in example directory.

nano index.html

```
root@nishit-VirtualBox:/var/www/html# nano index.html
```

Step 3: Change ownership of /var/www/html/example.

chown -R $USER:$USER /var/www/html/example/

Step 4: Using the sites-available default file create another copy file.

cp /etc/nginx/sites-available/default /etc/nginx/sites-available/example

Step 5: Make the changes in newly created file.

server {

listen 80;

listen [::]:80;

root /var/www/html/example;

 # Add index.php to the list if you are using PHP

index index.html index.htm;

server_name www.example.com example.co;

location / {

 # First attempt to serve request as file, then

 # as directory, then fall back to displaying a 404.

try_files $uri $uri/ =404;

}

```
server {
        listen 80;
        listen [::]:80;
```

```
        root /var/www/html/example;


        # Add index.php to the list if you are using PHP
        index index.html index.htm;

        server_name www.example.com     example.co;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;

        }
```

Step 6: Enabling Server Blocks

We must enable our server block files now that we have them. Symbolic links from these files to the sites-enabled directory, which Nginx reads from at startup, can be used to do this.

ln -s /etc/nginx/sites-available/example /etc/nginx/sites-enabled/

Step 7: Next, check all of your Nginx files to ensure there are no syntax errors:

nginx -t

```
root@nishit-VirtualBox:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Step 8: In order to activate your modifications, restart Nginx if no issues were discovered:

systemctl restart nginx

```
root@nishit-VirtualBox:~# systemctl restart nginx.service
```

Step 9: Modify etc/hosts file.

nano etc/hosts

Here add system ip and domain name.

```
192.168.10.133  www.example.com example
```

Step 10: Check the result.