DOCUMENTATION

This project uses a fork of Nmap called ScanPBNJ (from PBNJ 2.0 suite) to continuously monitor and update your database according to the services’ status.

It finally provides a clean and interactice UI to the admin to view the changes in the network.

Since it requires nmap, this project is made for Linux only and tested on Kali Linux 2018.

**Installation/ prerequisites**

Steps to install PBNJ:

Download the given file or sourceforge or from github:

Link 1: https://sourceforge.net/projects/pbnj/files/pbnj/pbnj-2.04/pbnj-2.04.zip/download?use\_mirror=excellmedia&download=

Link 2: github link

2. To Install PBNJ Properly (make sure you have the needed dependencies)

Go to terminal and type the following:

perl Makefile.PL

make

make test

\*\* As root \*\* make install

ScanPBNJ requires some additional perl modules to run

To download run the following commands (Install a module using CPAN):

$ sudo cpan

Make sure you have the latest version of CPAN installed

cpan>install CPAN

cpan>install Bundle::CPAN

Then when you see the cpan> prompt type install and the name of the module.

Type the following:

cpan>install Nmap::Parser

cpan>install File::HomeDir

cpan>install YAML

cpan>install DBI

cpan>install DBD::SQLite

cpan>install XML:: Twig

cpan>install File::Which

cpan>install CSV\_XS

Also, you will need Nmap (any version will do)

For more details, see : <http://pbnj.sourceforge.net/docs.html>

Try running scanpbnj using any ip,

Example : scanpbnj 127.0.0.1

The scan should run perfectly but the details are not being saved in a database

If it show error relating to shell,install:

cpan>install shell

3. Setting up a database (MYSQL v4 or v5)

Install mysql 4.1 or mysql 5.0

apt-get install mysql-server-4.1

or

apt-get install mysql-sever-5.0

Step 3.1 : Connect to mySql

#mysql

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 3.2: To read help

mysql>\h

... snipped ...

Note when using MySQL all lines start with mysql>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 3.3: Create a MySQL PBNJ database

mysql>CREATE DATABASE pbnj;

Output:

Query OK, 1 row affected (0.00 sec)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 3.4: Add a user (here, bob)

Type the following command to create a user called bob with a password called toor:

mysql> GRANT SELECT,INSERT,UPDATE,CREATE ON pbnj.\*

-> TO 'bob'@'SERVER IP' IDENTIFIED BY 'toor';

Please, note that: SERVER IP should be replaced with the actual

ip address of the server.

ex:

mysql> GRANT SELECT,INSERT,UPDATE,CREATE ON pbnj.\*

-> TO 'bob'@'192.168.1.50' IDENTIFIED BY 'toor';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

s

Step 3.5: Setting up the configuration file config.yaml in .pbnj-2.0

# cd

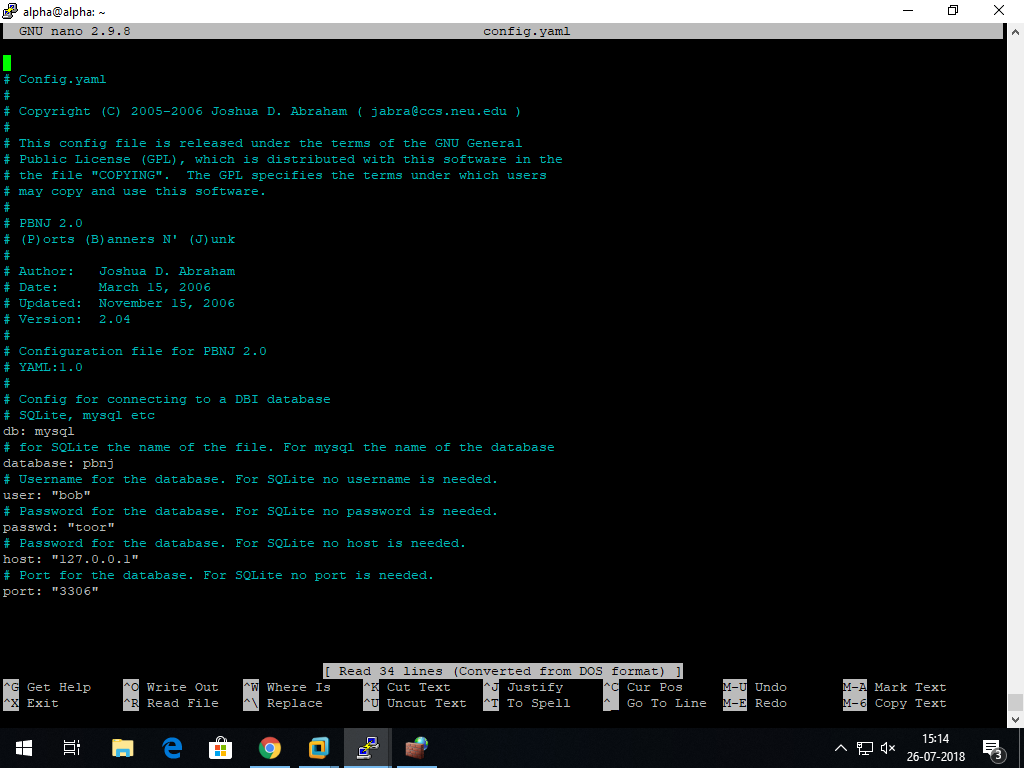
# mkdir –p .pbnj-2.0

# cd .pbnj-2.0/

# nano config.yaml

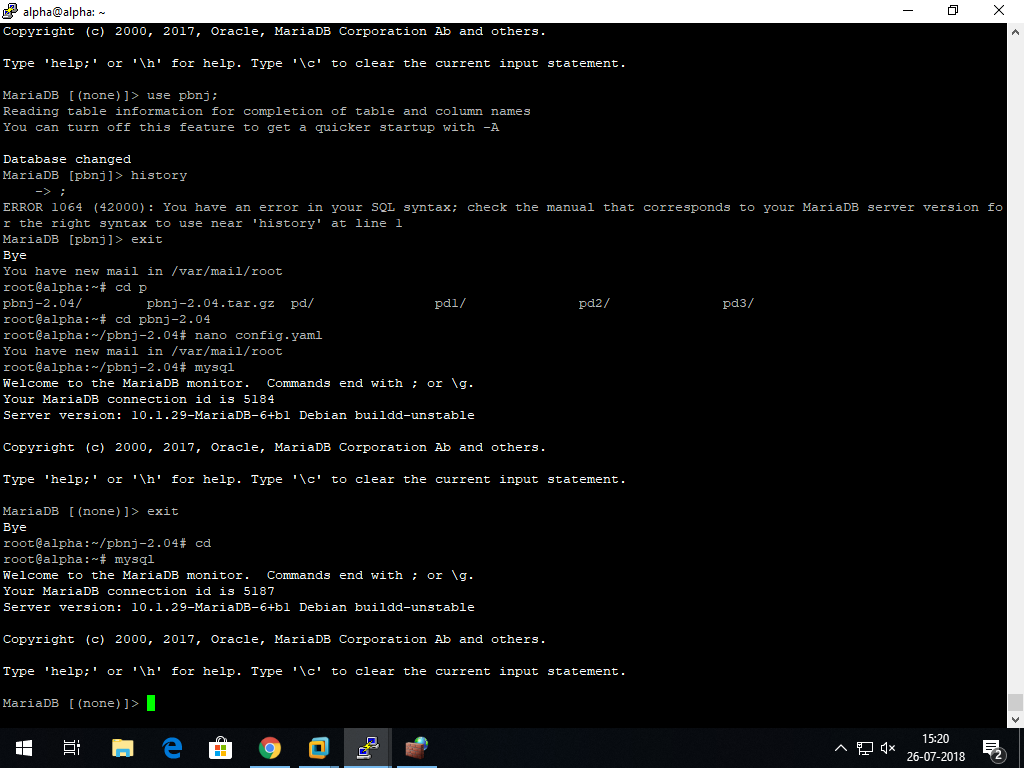
# nano config.yaml

Set the following configuration:

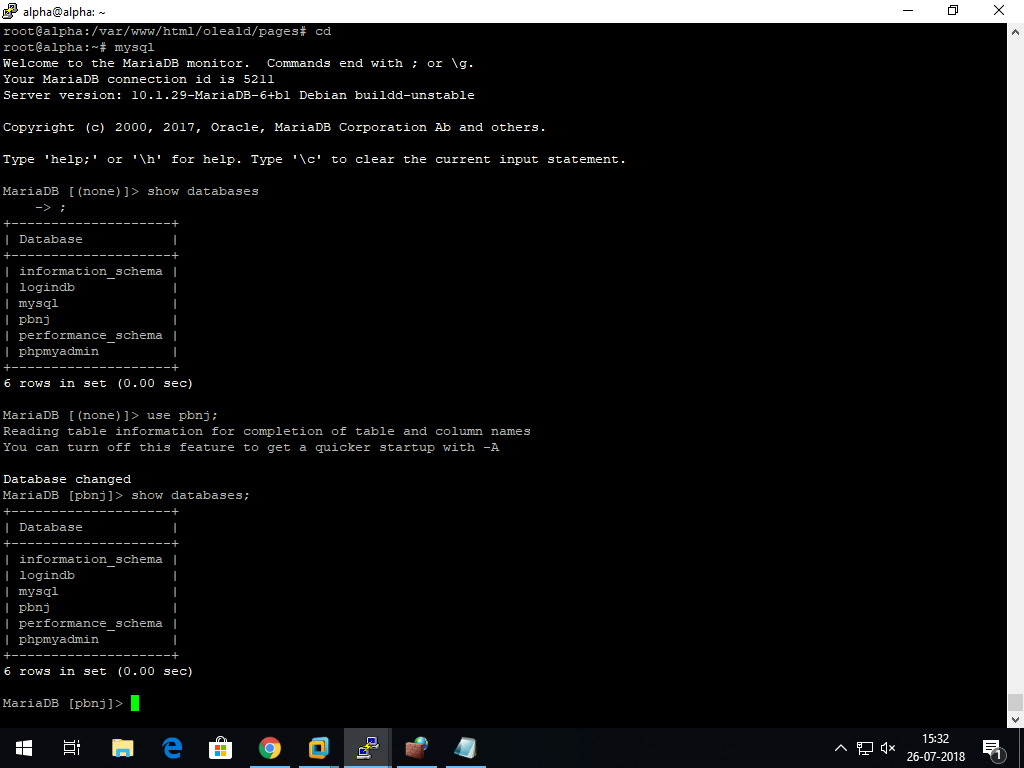


Step #7: Verify your data is in the database:

# mysql



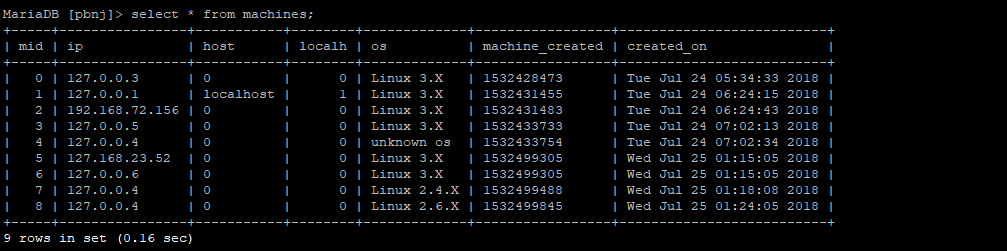
mysql> show databases;



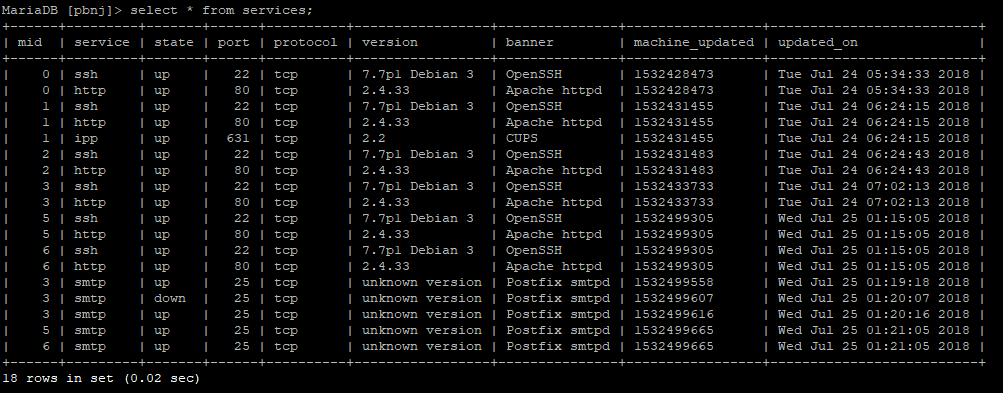
We will create in the steps to come

mysql> use pbnj;

mysql> select \* from machines;



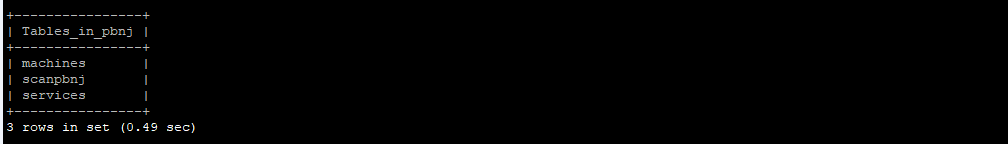
mysql> select \* from services;



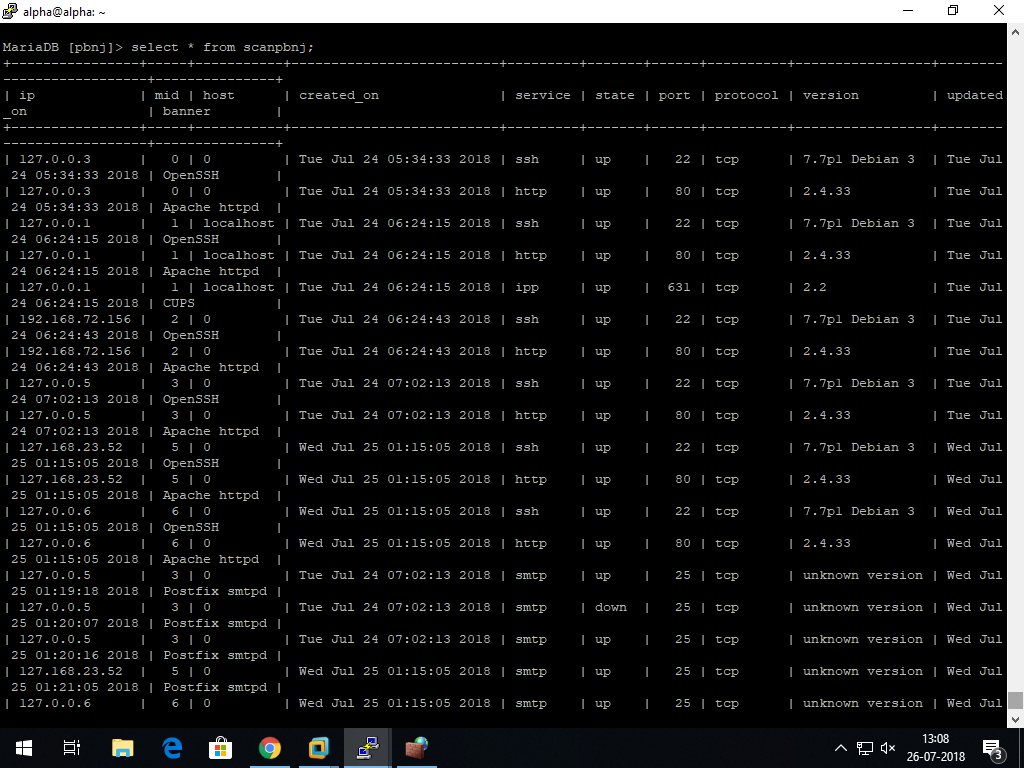
Create a view to merge both the tables services and machines so that data can entirely be co related, by typing the following command:

mysql> create view scanpbnj M.ip AS ip,M.mid AS mid, M.host AS host, M.created\_on AS created\_on,S.service AS service,S.state AS state,S.port AS port,S.protocol AS protocol,S.version AS version,S.updated\_on AS updated\_on,S.banner AS banner from (machines M join services S) where (M.mid = S.mid;

mysql> show tables;



mysql>select \* from scanpbnj;



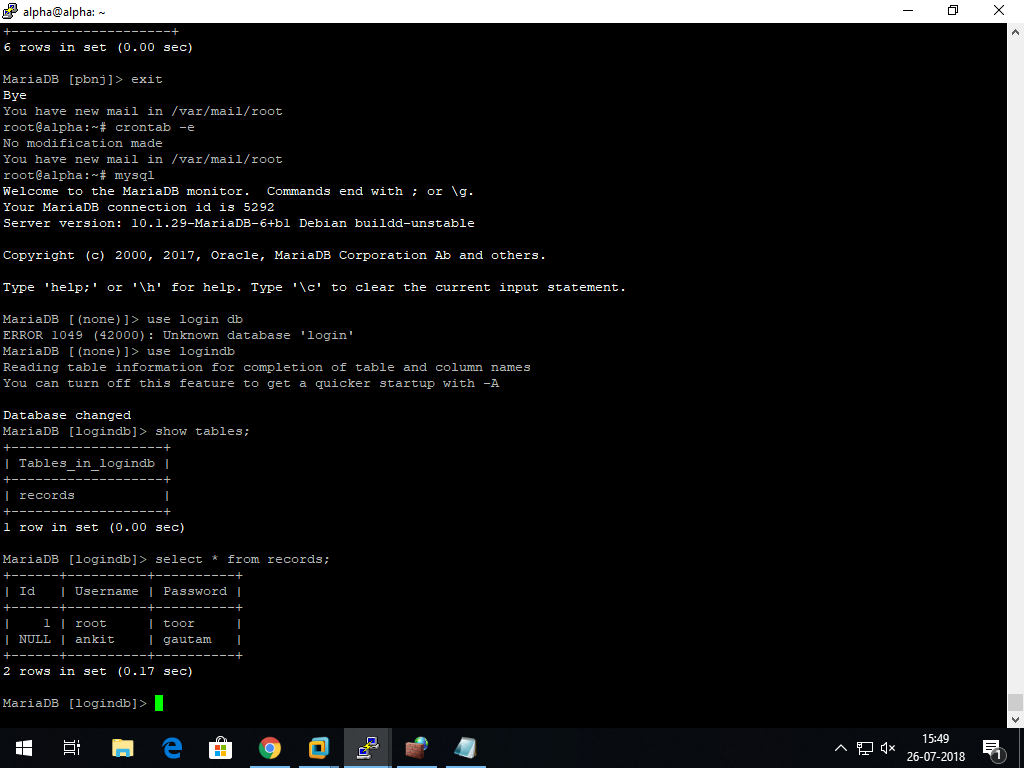
Create a database to keep your credentials for login called ‘logindb’

mysql>create database logindb;

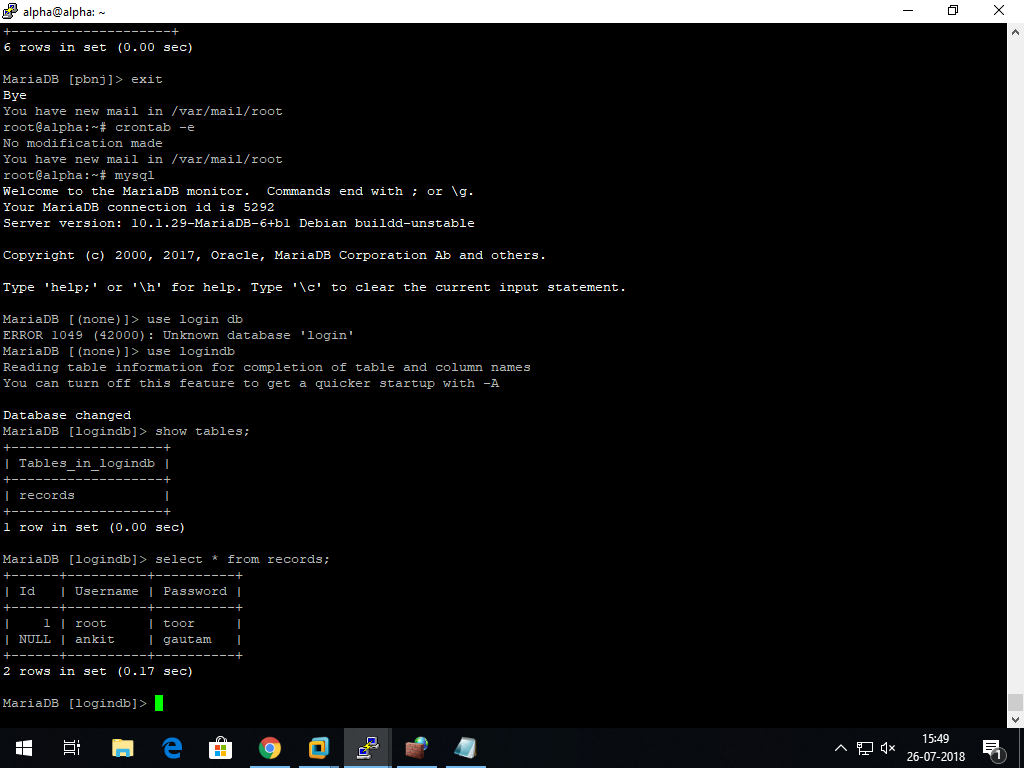
mysql>use logindb;

mysql>create table records (Id integer AUTO\_INCREMENT , Username varchar(20), Password varchar(30));

mysql>show tables;



mysql>select \* from records;



That's it your backend is all set

Setting up cronjob:

To run the scan automatically after ‘x’ amount of time set up a cronjob.

The scripts to run it every minute has been provided.

Alter the duration according to your needs.

Place the scanx.sh file under the root folder, path: /root/

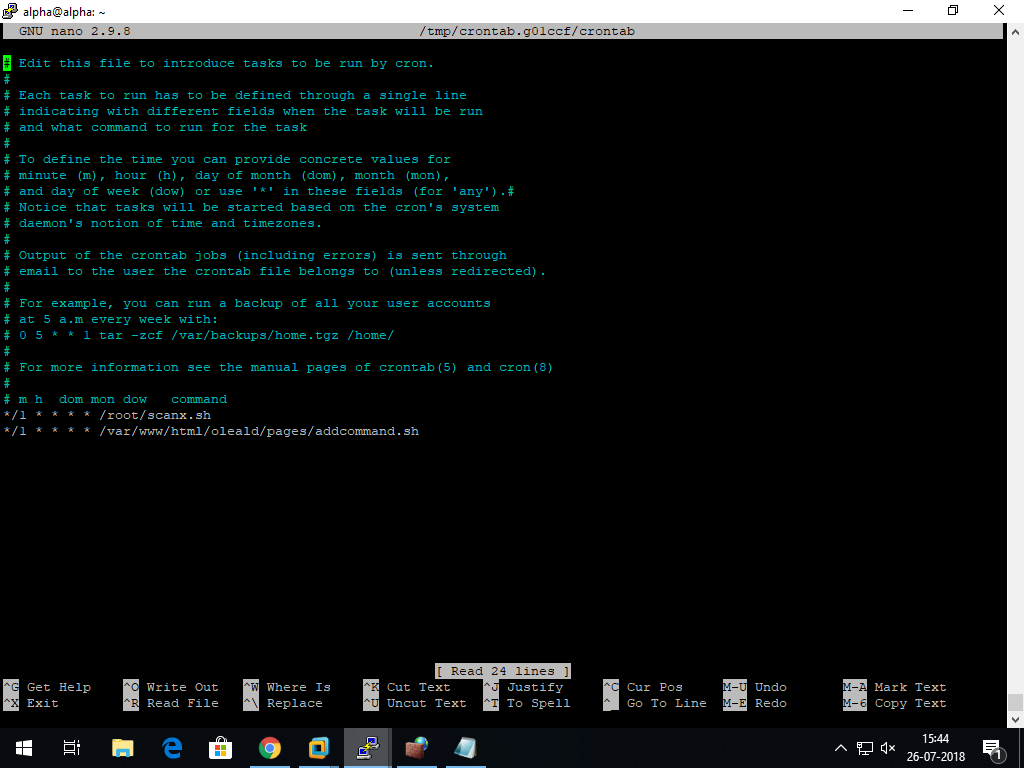
Goto terminal write:

crontab -e

then write this in your file :

\*/1 0-23 \* \* \* /root/scanx.sh

\*/1 \* \* \* \* /var/www/html/oleald/pages/addcommand.sh



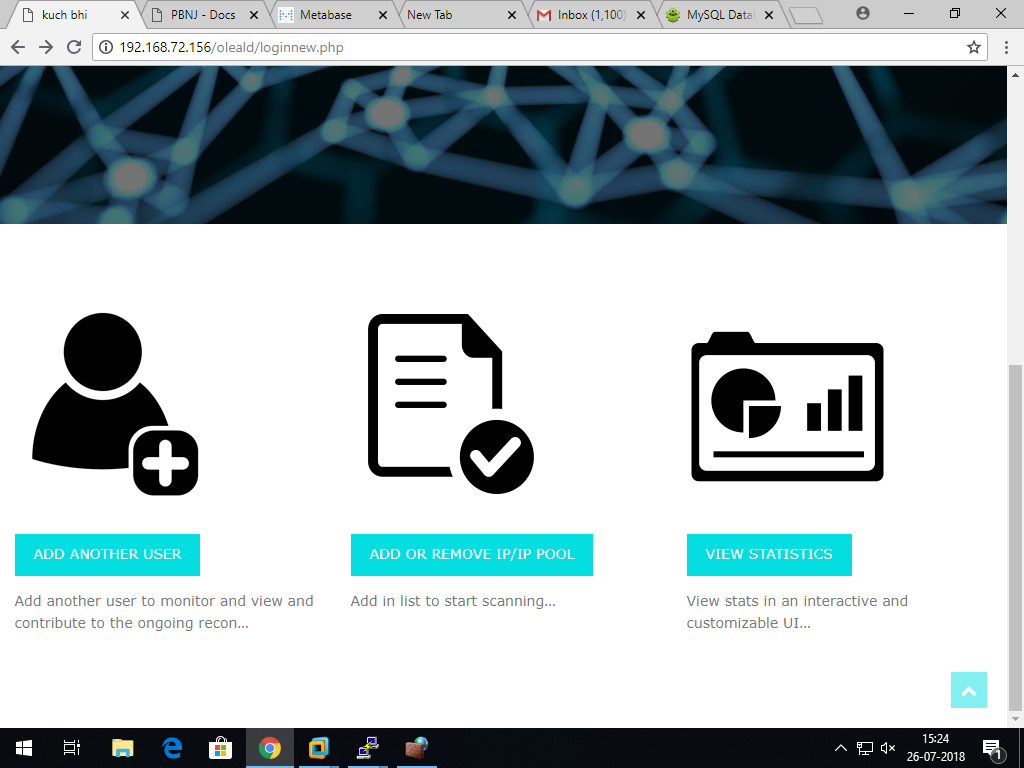
Now for the front end,

Download and save all the given files to the path where your server can access these.

For eg , if running on localhost and using apache2 server , save your files in /var/www/html folder.

Then go to your browser and type XX.XX.XX.XX/AA/index.php

The following should be displayed after logging in:



Now you can easily navigate through the site.

To view statistics, click on the tab, this automatically open the stats tab and runs metabase (the reporting tool):

If you are opening it for the first time, you will have to setup metabase according to the following :

<https://www.metabase.com/docs/latest/setting-up-metabase.html>

You can customize your dashboard. Here is a sample :

