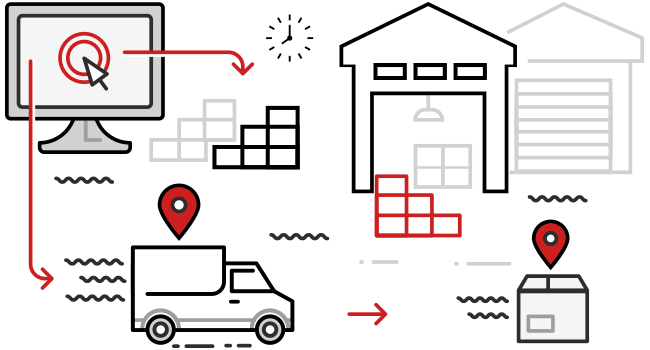
BIG DATA for e-LOGISTICS.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

****-

Authors

Tawanda Matsika, UWC

Kloniphani Maluleke, UWC

Supervisors

Prof. Osden Jokonya, UWC

Prof., Dr. Dany Meyer, HS-Neu-Ulm

Date

DECEMBER, 2018

# INTRODUCTION

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

# SCENARIO

# ENVIRONMENT

## Equipment

* Raspberry Pi 3 (B+)
* Sense Hat

## Software Framework

* Python
  + Flask, for the web interface
  + Sensor Hat
  + PupNub
  + Pyrebase
  + xlsxwriter
  + Pandas
* Cordova, JavaScript, Node.js
* MyMSQL
* Cloudera VM

## Online Accounts

* Google
  + Username - hnuelogistics@gmail.com
  + Password - dedomena123
    - Firebase, https://sensehat-51bd7.firebaseio.com
* Pubnub
  + ap…...
* Here Map
  + JavaScript
    - App ID - YVx7VH25wvkdlUFvGjC8
    - App Code - \_DmLSgB2oR1uQ1LrG0GbVQ

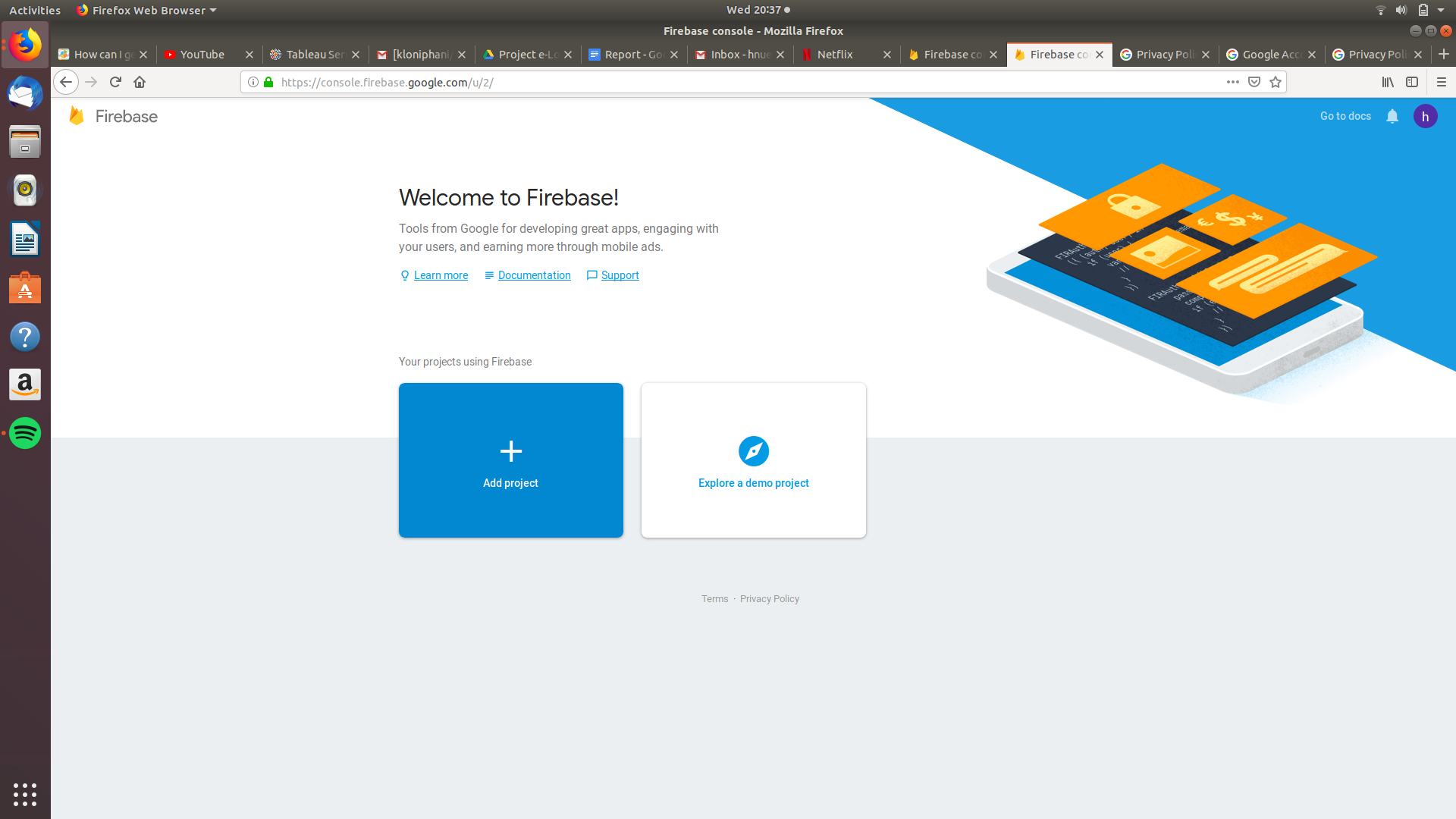
## Procedure

* Lorem ipsum dolor sit amet
* Consectetuer adipiscing elit
* Sed diam nonummy nibh euismod

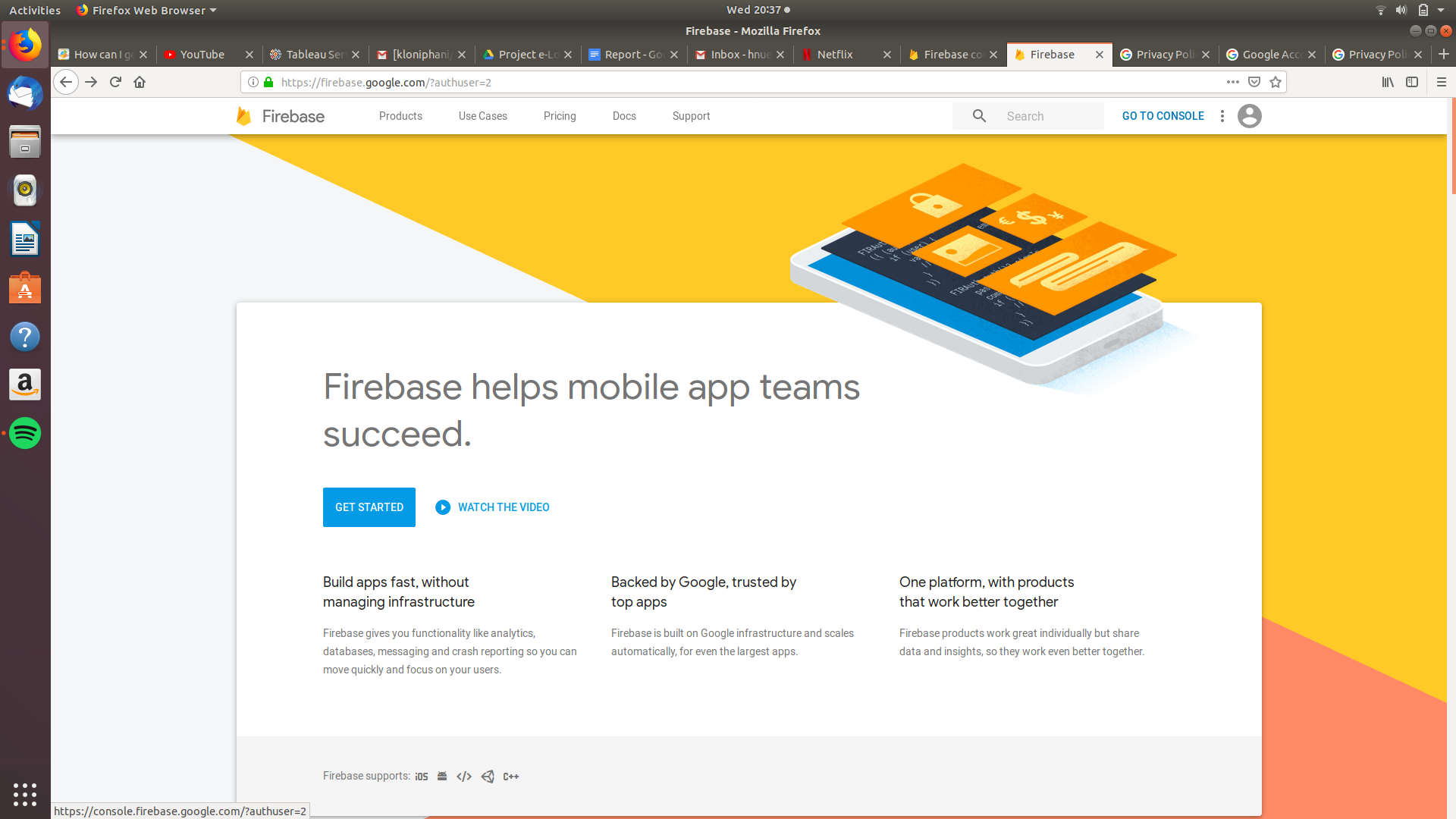
# MODEL

## Creating the realtime NoSQL Firebase database

**Step 1 :** Go to the Firebase homepage and click on the **Go to console** link on the top right corner and login with your Google account details if your not logged in.



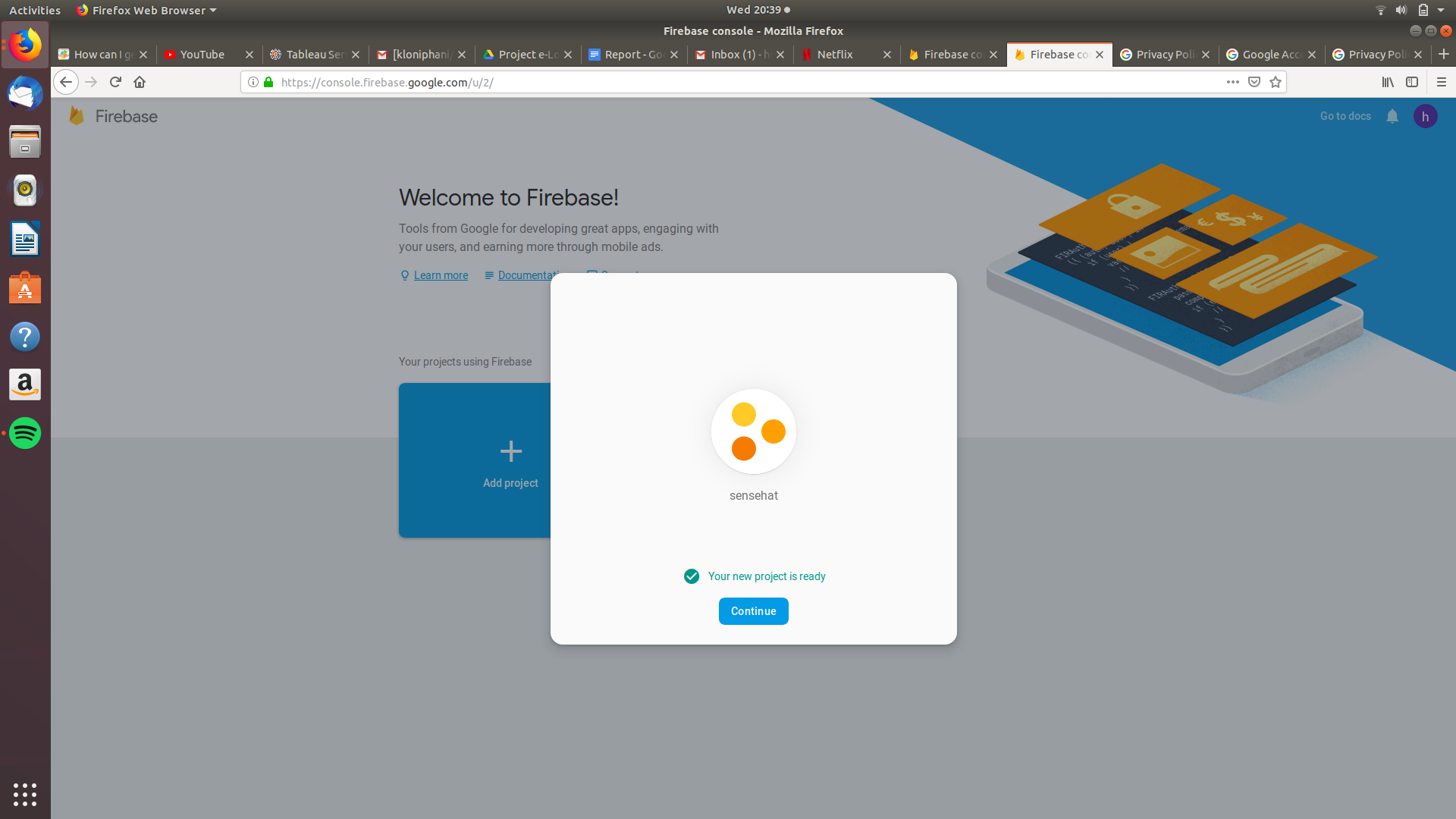
**Step 2:** To create a new project click on **Add project**.



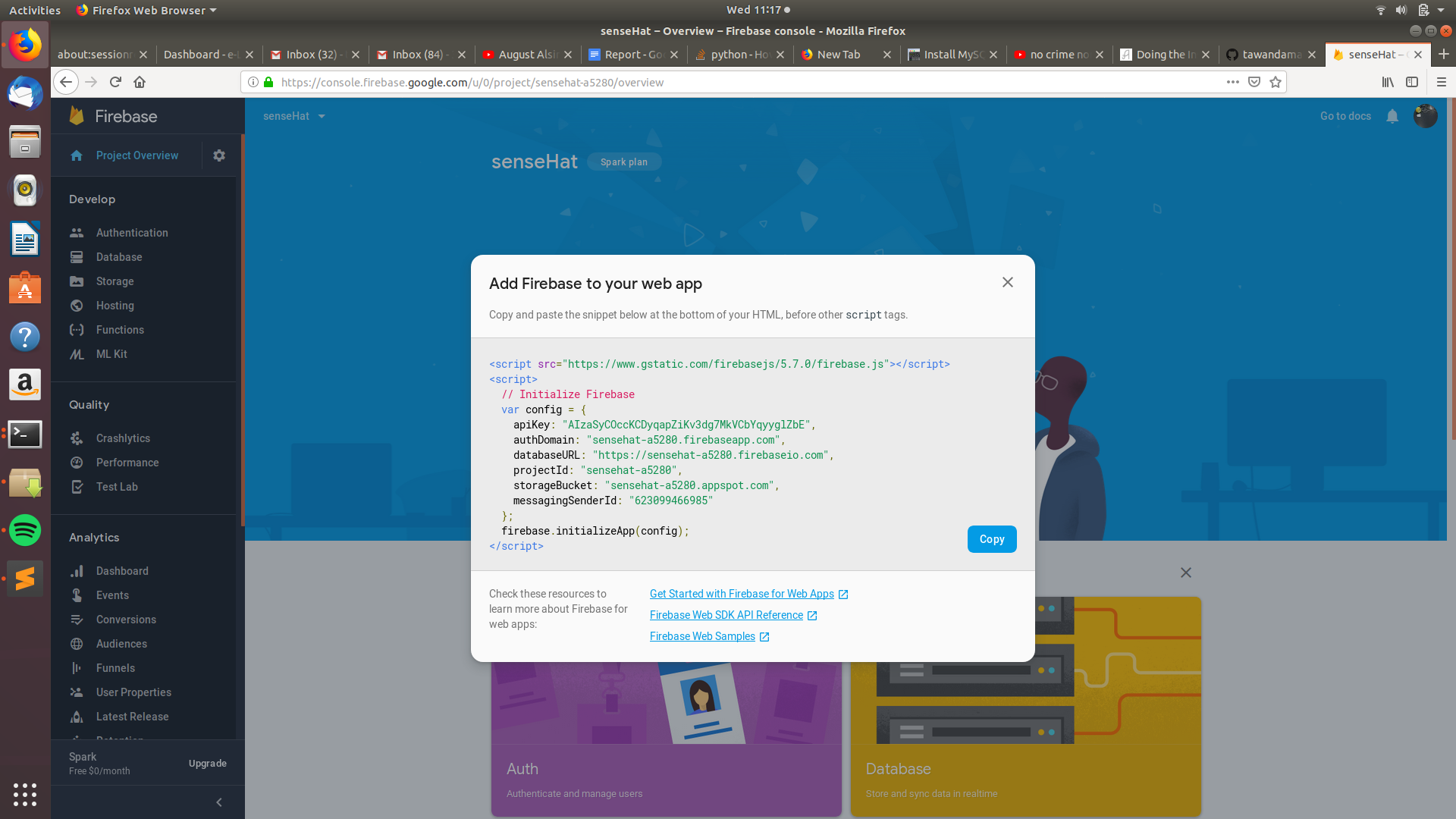
**Step 3:** A dialog will pop up, enter the **Project name** and your **Location** and agree to the Firebase terms and conditions and click on the **Create Project** button.

****

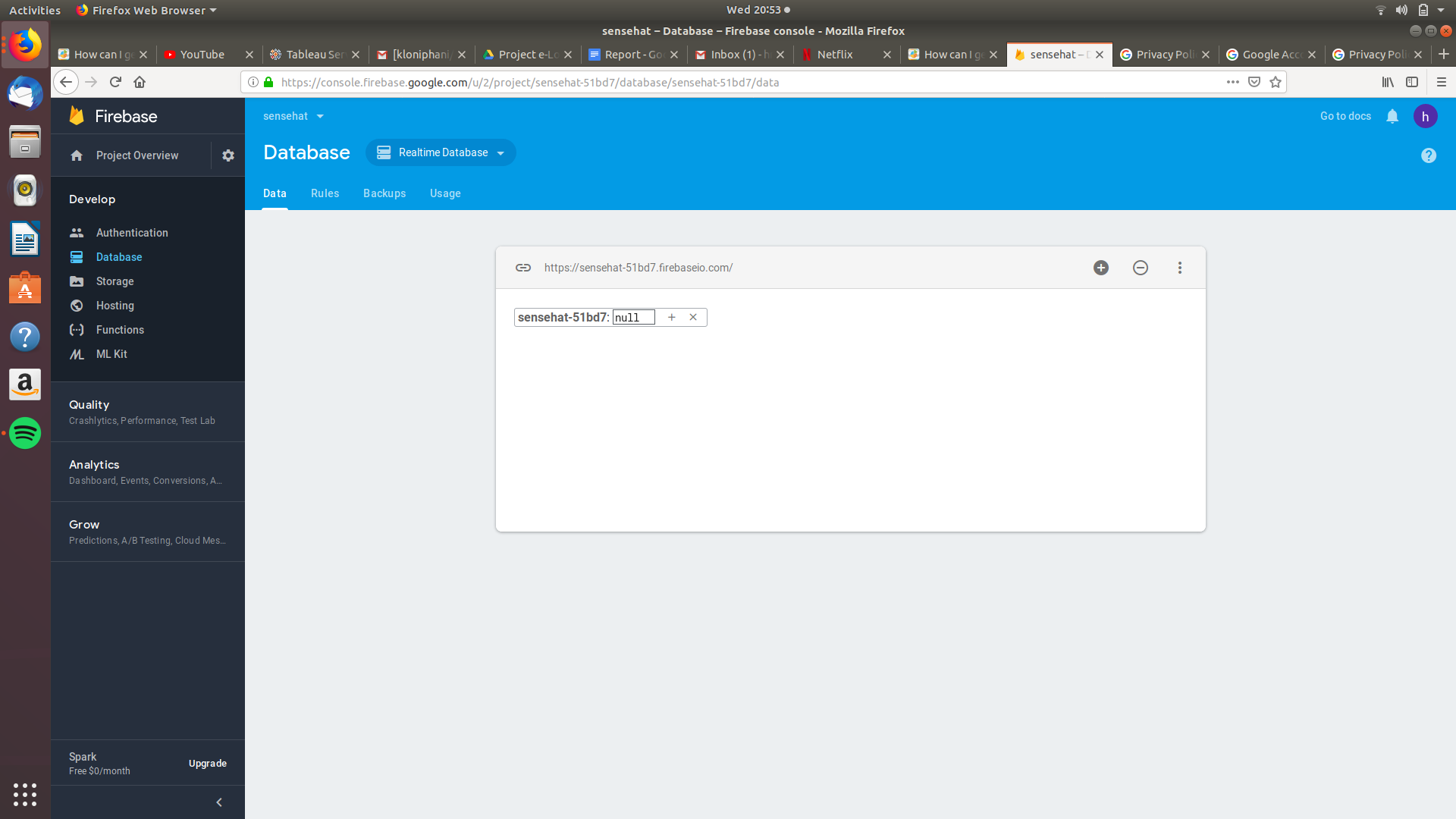
**Step 4:** Click on the **Continue** button.



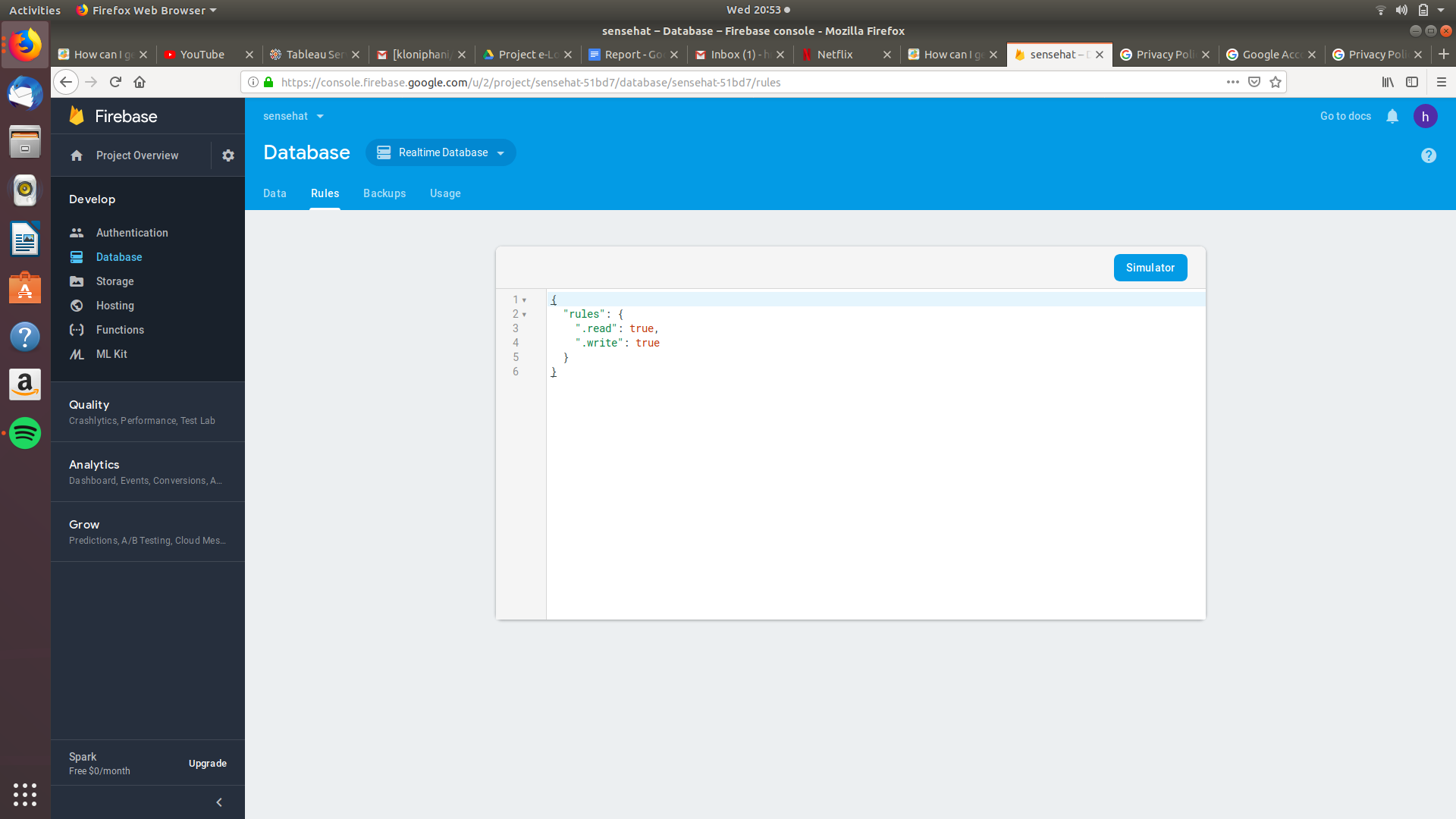
**Step 5:** Click on the **</>** link and save the Firebase database API keys and details that appear.

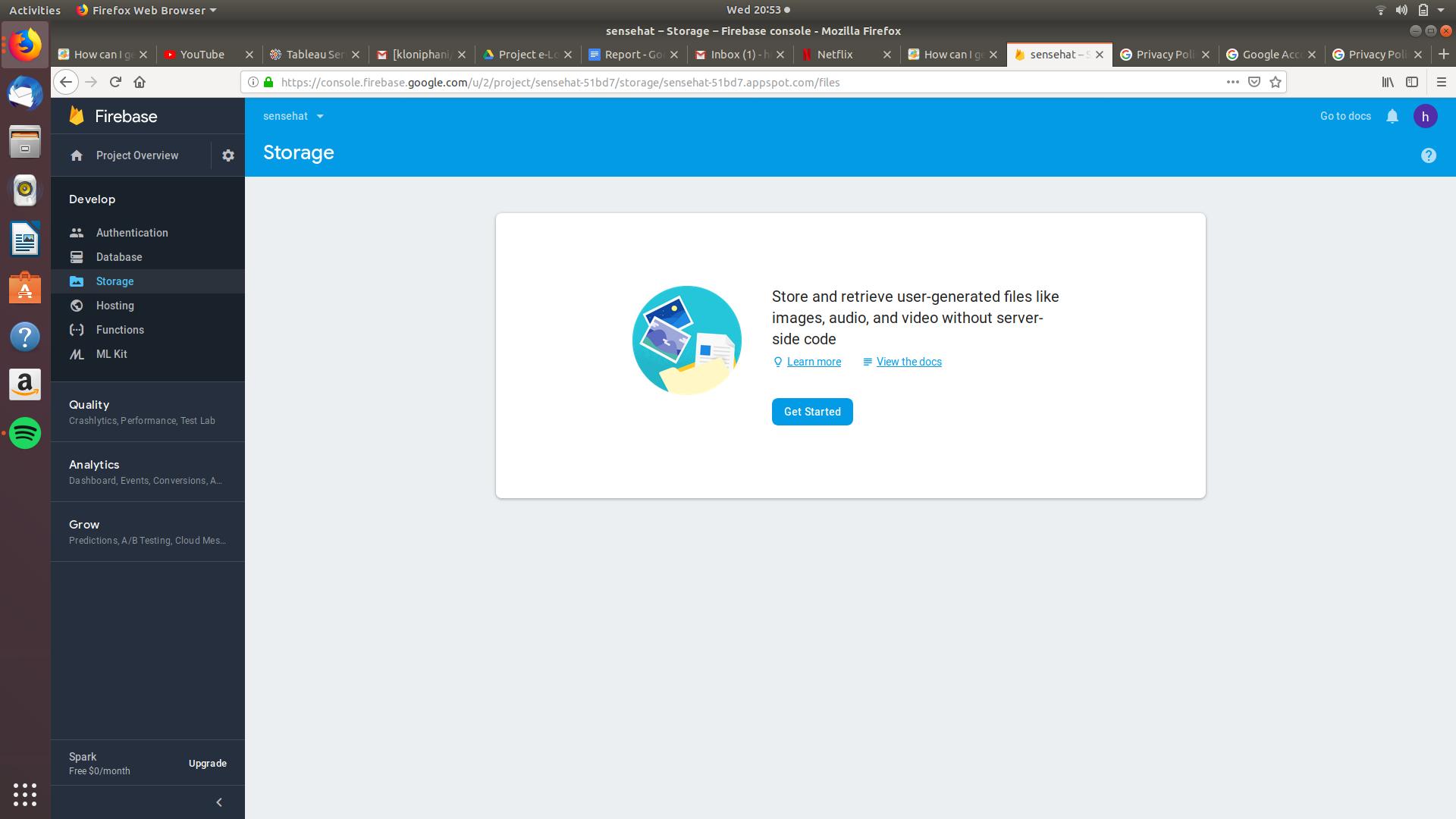
****

**Step 6:** Click on the **Databases** tab on the left.

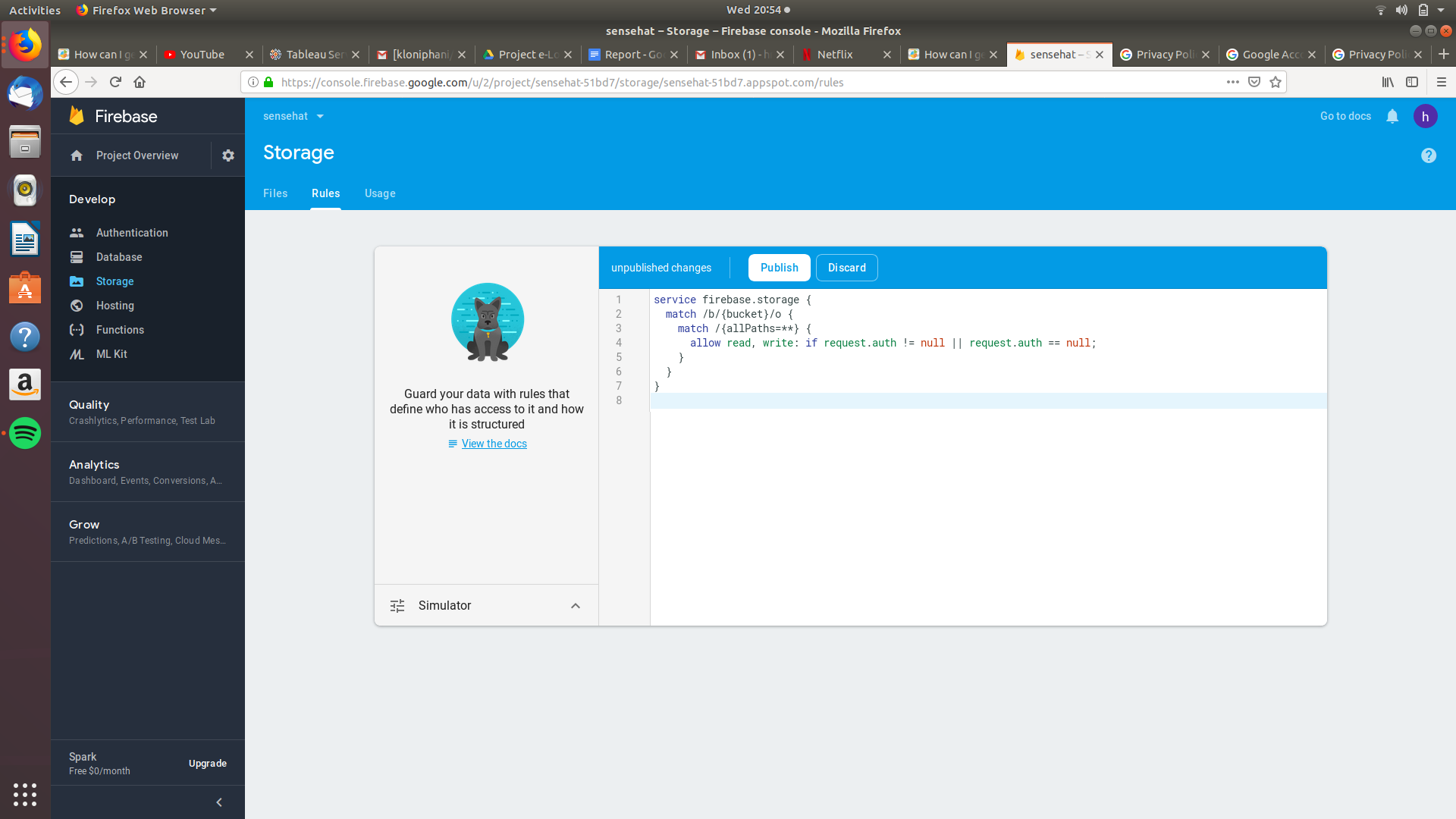


**Step 7:** Click on the **Rules** and set them to what is shown in the image below.

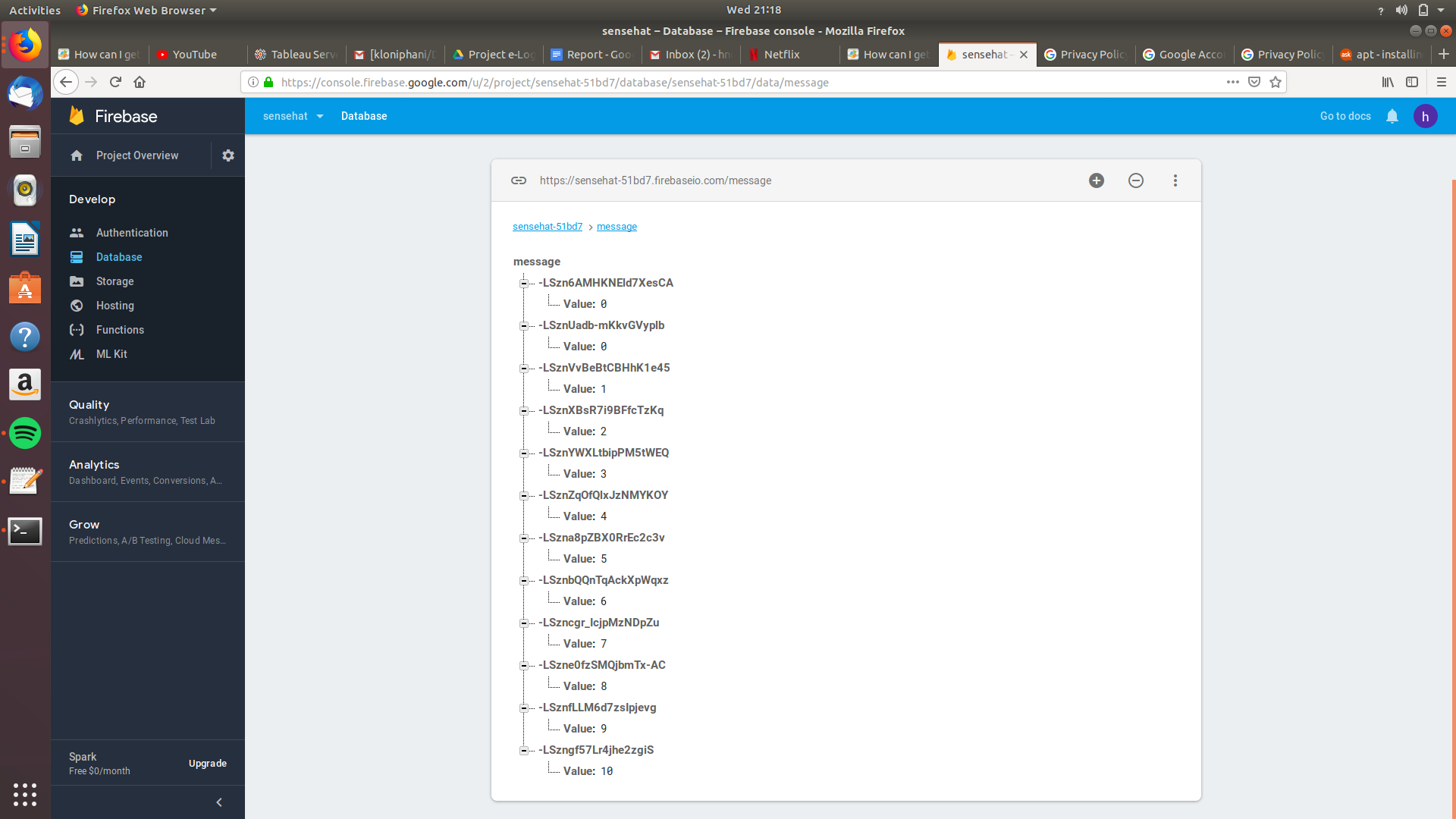


**Step 8:** Click on **Storage** on the left tab.

**Step 9:** Change the rules to what is shown in the picture. Add the “ || request.auth ==null ” on line 4 as shown in the picture.



**Step 10:** After linking your app with your database and pushing data to it. Click on **Databases** on the left tab to view the data.



## Reading and Uploading Sensor Data

## Web interface

## Android Application

Code to add

<script src="http://js.api.here.com/v3/3.0/mapsjs-core.js"  
 type="text/javascript" charset="utf-8"></script>  
 <script src="http://js.api.here.com/v3/3.0/mapsjs-service.js"  
 type="text/javascript" charset="utf-8"></script>

<!--Responsive scripts: JQuery, Popper, Bootstrap-->  
 <script type="text/javascript" src="../bower\_components/jQuery/dist/jquery.slim.js"></script>  
 <script type="text/javascript" src="../bower\_components/popper.js/dist/popper.min.js"></script>  
 <script type="text/javascript" src="../bower\_components/bootstrap/dist/js/bootstrap.min.js"></script>

Errors

<meta http-equiv="Content-Security-Policy" content="default-src gap://ready file://\* \*; style-src 'self' http://\* https://\* 'unsafe-inline'; script-src 'self' http://\* https://\* 'unsafe-inline' 'unsafe-eval'">

## Storing data in InfluxDB and visualising with Grafana

**Installing InfluxDB on Raspberry PI**

* Adding repositories

|  |
| --- |
| curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -source /etc/os-release  test $VERSION\_ID = "7" && echo "deb https://repos.influxdata.com/debian wheezy stable" | sudo tee /etc/apt/sources.list.d/influxdb.list  test $VERSION\_ID = "8" && echo "deb https://repos.influxdata.com/debian jessie stable" | sudo tee /etc/apt/sources.list.d/influxdb.list  test $VERSION\_ID = "8" && echo "deb https://repos.influxdata.com/debian stretch stable" | sudo tee /etc/apt/sources.list.d/influxdb.list |

* Install InfluxDB from the added repository

|  |
| --- |
| sudo apt-get update && sudo apt-get install influxdb |

* Start the InfluxDB service

|  |
| --- |
| sudo service influxdb start |

* Test the installation by clicking the link below.

|  |
| --- |
| localhost:8086 |

If the installation was successful, the page will display “**404 page not found**”.

* Try to configure InfluxDB if it doesn’t work. Uncomment all the lines in the **[Admin]** section

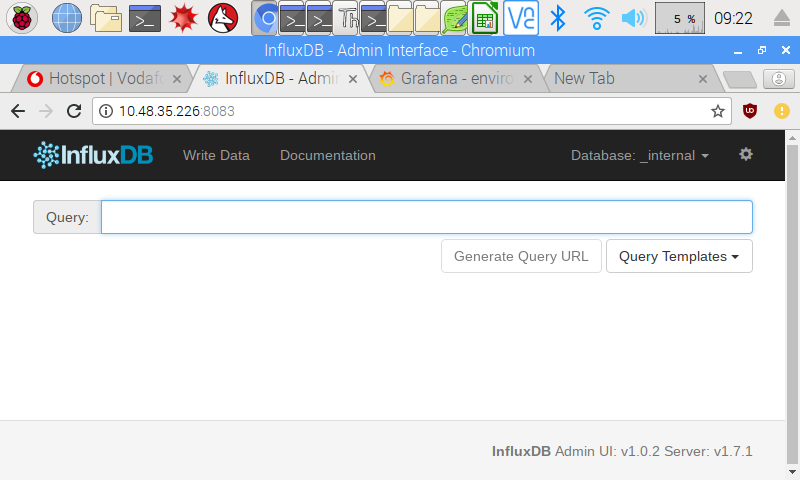
by removing all the “#” .

|  |
| --- |
| sudo nano /etc/influxdb/influxdb.conf |

* Test the Web UI by clicking on the link below.

|  |
| --- |
| localhost:8083 |

The page below is supposed to show :



**Installing Grafana on a Raspberry Pi**

###### Install some dependency packages

|  |
| --- |
| sudo apt-get install apt-transport-https curl |

* Add the key

|  |
| --- |
| curl https://bintray.com/user/downloadSubjectPublicKey?username=bintray | sudo apt-key add - |

* Add Grafana to the update list

|  |
| --- |
| echo "deb https://dl.bintray.com/fg2it/deb stretch main" | sudo tee -a /etc/apt/sources.list.d/grafana.list |

* Install Grafana

|  |
| --- |
| sudo apt-get install grafana |

* Start the Grafana server service

|  |
| --- |
| sudo systemctl start grafana-server |

* Set Grafana to run on startup

|  |
| --- |
| sudo systemctl enable grafana-server.service |

* Click the link below to login to Grafana

|  |
| --- |
| localhost:3000 |

**Creating a database in InfluxDB and visualising the data using Grafana**

* To create a database in InfluxDB type the following on the terminal. This will open the influxDB shell which accepts database queries (Influx Query Language).

|  |
| --- |
| $ influx -precision rfc3339  Connected to http://localhost:8086 version 1.2.x  InfluxDB shell 1.2.x  > |

* Type the following command to create a database named **sense\_Hat.**

|  |
| --- |
| > CREATE DATABASE senseHatDB  > |

* To check if the database was created successfully type the following query.

|  |
| --- |
| > SHOW DATABASES  name: databases  ---------------  name  \_internal  senseHatDB |

* Install the InfluxDB Python module.

|  |
| --- |
| sudo pip install influxdb |

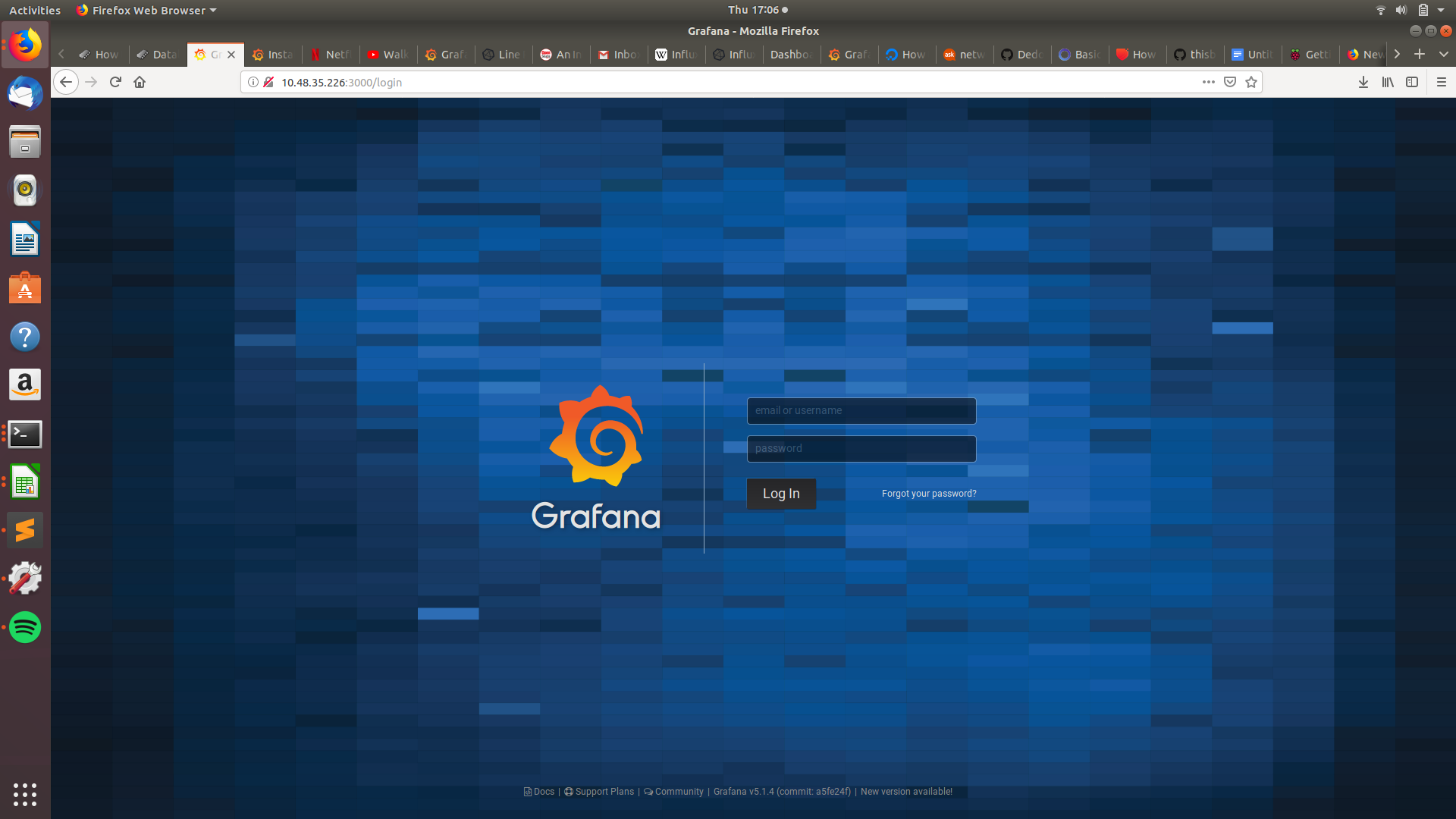
* Run the python script which reads in sensor readings and pushes them to the database created by typing the following on the Raspberry Pi terminal.

|  |
| --- |
| python sense\_influxDB.py -db=dedomena2 -sn=test1 |

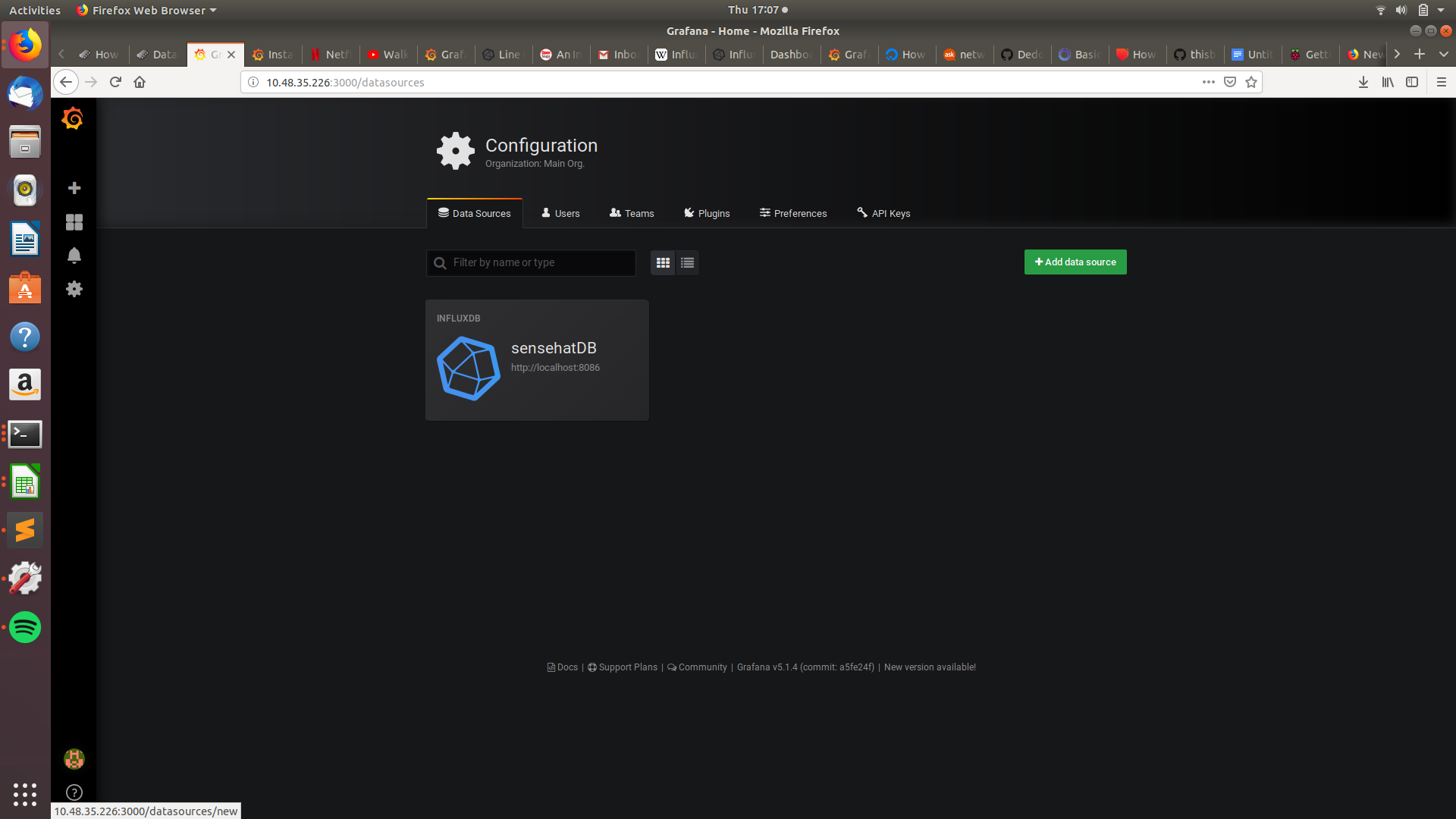
The python script is on GitHub in the Dedomena/Sensor folder, link : <https://github.com/kloniphani/Dedomena/blob/master/Sensor/sense_influxDB.py>

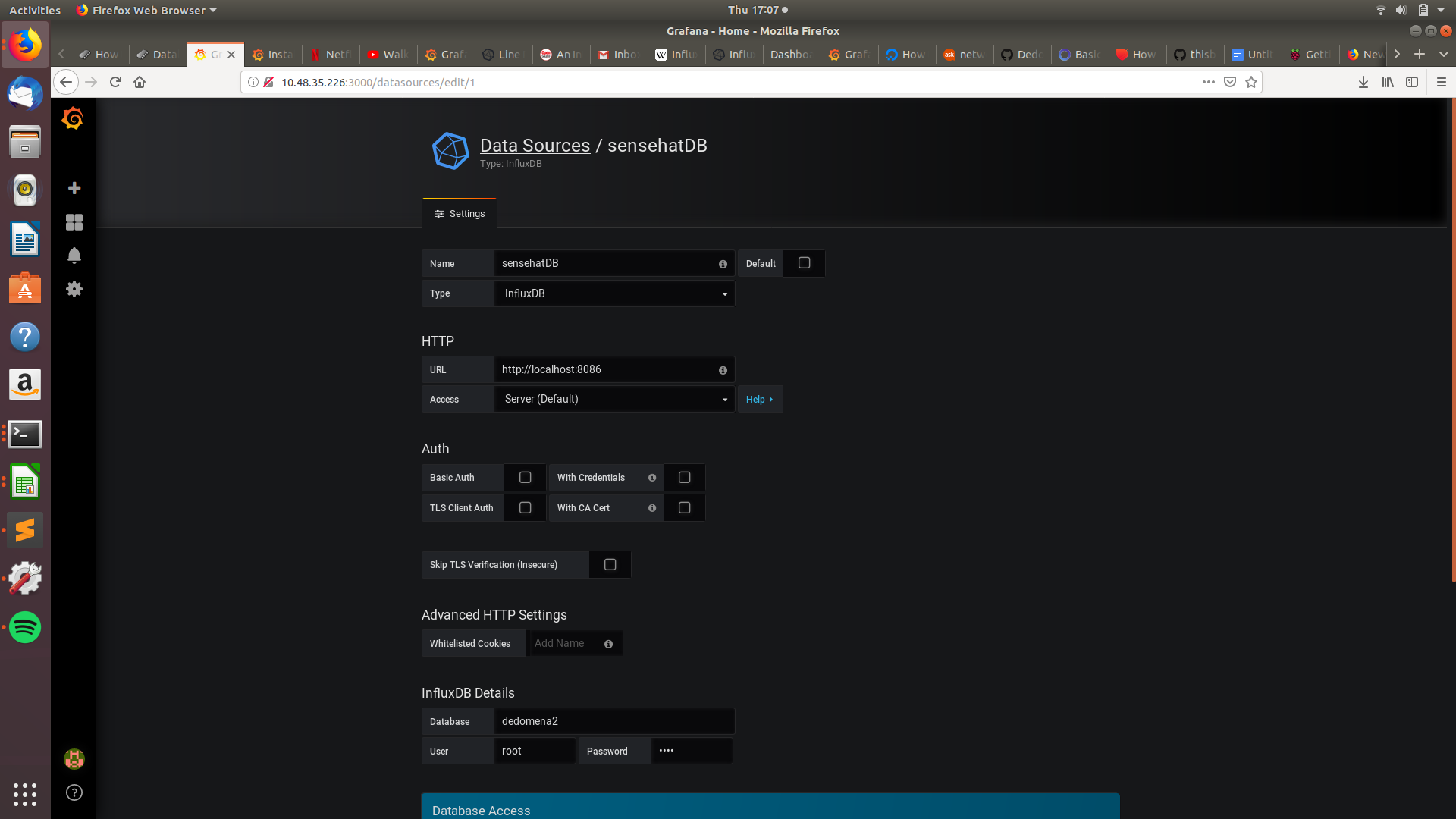
* Login to Grafana at localhost:3000

(username= “admin”, password=”admin”)



* Navigate to Datasource->Add New and fill the form as below



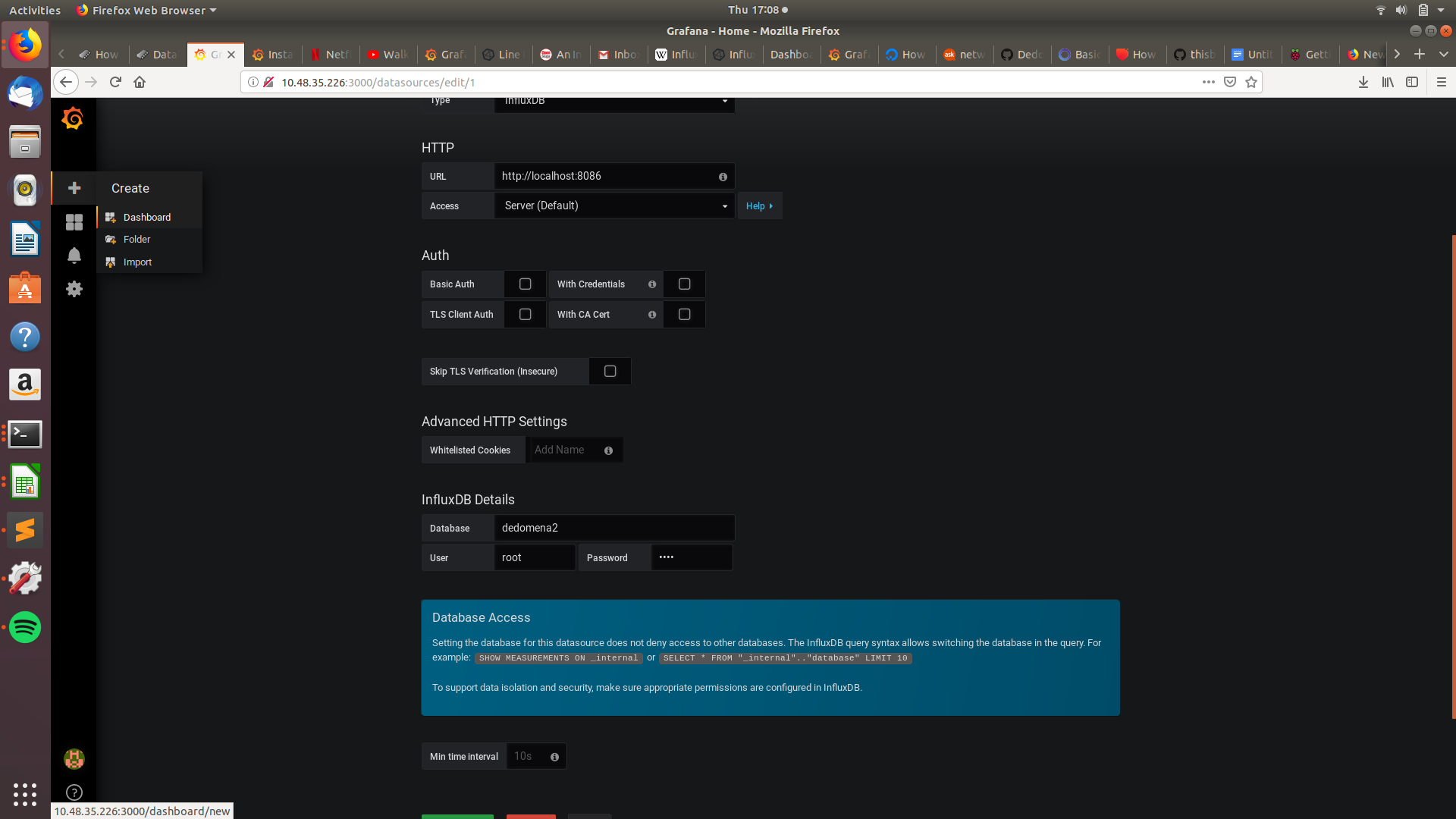


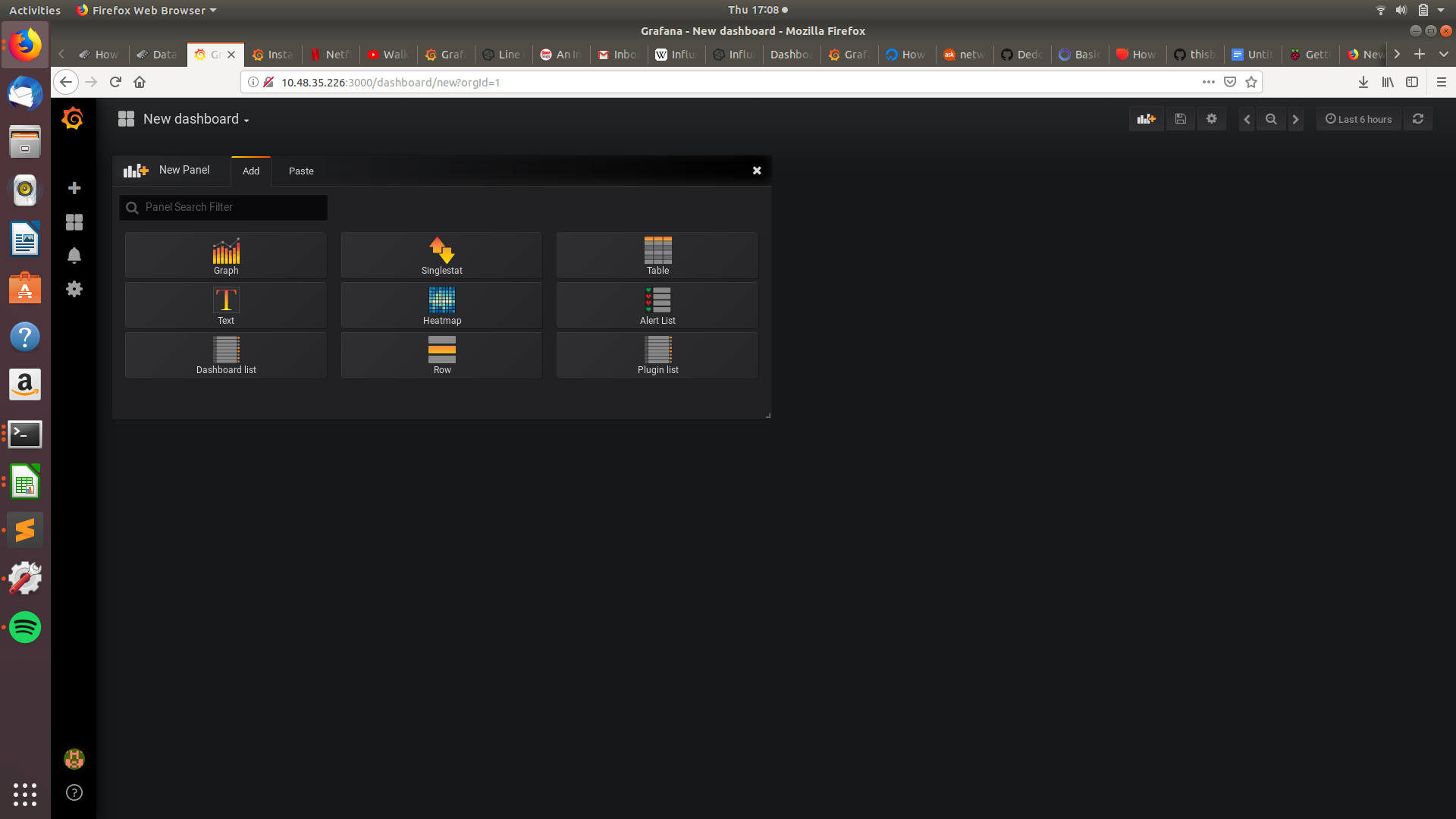
The influxDB details of the database we created are :

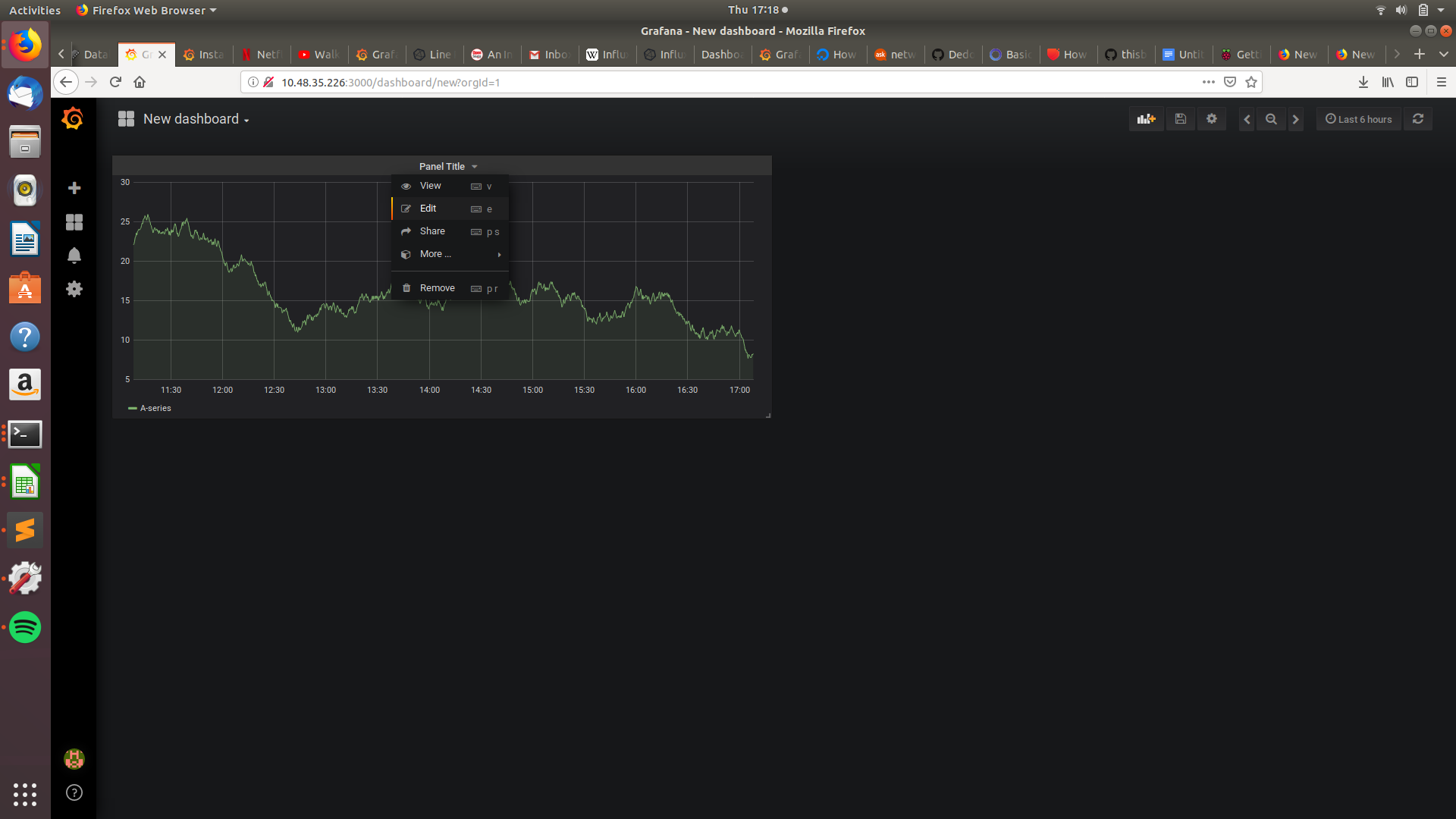
User: root

Password: root

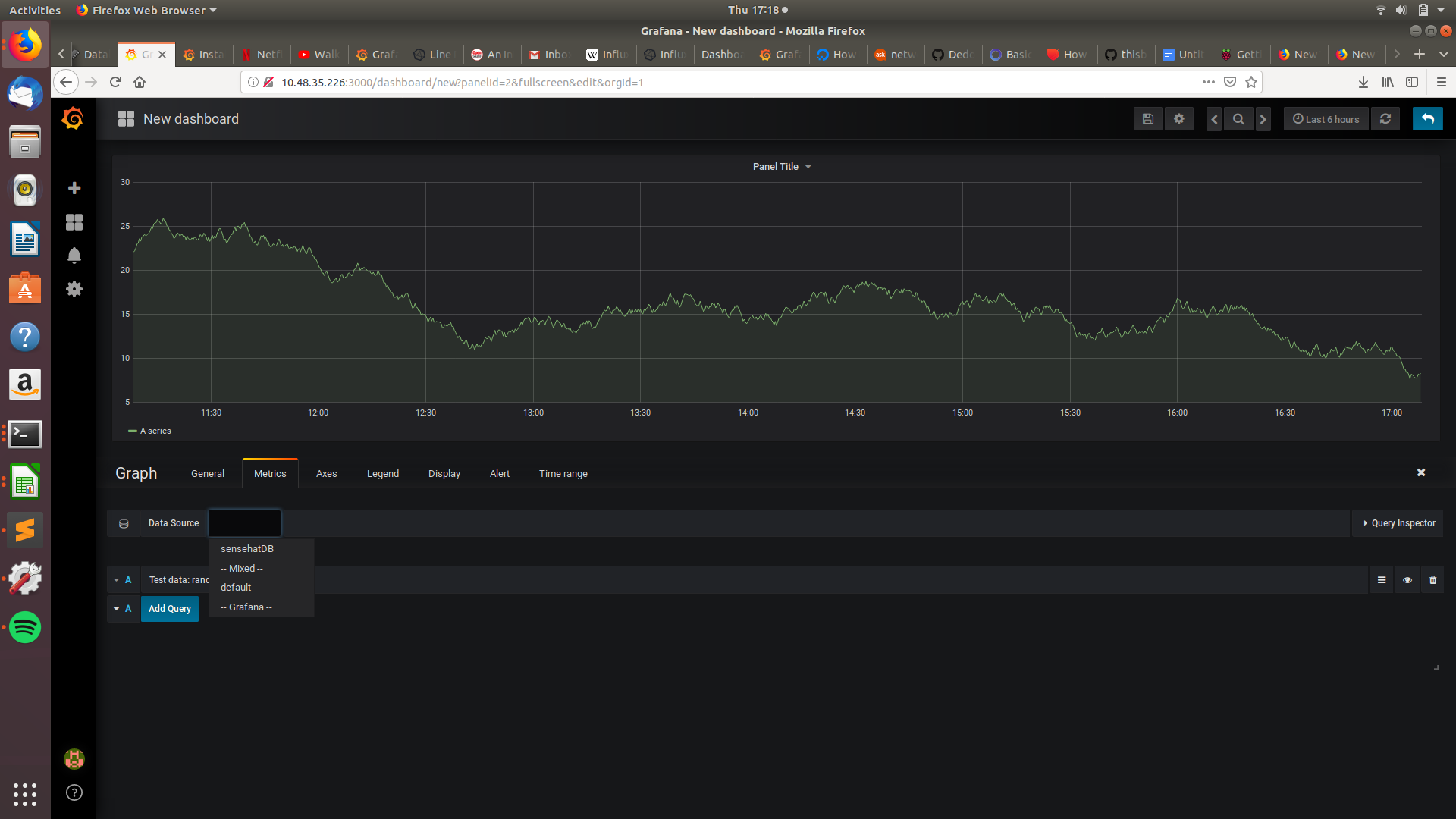
* Create a new dashboard and add graphs to it as shown below



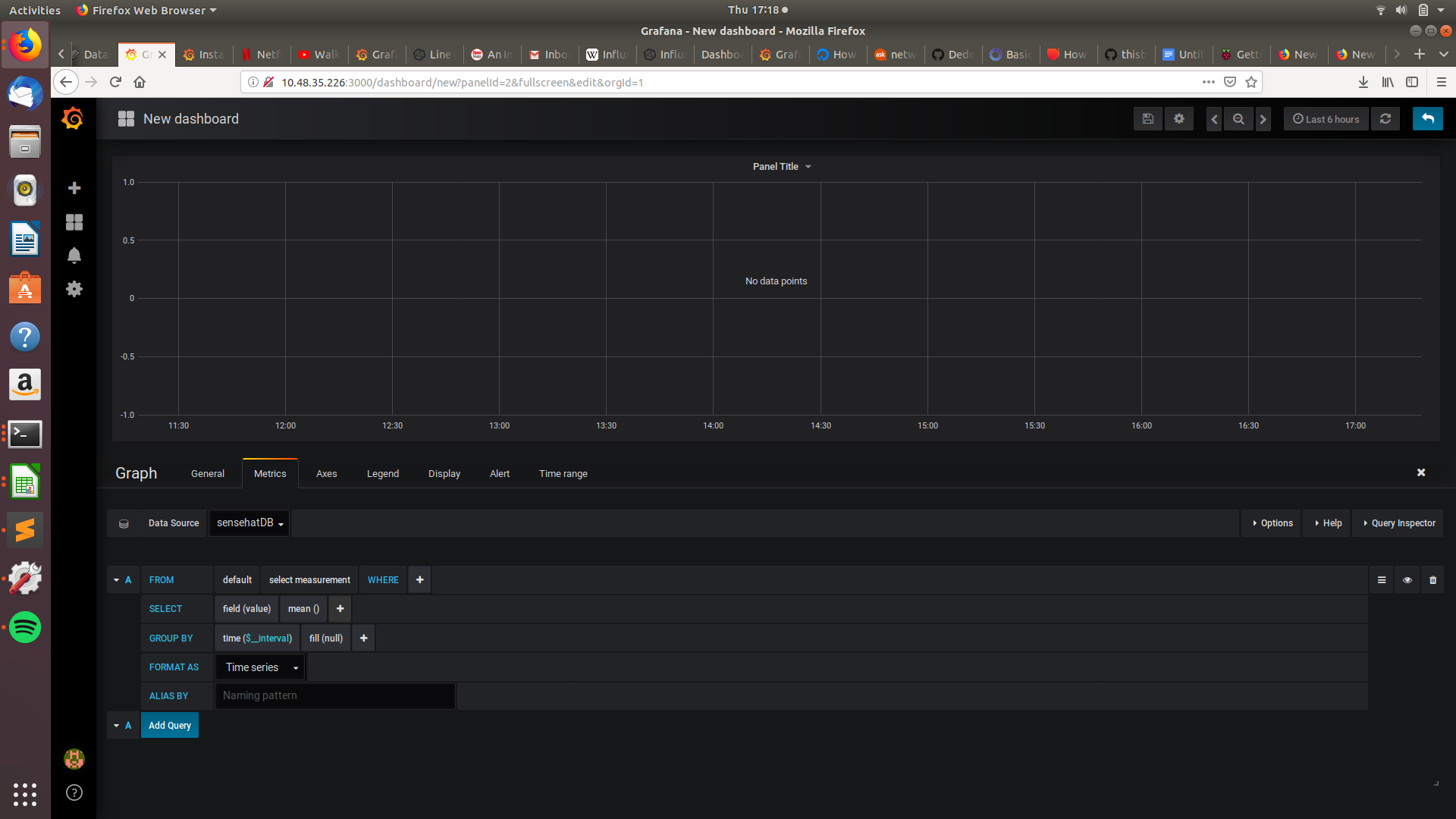




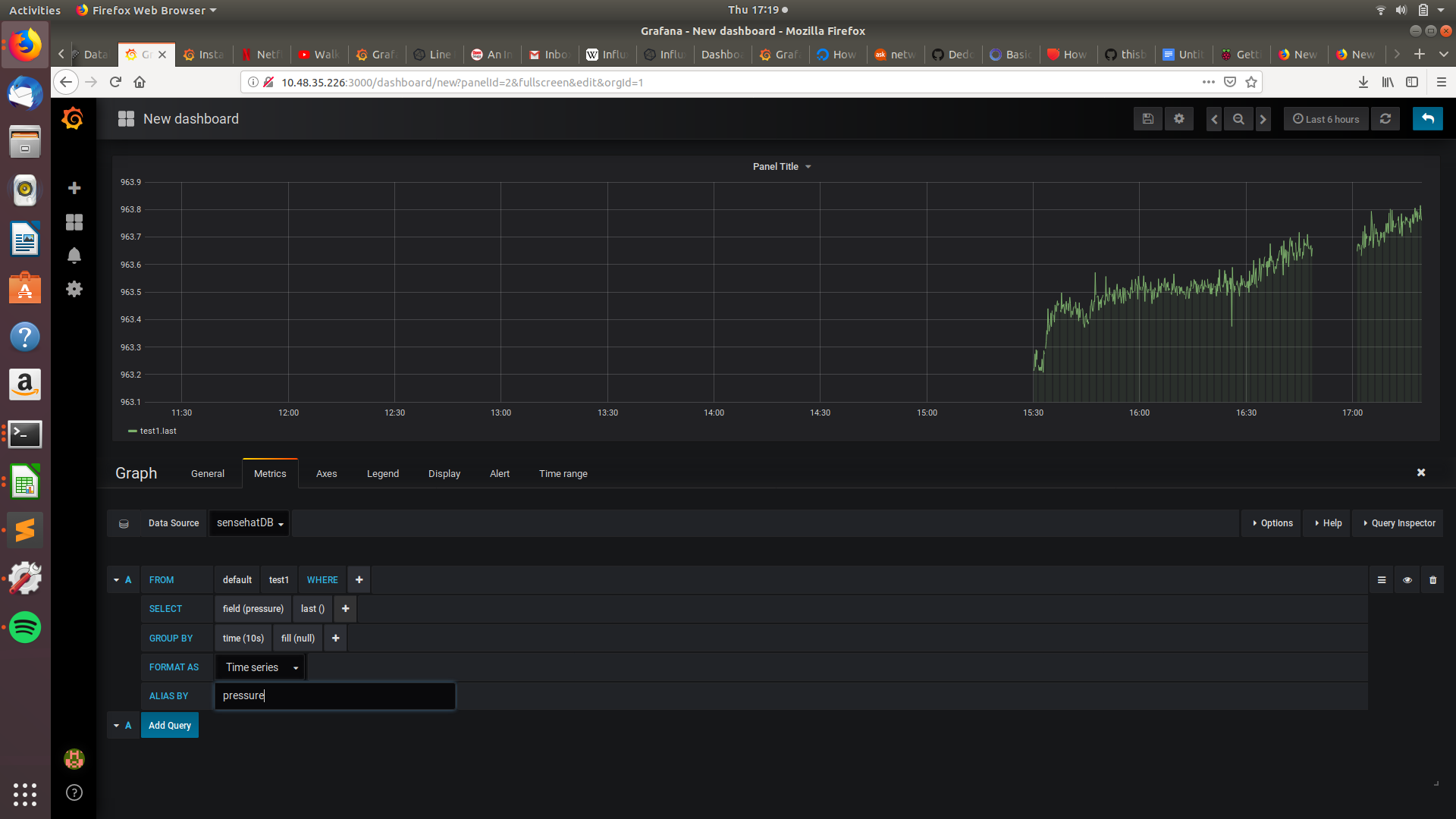
Click on the **edit** option



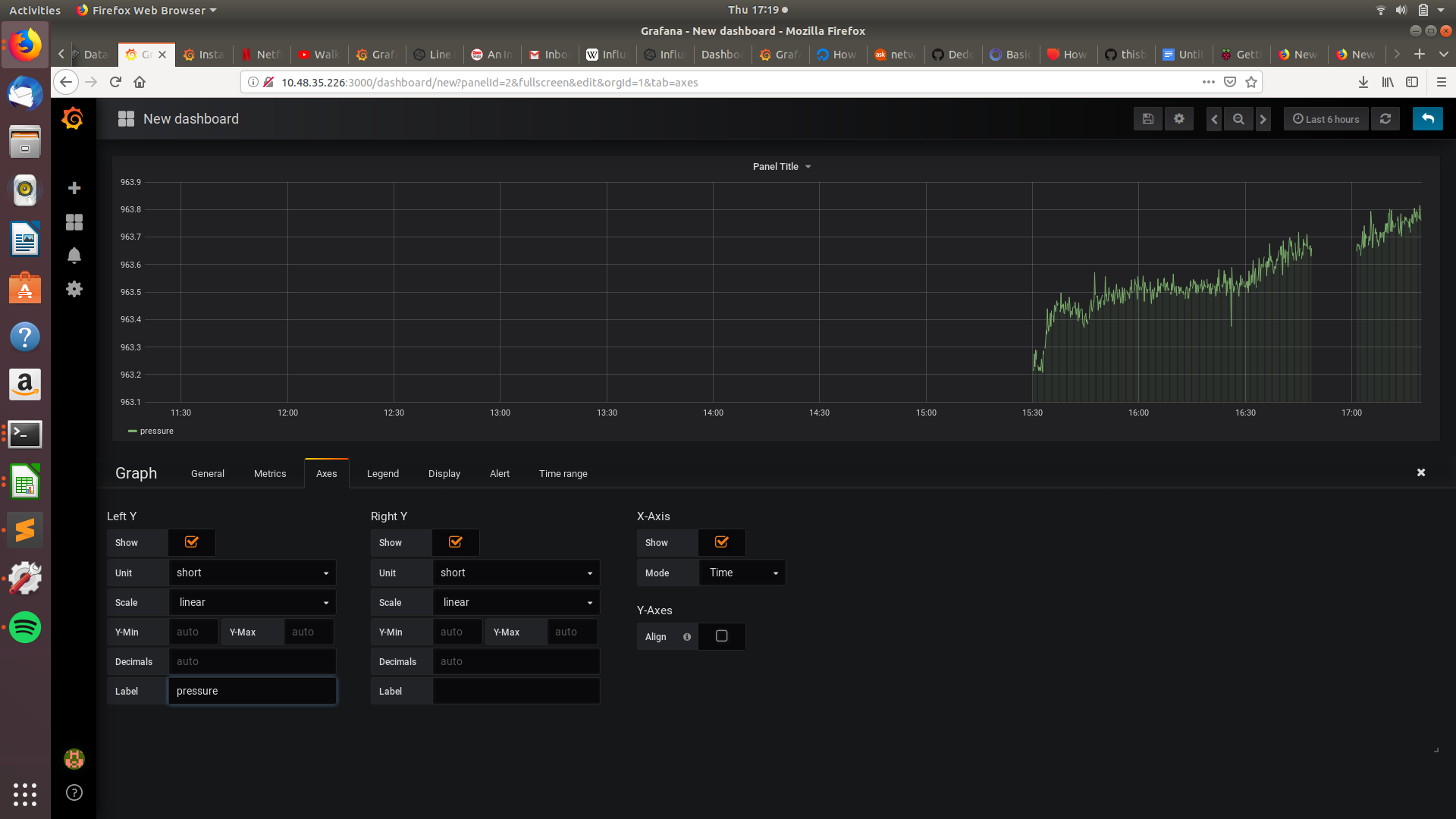
Select the data source as **senseHatDB**

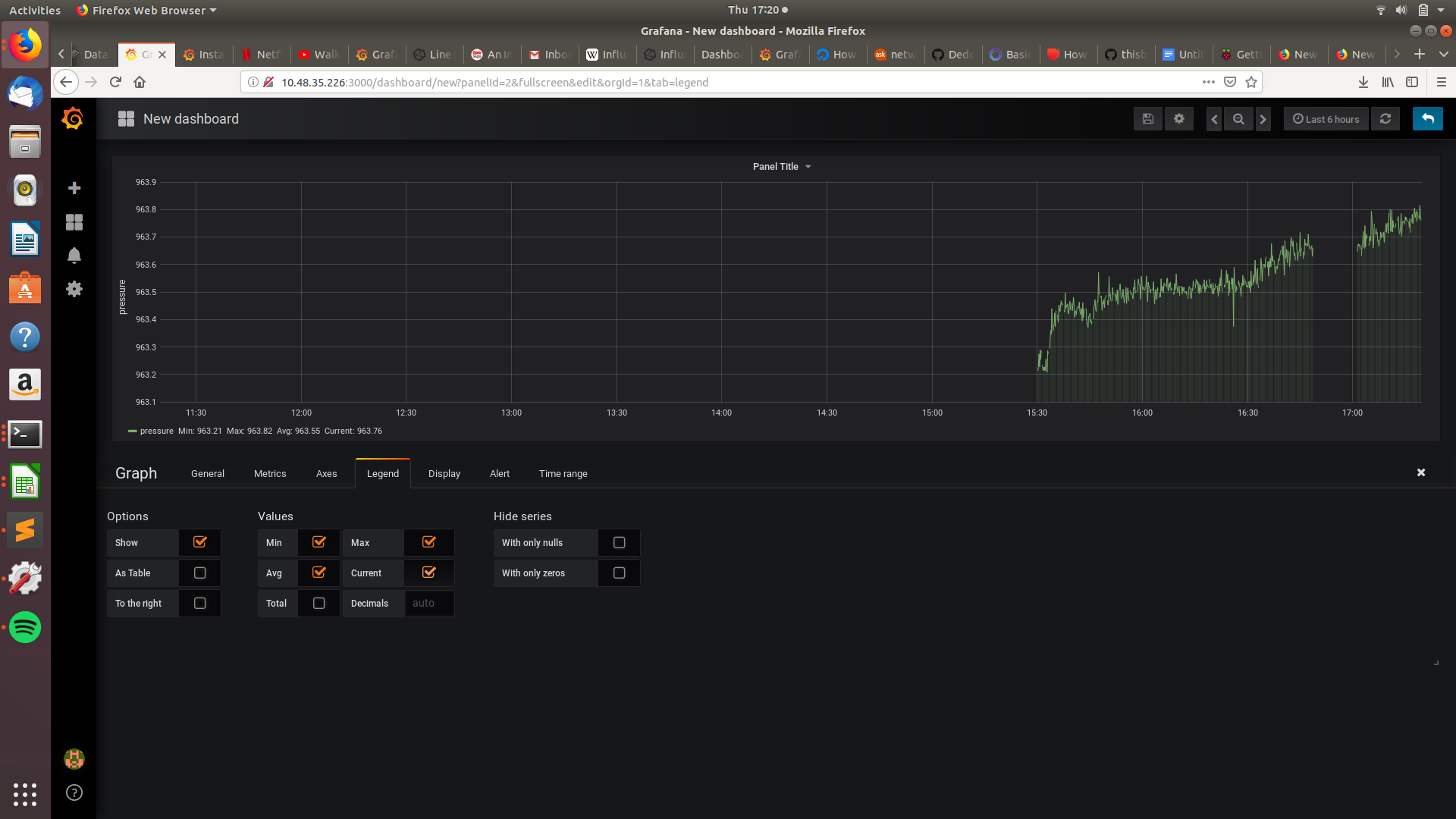
****

Create a query which will show data on the graph as shown below

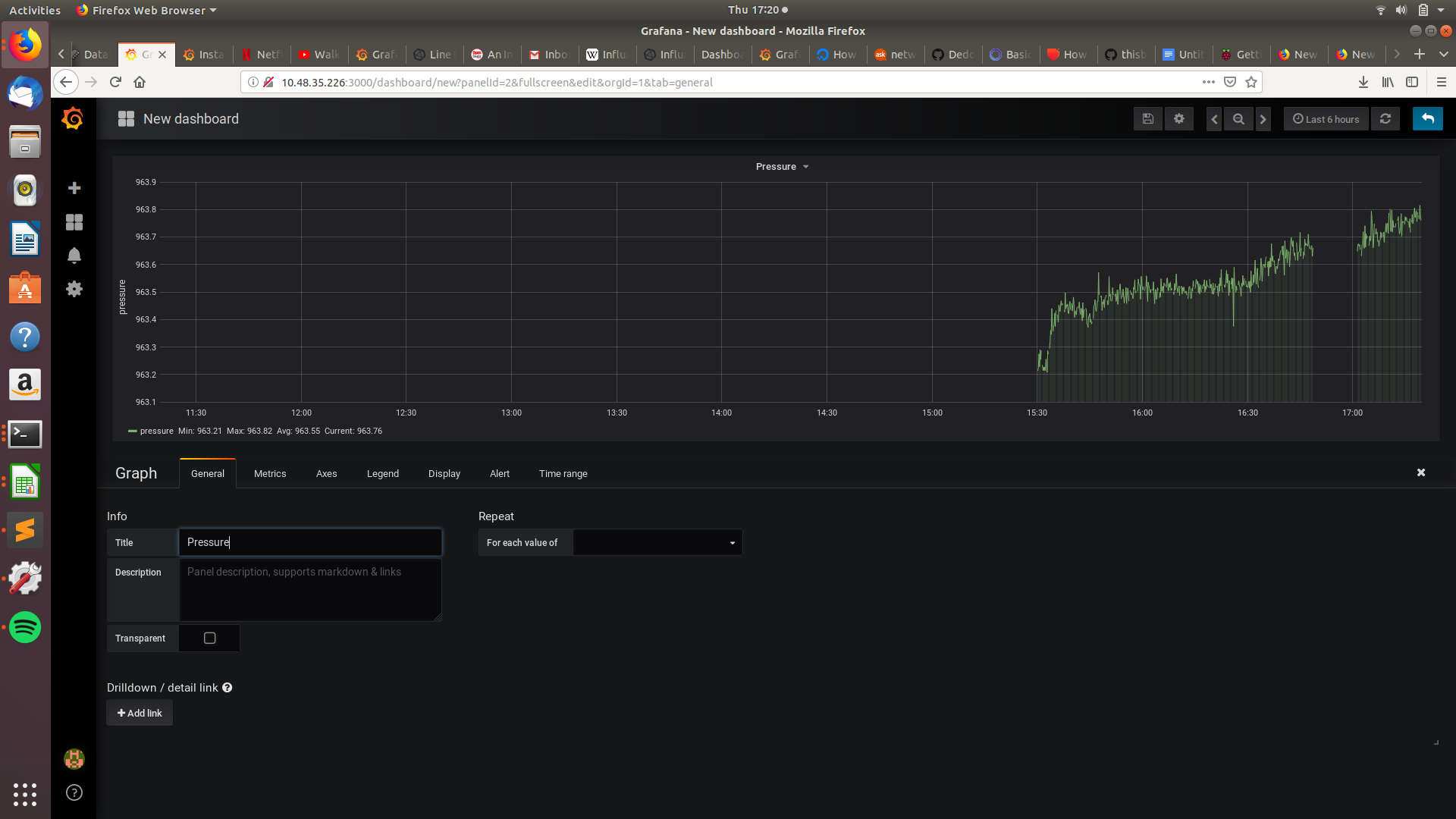


You can now visualise the data on the graph, you can now label the axes and add legends as shown below

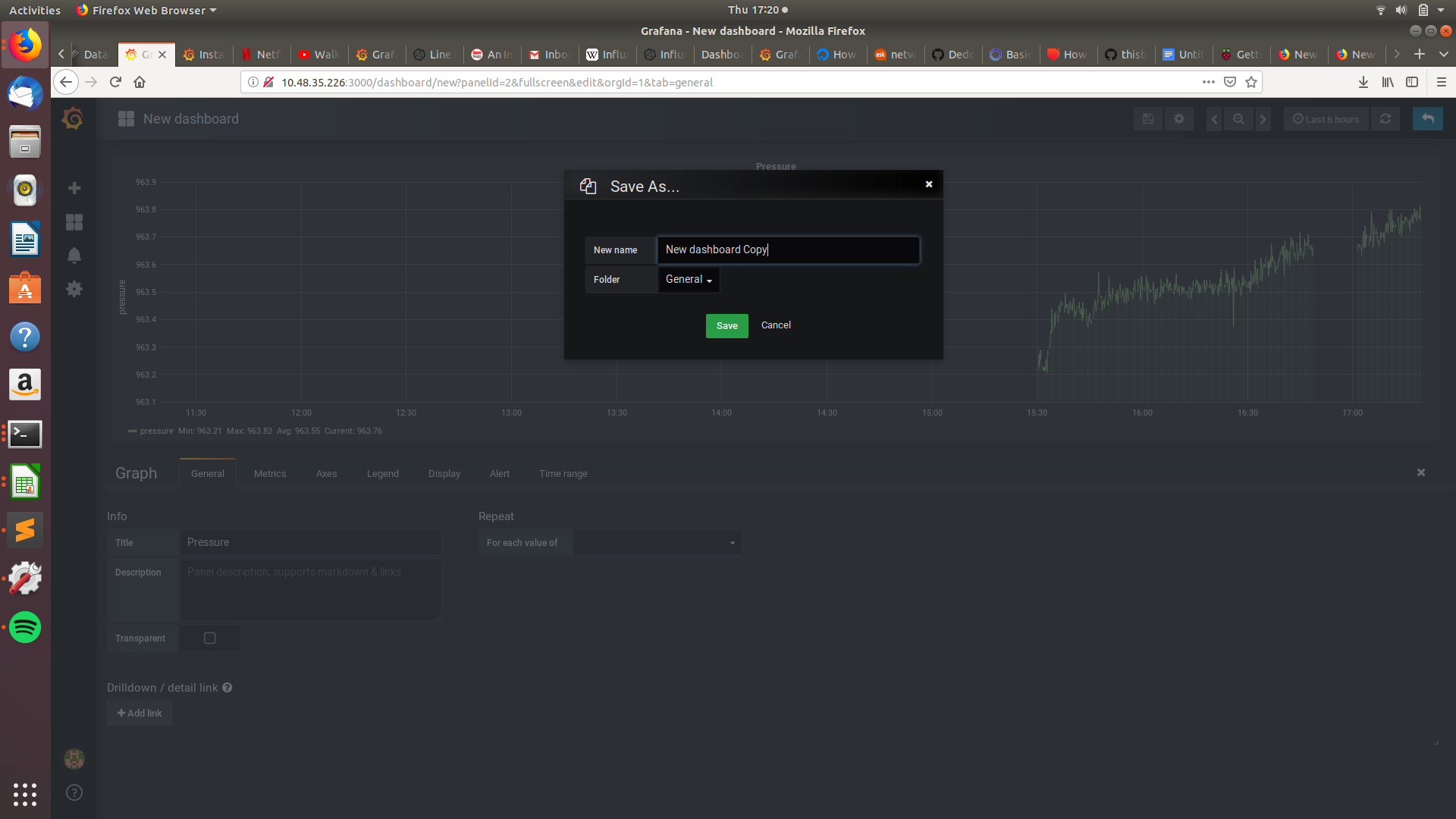




Add the max, min, avg and cur to your legends as shown above.

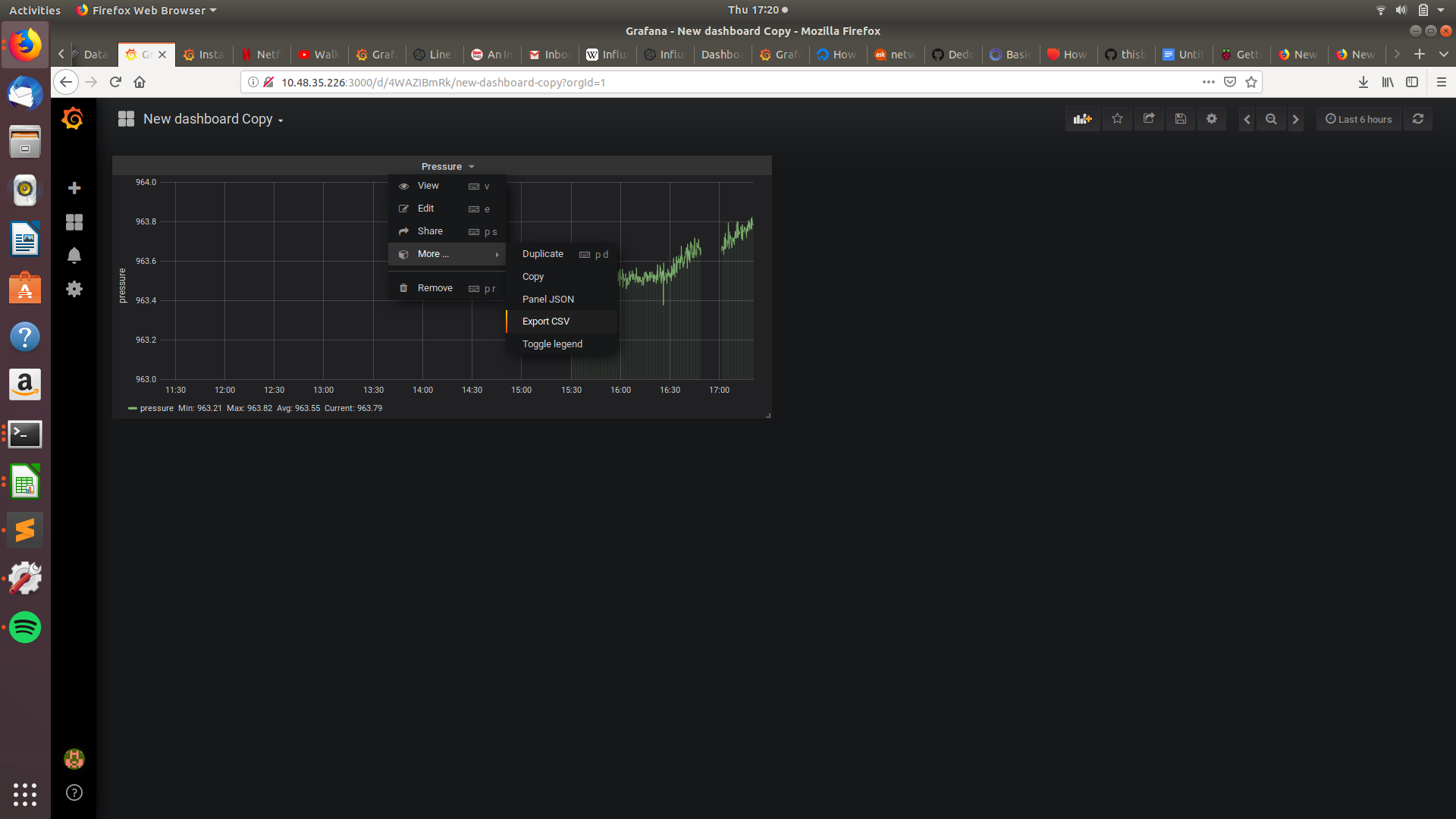


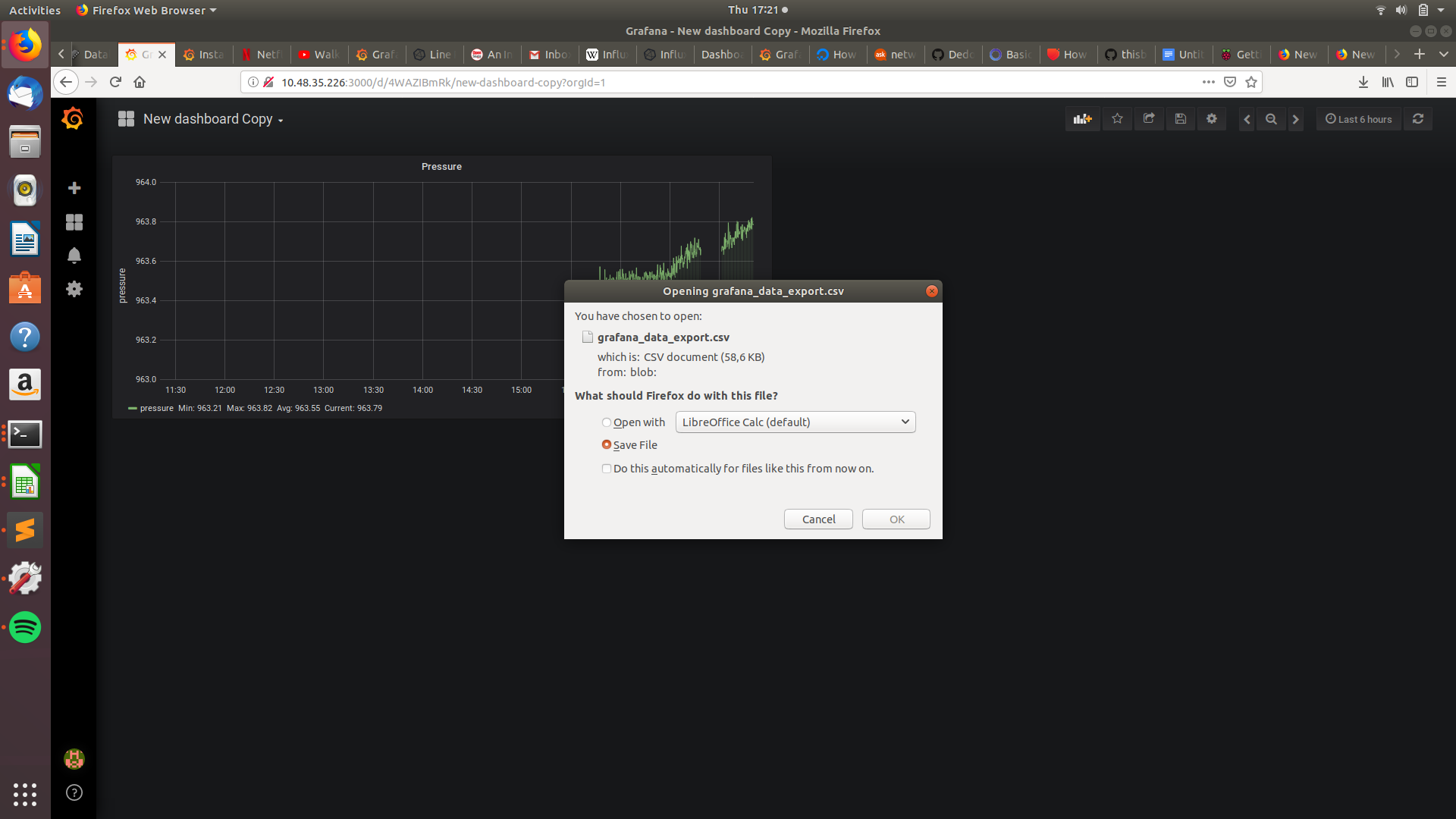
Add a **Title** to the graph as shown above.



Save the **Dashboard** when you’re done.

**Exporting the data as a CSV file**

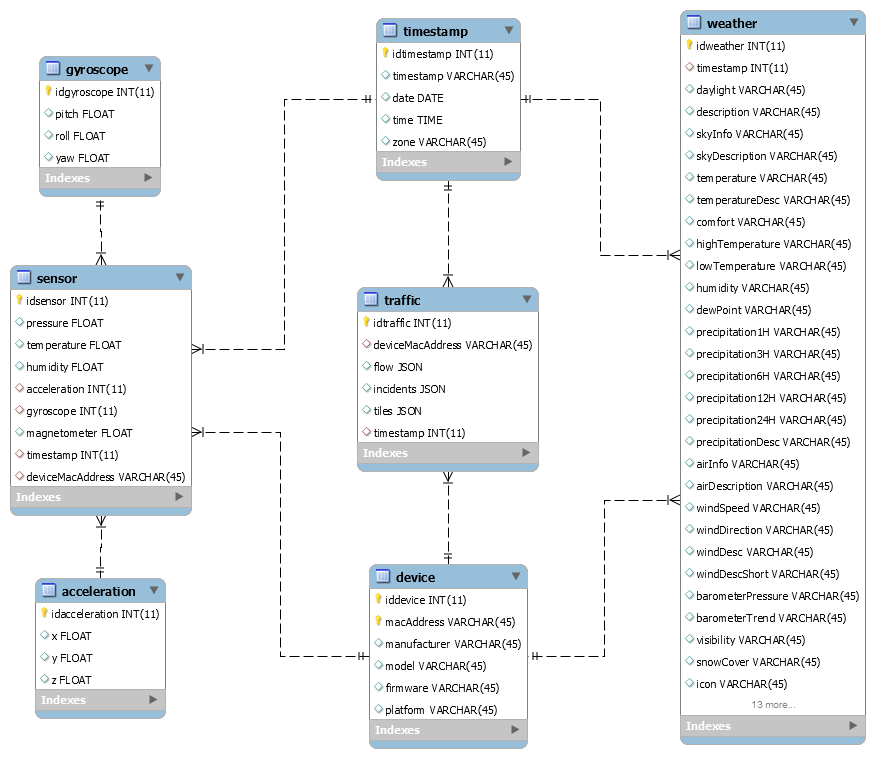
****

****

Save the CSV file.

# RESULTS

## Database



# CONCLUSION

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

# REFERENCES

1. <https://pythonhosted.org/sense-hat/>
2. <https://www.db4free.net/signup.php>
3. <https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.5/bk_command-line-installation/content/install_hue_package.html>