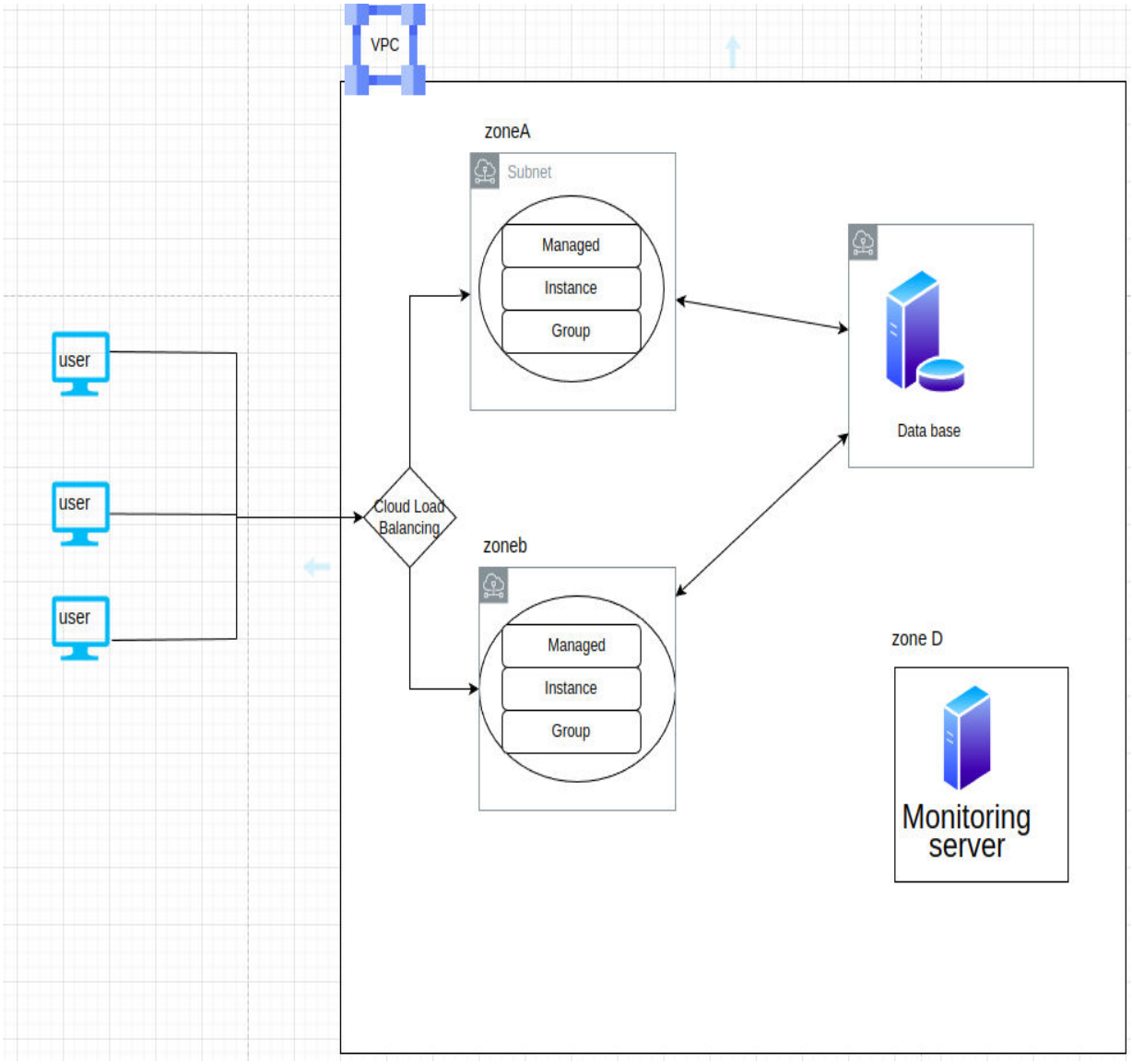
Hosting a ESPOCRM Application on Google Cloud Platform

EspoCRM is a free open-source web application that allows you to see, enter and analyze your company's relationships with customers and partners alike. Lightning fast and easily customizable, this web-based CRM solution will provide you with the tools to understand your consumers' behavior and cater your products and services specifically to their needs.

- Below is the Architecture for hosting ESPOCRM Application



How to Install EspoCRM on Ubuntu :

Creating a Compute Engine Instance:

We can create a instance using cloud Console or Cloud SDK, or using Cloud Shell.

■ Using Cloud Console:

Ensure that We have a project.

In Cloud Console =>Navigate to Compute Engine Service => Create Instance => 'instance Name' (we can specify our required reguirements like region, zone, machine type,etc)

■ Using Cloud Shell :

```
gcloud compute instances create <instance name>
```

```
gcloud compute instances create instance-1 –machine-type=n1-standard-1
    --zone=us-central1-b
```

After instance Creation, click on SSH to install required reguirements and dependencies to host the application

Requirements:

- Php 7.4 and later
- mysql (5.7 &above) or mariadb(10.1 &above)
- Nginx

➢ **To install mysql run below commands :**

```
sudo apt-get update
sudo apt-get install mysql-server
sudo mysql_secure_installation
```

You will get a prompt to answer the following questions:

- Enter the current password for root (Press Enter for none): Then, just press the Enter key
- Set the root password? [Y/n]: Y
- New-password: Enter the password
- Next, re-enter the new password: Repeat your password
- Remove any anonymous users? [Y/n]: Y
- Disallow the root-login remotely? [Y/n]: Y
- Remove the test database and access to it? [Y/n]: Y
- Reload the privilege tables now? [Y/n]: Y

You will need to restart the MariaDB server using the following command:

```
sudo systemctl restart mysql.service
```

- **To install php & nginx run below commands :**

```
sudo apt-get install php7.4
sudo apt-get install nginx
```

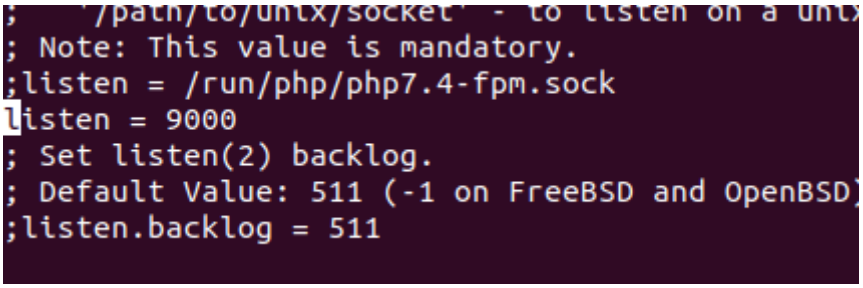
Dependencies:

1. To Install required php libraries run the following command

```
sudo apt-get update
sudo apt-get install php-mysql php-json php-gd php-zip php-imap php-mbstring
sudo apt-get install php-curl php-exif php-ldap
sudo phpenmod imap mbstring
sudo apt-get install php7.4-fpm
sudo service nginx restart
```

By default fpm is listening to socket. We need to change it to the port .

```
vim /etc/php/7.4/fpm/pool.d/www.conf
```



Comment(;) the line which is listening to socket(php7.4-fpm.sock) and add the line `listen = 9000`

we made fpm to listen on 9000 port , then restart the fpm service

```
service php7.4-fpm restart
```

- Edit the php.ini file add change follwing changes:

```
max_execution_time = 180
max_input_time = 180
memory_limit = 256M
post_max_size = 50M
upload_max_filesize = 50M
```

■ **To Download the ESPOCRM installation package :**

1. To get the latest version of espoCRM follow the below given link ⬇

```
https://www.espocrm.com/download/
```

➢ Then, unzip the downloaded zip file using unzip command:

```
unzip <file name>
```

➢ Then move the file to *var*/www/html location:

```
mv <file name>  var/www/html
```

➢ To edit the nginx configuration :

```
cd etc/nginx/sites-availble
```

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/espocrm.conf
```

◆ To edit espocrm.conf file use vim espocrm.conf
remove the entire configuration and copy and paste below provided configuration : ⬇

```
https://docs.espocrm.com/administration/nginx-server-configuration/
```

In espocrm.conf file change the *path-to-espo the absolute path of your espocrm path (var/www/html/espocrm/public)*

```
root /var/www/html/espocrm/public; # path to public dir

location /client {
    root /var/www/html/espocrm; # path to espocrm root dir
    autoindex off;

    location ~* ^.+.(js|css|png|jpg|jpeg|gif|ico|tpl)$ {
        access_log off;
        expires max;
    }
}
```

And add the port number where the fpm is running .

```
    location ~ \.php$ {
fastcgi_pass espocrm-php:9000;
 include fastcgi_params;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
fastcgi_param QUERY_STRING $query_string;
}
```

```
location ~ \.php$ {
    fastcgi_pass localhost:9000;
```

Change the espocrm-php:9000 to localhost:9000;

- Link the espocrm.conf file to */etc*/nginx/sites-enabled/ location

```
sudo ln -s /etc/nginx/sites-available/espocrm.conf  /etc/nginx/sites-enabled/espocrm.conf
```

- To check the nginx configuration file syntax run command

```
nginx -t
```

```
sudo service nginx restart
```

- **Creating EspoCRM Database :**

After, installing all the packages, now we need to create databaseNow, run the command to create the EspoCRM database.

Run the below command to logon to the database server. You will get a prompt for a password. So, type the root password you created above:

```
sudo mysql -u root -p
```

1) To create Database :

```
CREATE DATABASE espocrm;
```

2) Then create a database user as espo-user, with new password.

```
CREATE USER 'espo-user'@'localhost' IDENTIFIED BY 'password';
```

3) Grant the user Privileges:

```
GRANT ALL ON espocrm.* TO 'espo-user'@'localhost';
```

4) Save the changes and exit

```
FLUSH PRIVILEGES;
```

```
exit ;
```

Then move to Espocrm directory location : /*var*/www/html/espocrm

- To set the permisions , execute below commands in the terminal :

```
sudo chown -R www-data:www-data /var/www/espocrm
```

- Change access permissions for the directory.

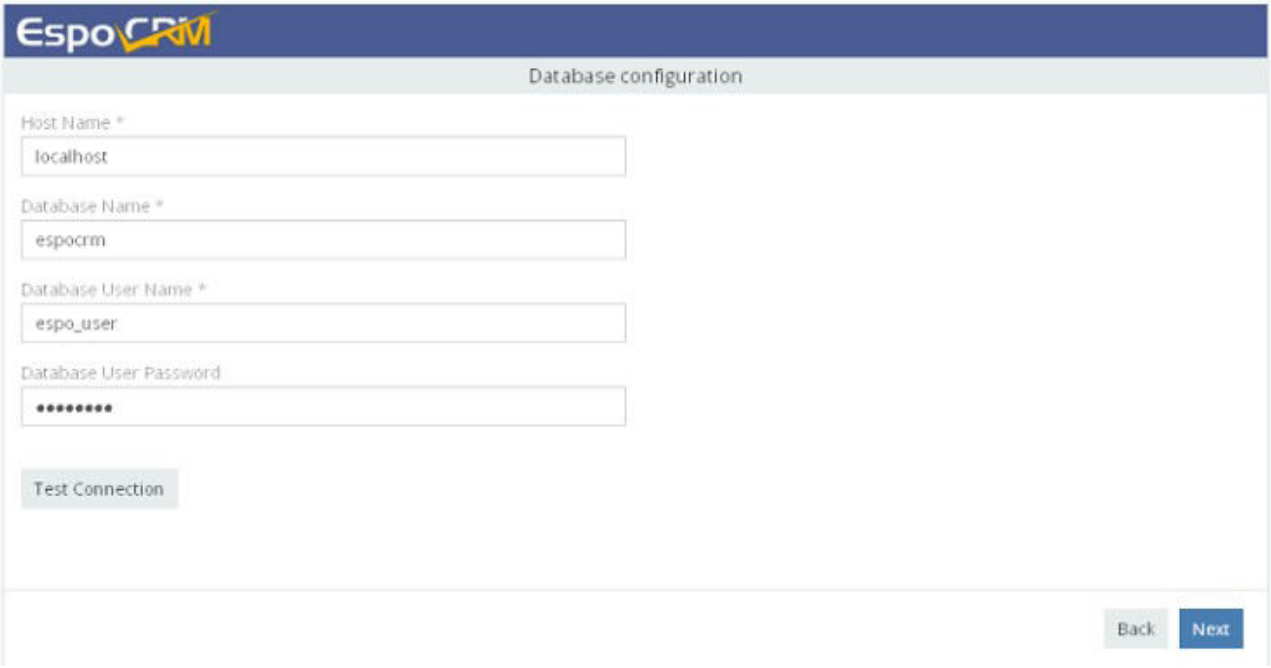
```
sudo chmod -R 755 /var/www/espocrm
```

If you see the following screen the installation was successful

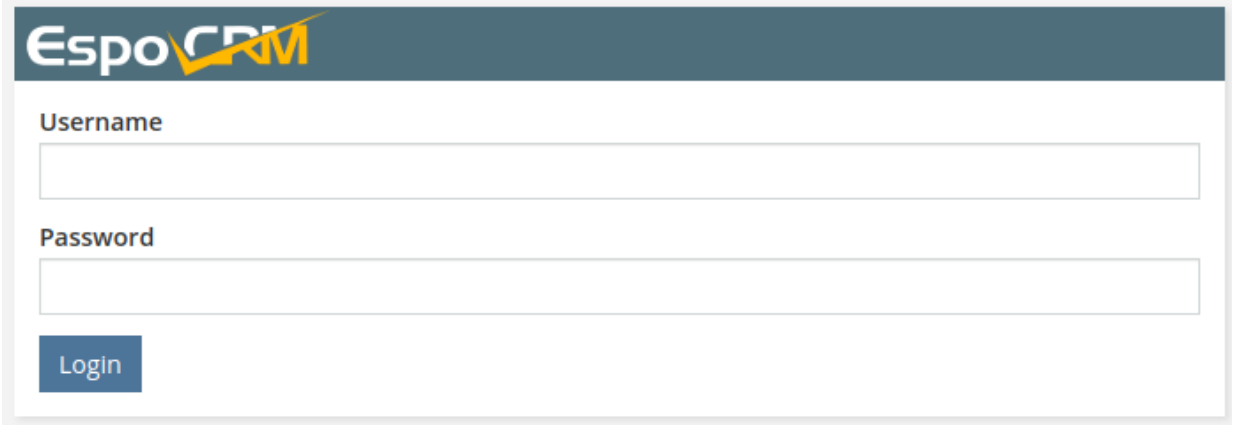
- Then open the browser and enter the domain name / localhost:9000 to view the Espocrm application

If you see the following screen the installation was successful

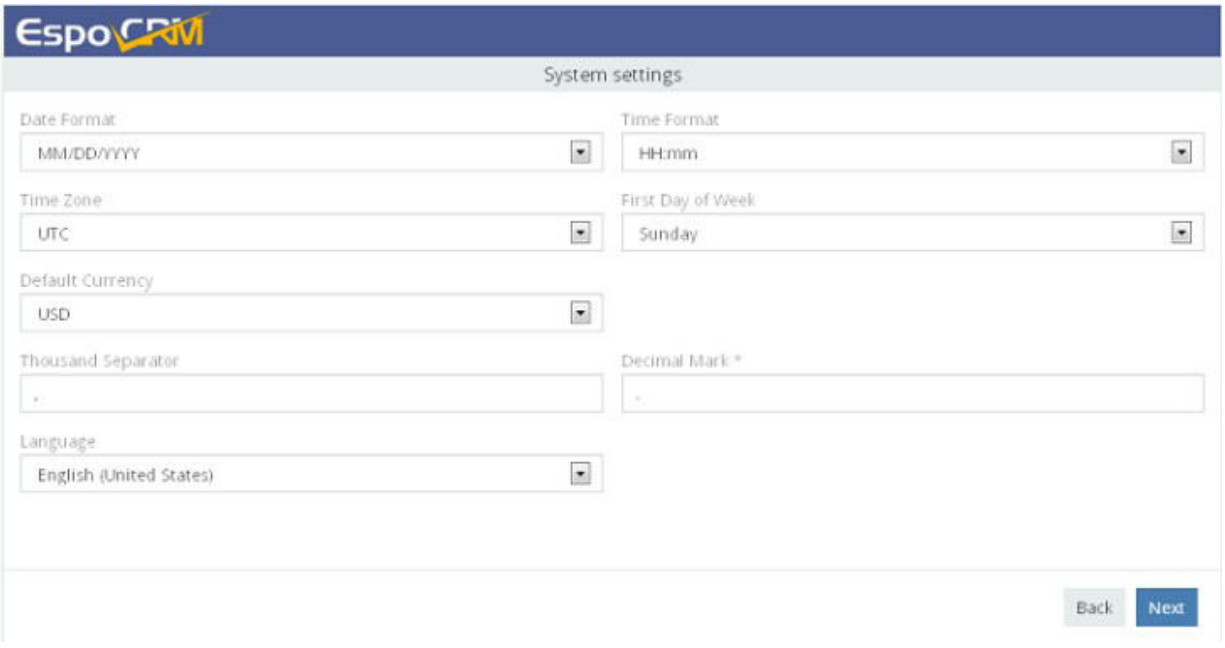
- Enter the details for your newly created MySQL database.



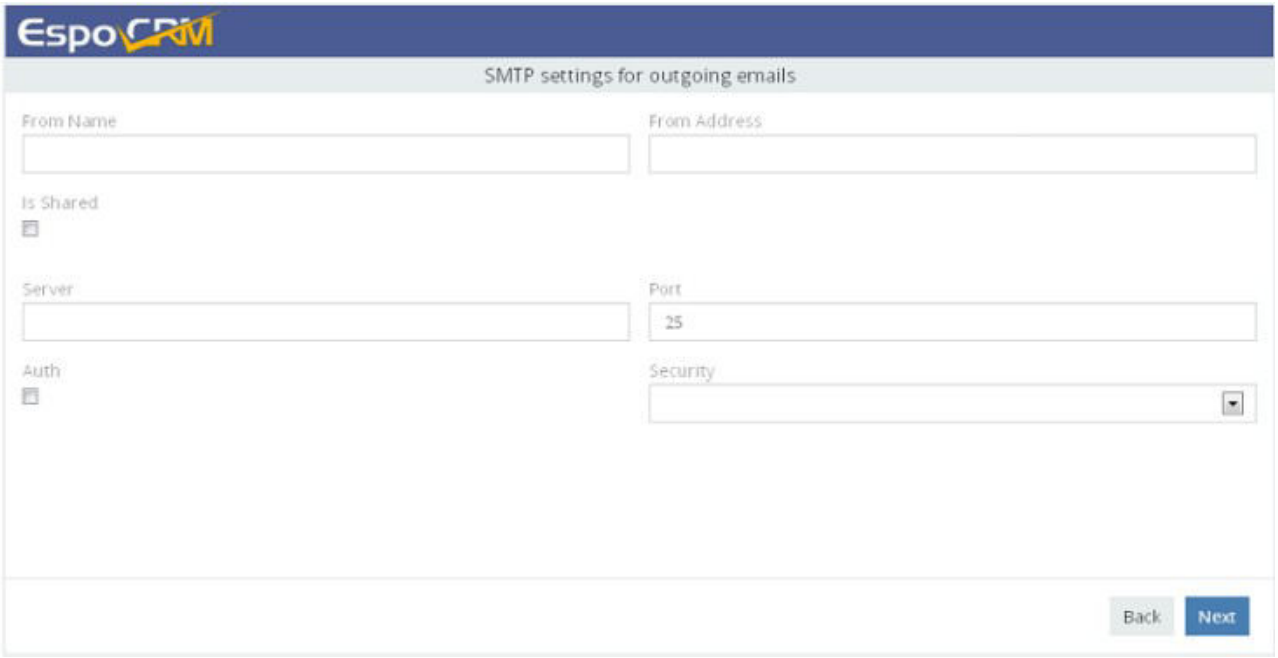
- After entering everything correctly. Then, you will have EspoCRM installed and ready to use.



Enter user name and password to login,



- Enter SMTP settings for outgoing emails. This step can be skipped by clicking the *Next* button.



The installation was done , now we need to set up a corn

Set up the crontab expression

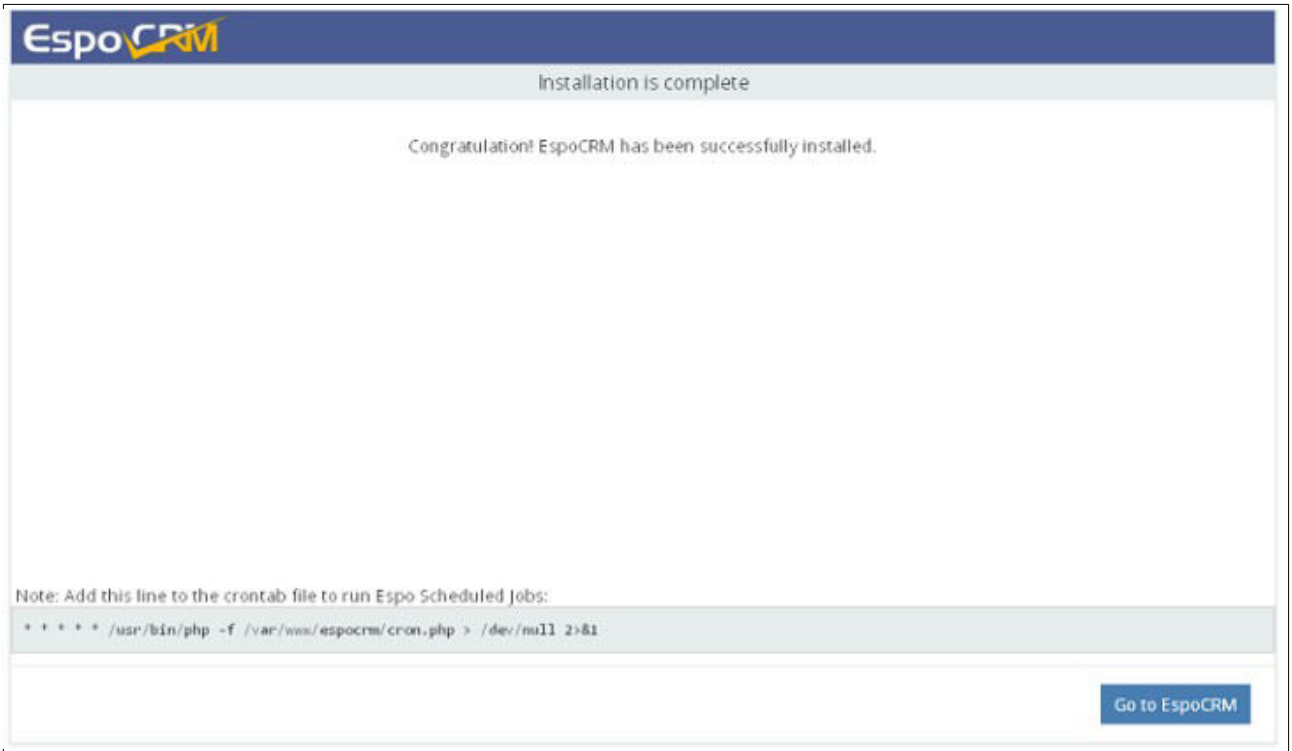
- Edit crontab to enable automation of EspoCRM tasks using Crontab.

  Open a terminal and run this command:

```
crontab -e -u WEBSERVER_USER
```

- Copy the below expression and Add the following expression at the bottom of the file. Save and close the file.

```
* * * * * /usr/bin/php -f /var/www/html/espocrm/cron.php > /dev/null 2>&1
```



Installing Prometheus for Monitoriing

- Requirements:

  prometheus

  grafana

  node exporter

Prometheus is a open-source monitoring system that collects metrics from our services and stores them in a time-series database and has a good alerting mechanism.

- *create a user for prometheus :*

```
sudo useradd --no-create-home --shell /bin/false prometheus
```

- *create folders for prometheus in /etc directory & /var/lib :*

```
sudo mkdir /etc/prometheus
sudo mkdir /var/lib/prometheus
```

- set the user and group ownership on the new directories to the prometheus user:

```
sudo chown prometheus:prometheus /etc/prometheus
sudo chown prometheus:prometheus /var/lib/prometheus
```

- To install prometheus follow below steps :

```
wget https://github.com/prometheus/prometheus/releases/download/v2.38.0/
      prometheus-2.38.0.linux-amd64.tar.gz
```

- to unzip the tar file, run below command:

```
tar -xvzf  prometheus-2.38.0.linux-amd64.tar.gz
```

Then copy the binary files (prometheus & promtool) to /usr/local/bin and configuration directories and files to (console_libraries, consoles & prometheus.yml) to /etc/prometheus.

```
sudo cp prometheus-2.38.0.linux-amd64/prometheus /usr/local/bin/
sudo cp prometheus-2.38.0.linux-amd64/promtool /usr/local/bin/
```

- set the user and group ownership on the binaries to the prometheus user :

```
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
```

- copy the consoles, console_libraries,prometheus.yml files to /*etc*/prometheus location:

```
sudo cp -r prometheus-2.0.0.linux-amd64/consoles /etc/prometheus
sudo cp -r prometheus-2.0.0.linux-amd64/console_libraries /etc/prometheus
sudo cp  prometheus-2.0.0.linux-amd64/prometheus.yml /etc/prometheus
```

- set the user and group ownership on the directories to the prometheus user:

```
sudo chown -R prometheus:prometheus /etc/prometheus/
```

- create the prometheus service file :

```
sudo vim /etc/systemd/system/prometheus.service
```

- Add the following code to prometheus.service file

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file /etc/prometheus/prometheus.yml \
    --storage.tsdb.path /var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

- then reload the systemd

```
sudo systemctl daemon-reload
```

- then start the prometheus service using following command :

```
sudo systemctl start prometheus
```

- to check the service is started or not run the below command :

```
sudo systemctl status prometheus
```

- Install the node exporter:

node exporter exposes a wide variety of hardware and kernal related metrics.

- Create a user to node exporter:

```
sudo useradd --no-create-home --shell /bin/false node_exporter
```

To download node exporter run the below command on terminal:
using wget

```
https://github.com/prometheus/node_exporter/releases/download/v1.4.0-rc.0/
node_exporter-1.4.0-rc.0.linux-amd64.tar.gz
```

- to unzip the tar file run below command:

```
tar -xvzf node_exporter-1.4.0-rc.0.linux-amd64.tar.gz
```

- copy the binary to the /*usr*/local/bin directory and set the user and group ownership to the node_exporter :

```
sudo cp node_exporter-1.4.0-rc.0.linux-amd64/node_exporter /usr/local/bin
```

```
sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
```

- create a node_exporter service file :

```
sudo vim /etc/systemd/system/node_exporter.service
```

- Add the following code to node_exporter.service file

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

- then reload the systemd

```
sudo systemctl daemon-reload
```

- to start the node_exporter service using following command :

```
sudo systemctl start node_exporter
```

- to check the service is started or not run the below command :

```
sudo systemctl status node_exporter
```

- Configure prometheus.yml :
  The configuration file looks like this

```
# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets:
       - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: 'prometheus'
    scrape_interval: 5s


    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
    - targets: ['localhost:9090']
  - job_name: 'node_exporter'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9100']
```

save the file and exit

- Finally Restart the prometheus service:

```
sudo systemctl restart prometheus
```

- to check the service is started or not run the below command :

```
sudo systemctl status prometheus
```

- Installing Grafana:
  Grafana is a multi platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.
- By default, the Grafana package is not included in the Ubuntu 20.04 default repository. So you will need to add the Grafana official repository to your system.

- First, install all required dependencies using the following command:

```
apt-get install wget curl gnupg2 apt-transport-https software-properties-common -y
```

- Next, download and add the Grafana GPG key with the following command:

```
wget -q -O - https://packages.grafana.com/gpg.key | apt-key add -
```

- Next, add the Grafana repository to APT using the following command:

```
echo "deb https://packages.grafana.com/oss/deb stable main" | tee -a /etc/apt/sources.list.d/grafana.list
```

- Once the repository is added to your system, you can update it with the following command:

```
apt-get update -y
```

- To install grafana run below command in terminal:

```
sudo apt-get install grafana
```

- To start the grafana service run below command:

```
sudo systemctl start grafana-server
```

- to check the service is started or not run the below command :

```
sudo systemctl status grafana-server
```

- Next, you will need to install the Nginx as a reverse proxy for Grafana :

  Once the Nginx is installed, create an Nginx virtual host configuration file:

- Create grafana.conf in /*etc*/nginx/sites-available location :

```
sudo vim /etc/nginx/sites-available/grafana.conf
```

- Add the following code in grafana.conf file.

```
server {
        server_name _;
        listen 90 ;
        access_log /var/log/nginx/grafana.log;
    location / {
                proxy_pass http://localhost:3000;
        proxy_set_header Host $http_host;
                proxy_set_header X-Forwarded-Host $host:$server_port;
                proxy_set_header X-Forwarded-Server $host;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }
}
```

- Link and activate the configuration file

```
sudo ln -s /etc/nginx/sites-available/grafana.conf /etc/nginx/sites-enabled/grafana.conf
```

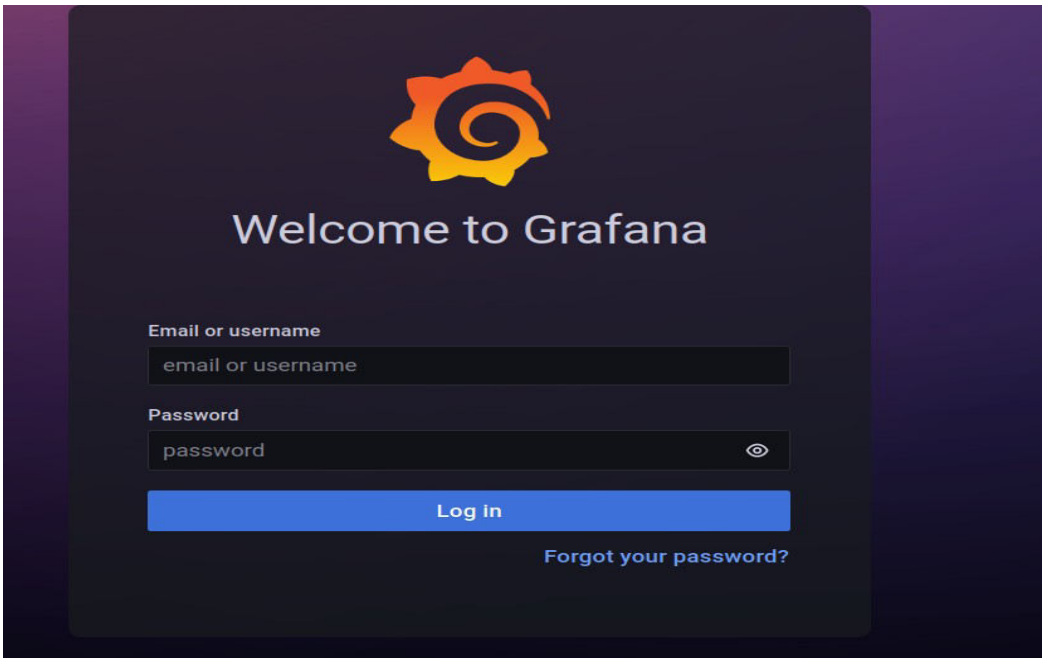
- to check the syntax

```
nginx -t
```

- Then restart the nginx service:

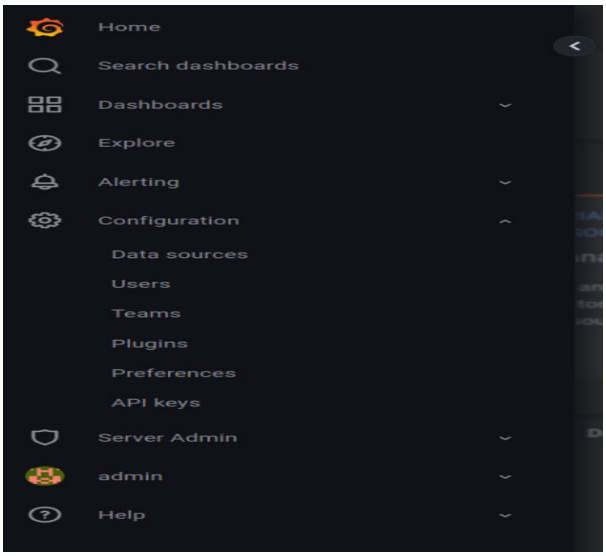
```
sudo systemctl restart nginx
```

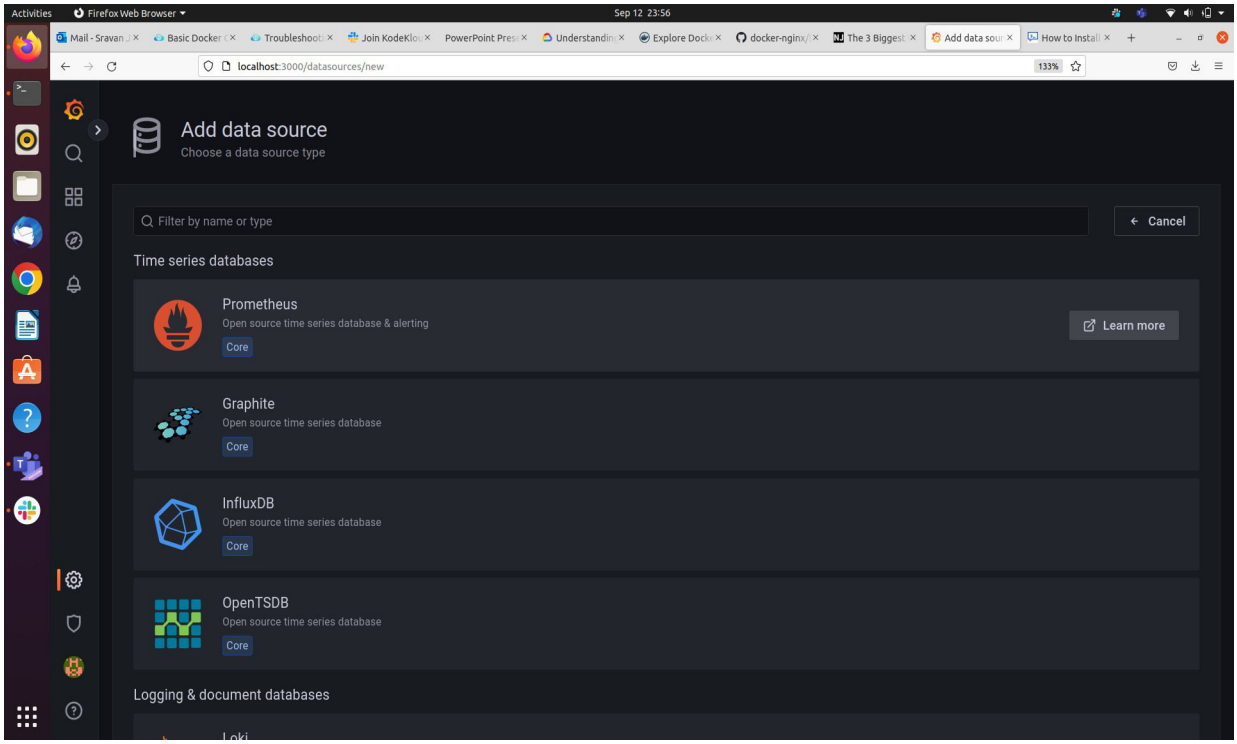
- Then open the browser and type localhost:3000 to access the grafana.



- Then create logins and login to the grafana dashboard.

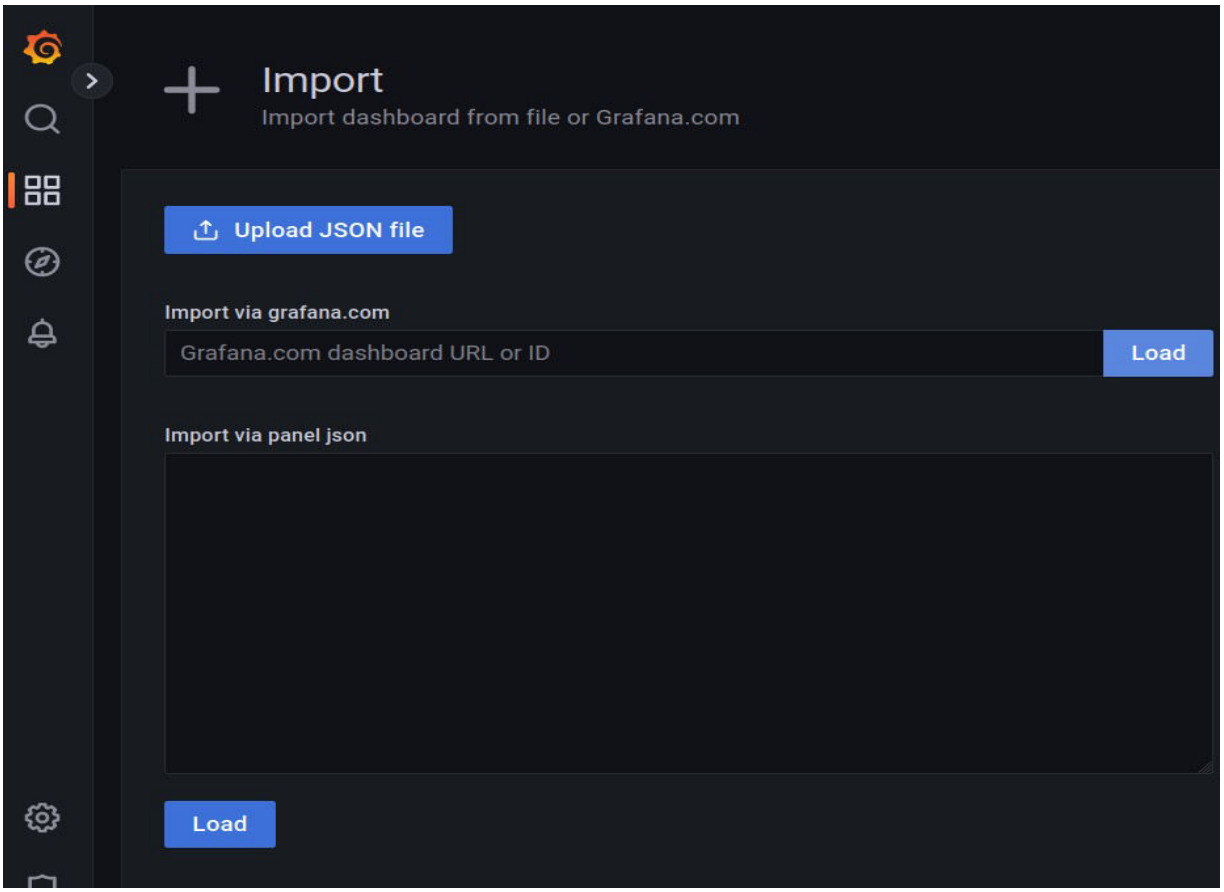
- Then Navigate to dashboard-->datasource-->add datasource--> prometheus



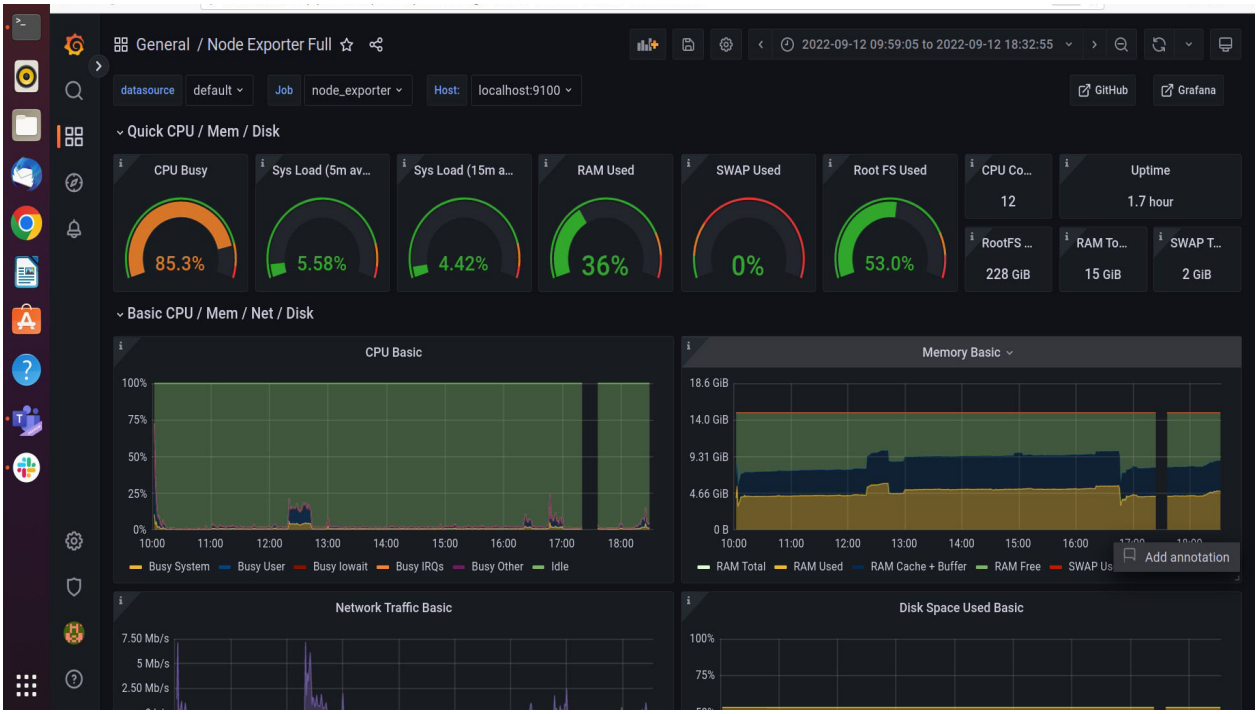


- Click on prometheus and then add the promethues url path to it(http://localhost:9000).

- Then navigate to dashboard-->import
- Then add the node exporter full id in the and load it.(1860)



➢ Then goto dashboards-->browse--->node_exporter full and view the metrics..



------THANK YOU -------