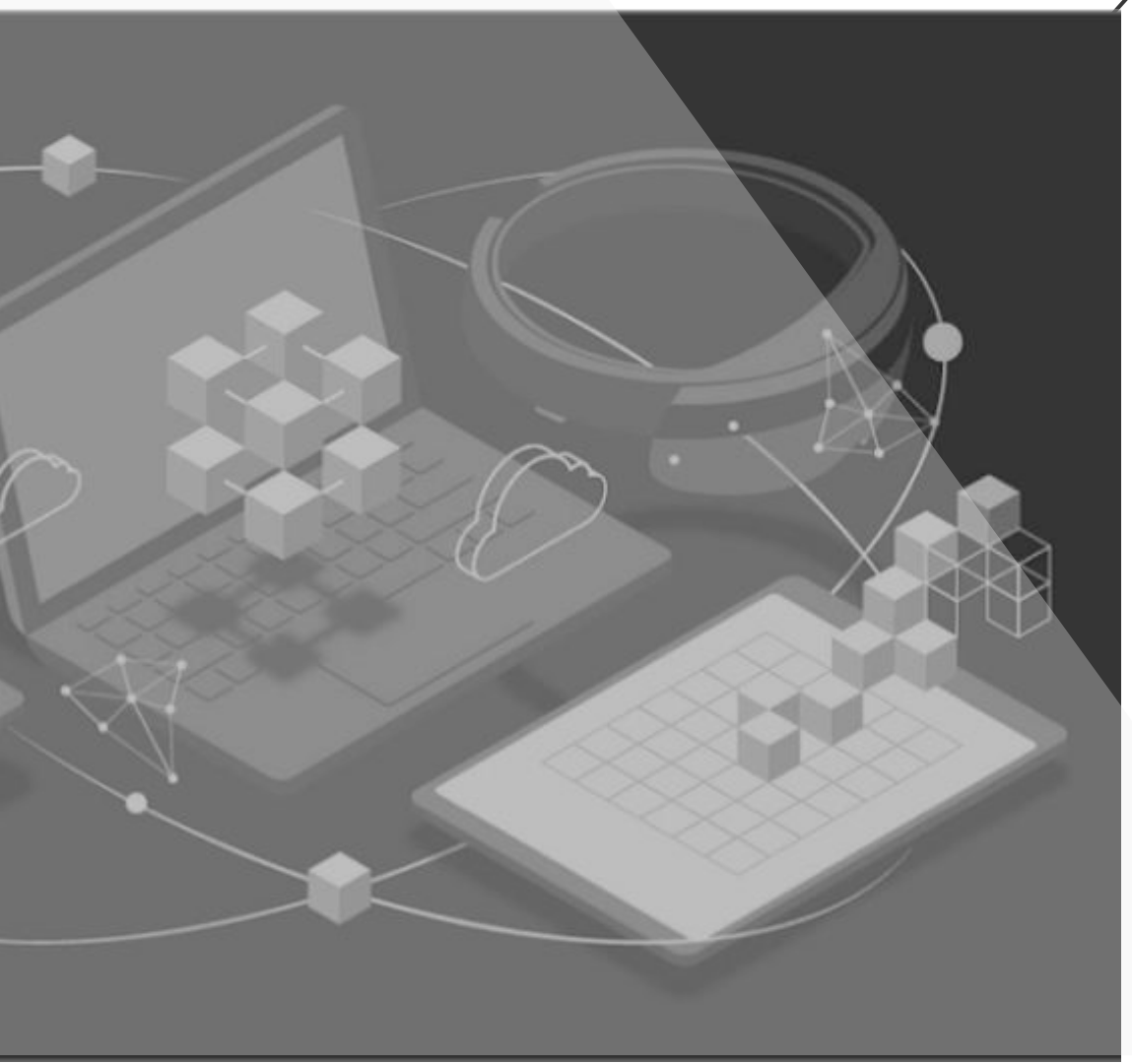
# Backend dan frontend IOT dashboard

Nodered influxdb backend dengan graphana

# Table of contents



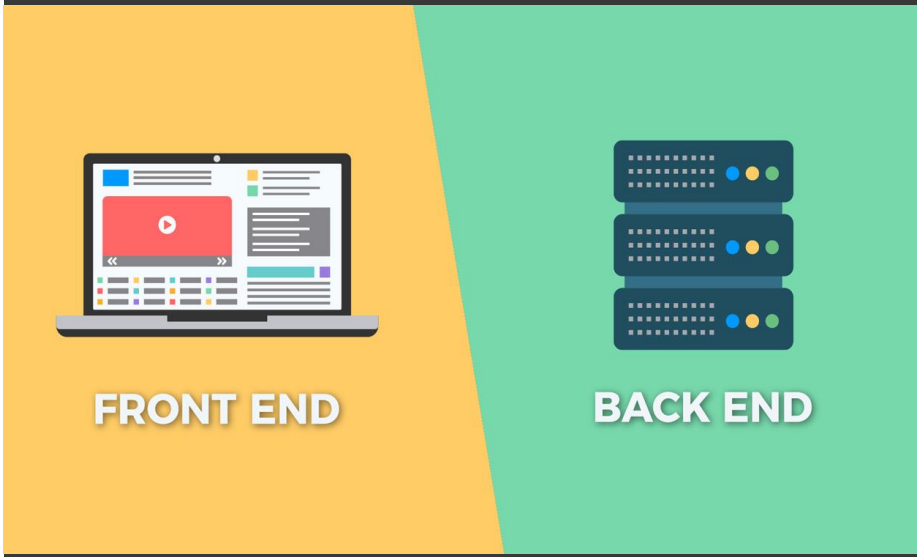
# 01

# Pengenalan

# Goal

- Memahami sistem front end, back end, and full-stack development.
- Menggunakan database sqlite dan influxdb
- Menggunakan fungsi pada node-red
- Menampilkan data time line pada dashboard grafana

# FRONT END, BACK END, AND FULL-STACK DEVELOPMENT.

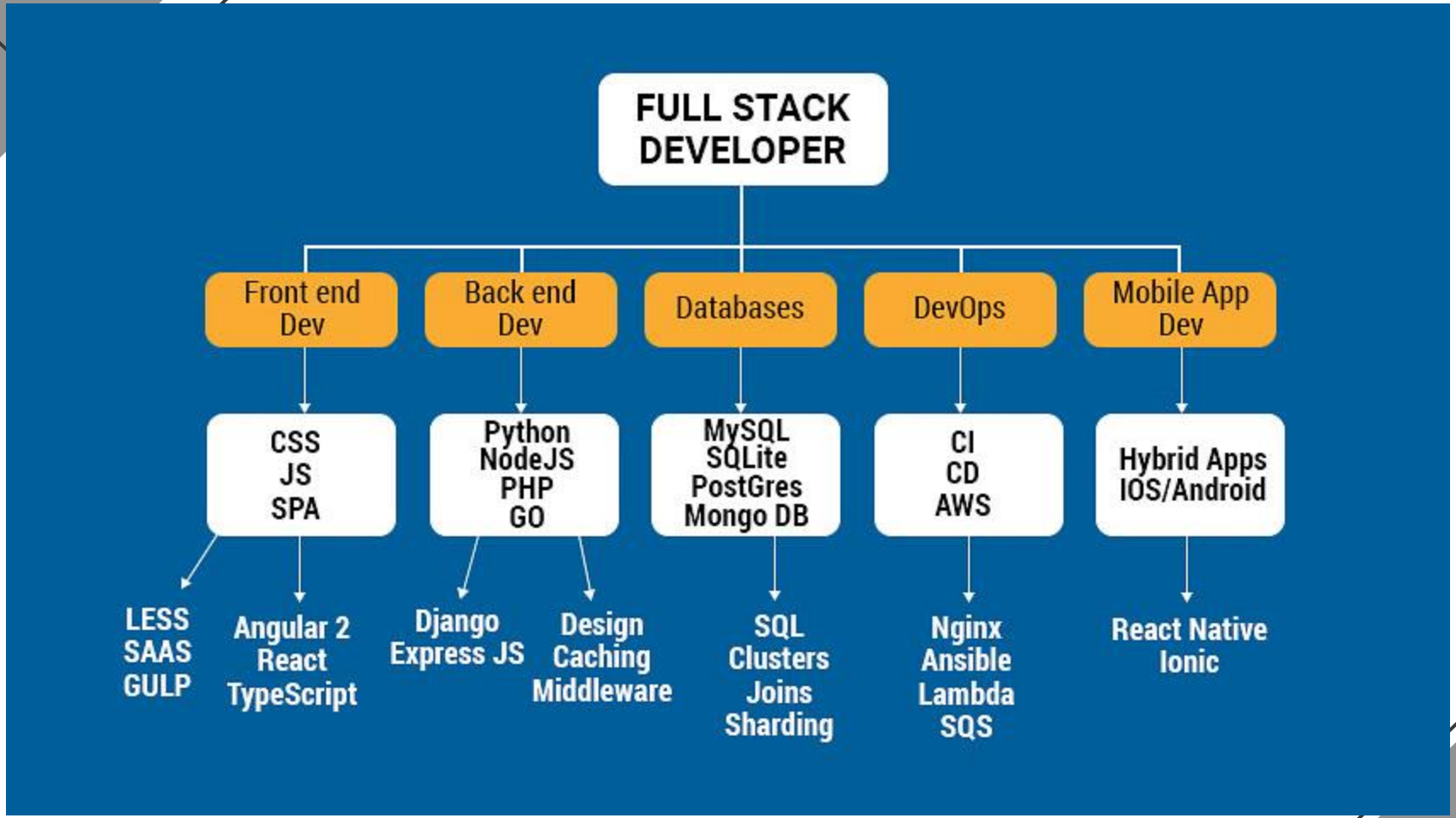


# FRONT END

1.Memastikan kelayakan teknis desain dari UI/UX
2.Mengoptimalkan kecepatan dari *website*/aplikasi
3.Menerapkan elemen visual situs web atau aplikasi
4.Menggunakan desain responsif dalam pembuatan antarmuka pengguna situs web atau aplikasi
5.Menguji situs web atau aplikasi untuk kegunaan
6.Memecahkan masalah kode apa pun yang tidak berfungsi
7.Meningkatkan arsitektur visual situs web atau aplikasi
8.Memastikan bahwa semua kelayakan dari tampilan *website* sebelum mengirimkannya ke tim back end
9.Berkolaborasi tim lainnya

# Back end

1.Mengelola dan mengembangkan sumber daya API (Application Programming Interface) yang berfungsi di seluruh perangkat
2.Membuat sistem pemrosesan pembayaran yang menyimpan data yang diperlukan dengan aman
3.Selalu perbarui aplikasi web, aman, dan cepat
4.Memantau status server
5.Menerapkan algoritma dan memecahkan masalah yang terkait dengan sistem server atau data base
6.Mengembangkan CMS (Content Management System);
7.Mendukung pengembangan frontend dengan komunikasi yang jelas dan dokumentasi yang baik
8.Menyimpan dan mengelola data secara efektif

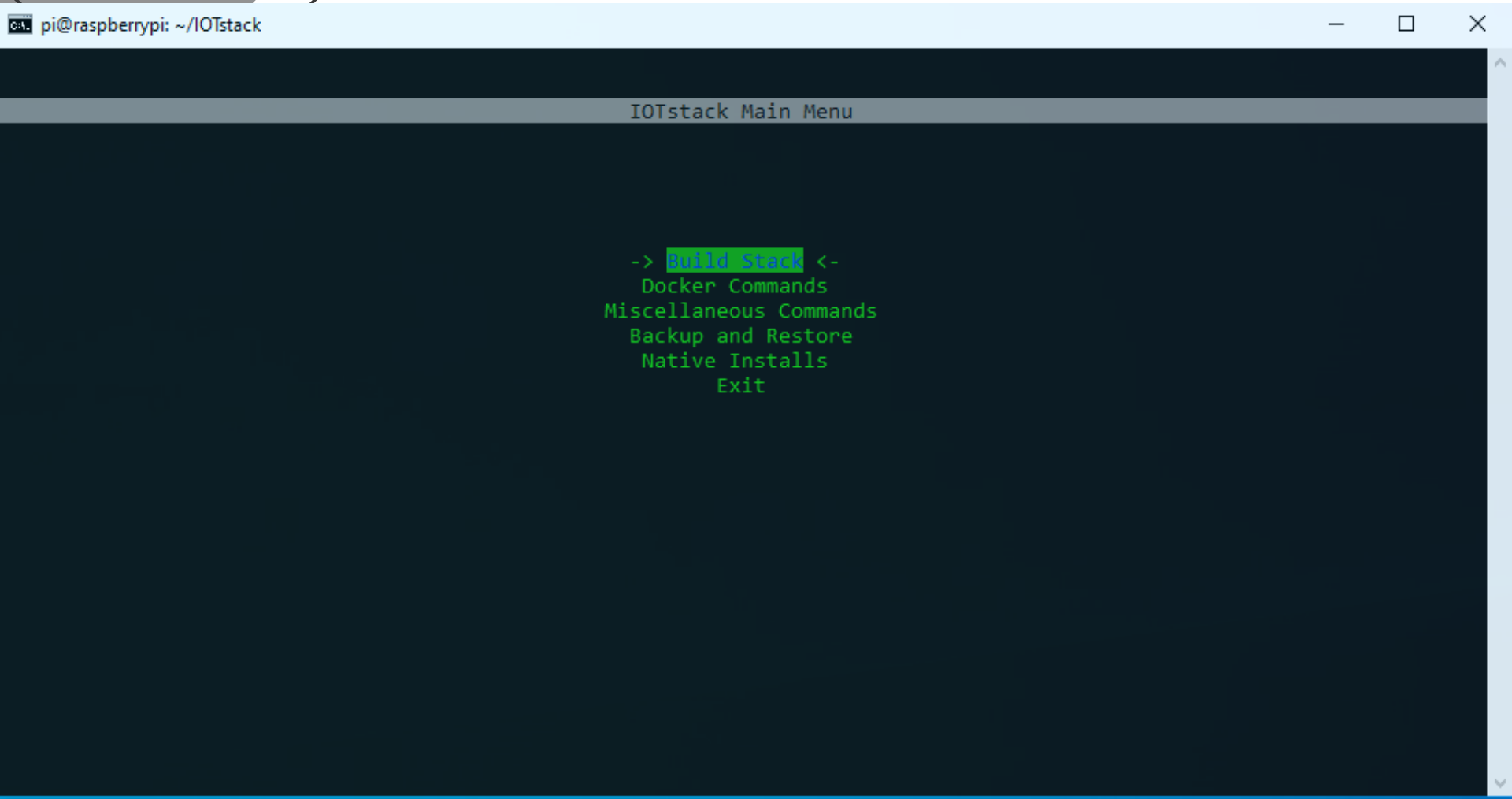

FULL STACK DEVELOPER
Front end Dev
Back end Dev
Databases
DevOps
Mobile App Dev
CSS
JS
SPA
Python
NodeJS
PHP
GO
MySQL
SQLite
PostGres
Mongo DB
CI
CD
AWS
Hybrid Apps
IOS/Android
LESS
SAAS
GULP
Angular 2
React
TypeScript
Django
Express JS
Design
Caching
Middleware
SQL
Clusters
Joins
Sharding
Nginx
Ansible
Lambda
SQS
React Native
Ionic

# INFLUXDB

Node-red untuk query for influxdb

# INFLUXDB

InfluxDB is an open-source high-performance time series database (TSDB) that can store large amounts of data per second. Each data point you submit to the database is associated with a particular timestamp. So, it is ideal for IoT datalogging projects like storing data from your weather station sensors.


pi@raspberrypi: ~/IOTstack
IOTstack Main Menu
-> Build Stack <-
Docker Commands
Miscellaneous Commands
Backup and Restore
Native Installs
Exit

# Service to install

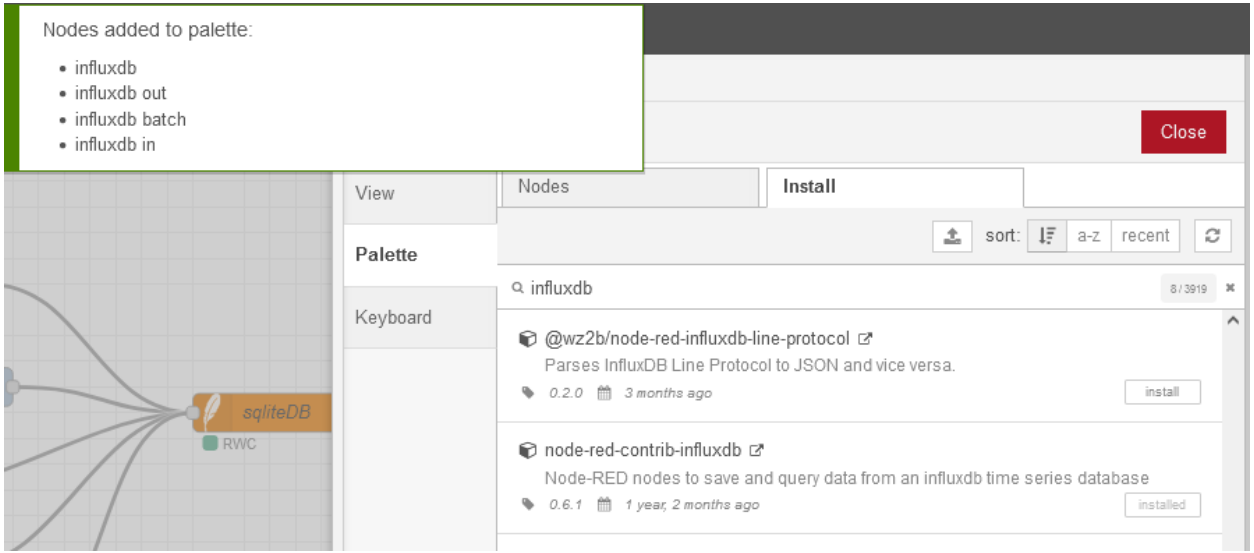
- Nodered
- Portainer-ce
- Mosquito
- Influxdb
- Grafana

# Install windows influxdb (service)

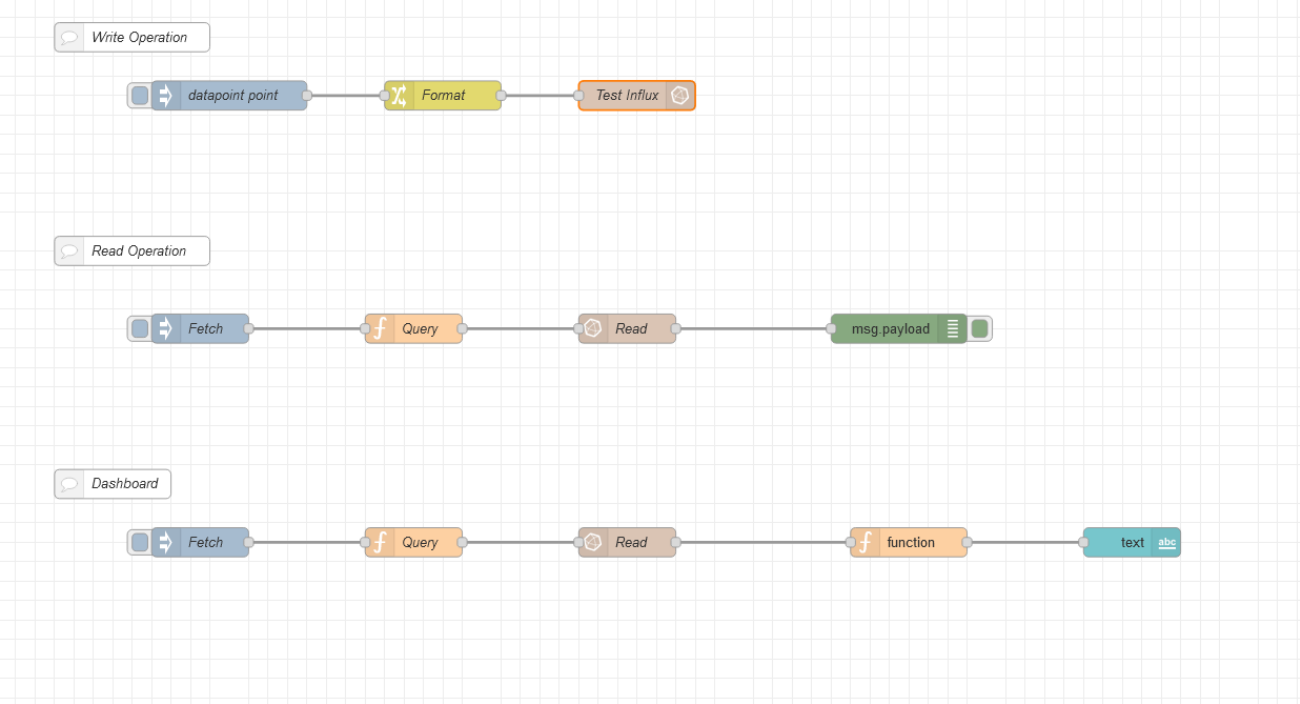
- https://portal.influxdata.com/downloads/
- wget https://dl.influxdata.com/influxdb/releases/influxdb-1.8.10_windows_amd64.zip -UseBasicParsing -OutFile influxdb-1.8.10_windows_amd64.zip
- Expand-Archive .\influxdb-1.8.10_windows_amd64.zip -DestinationPath 'C:\Program Files\InfluxData\influxdb\'

# INSTALL INFLUXDB

Install node-red-contrib-influxdb from install pallette

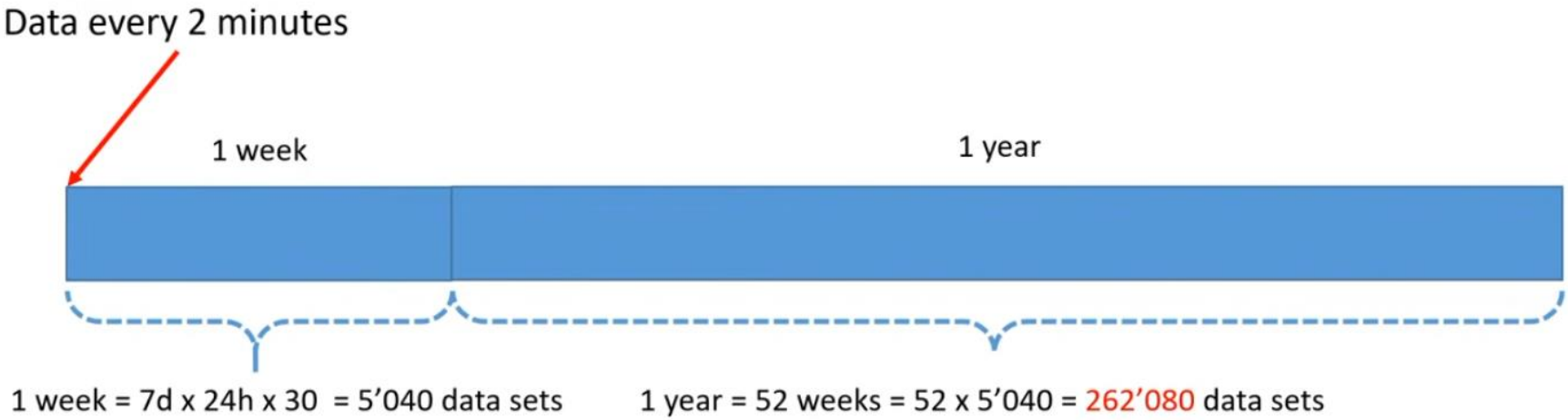


# INFLUXDB NODE RED TUTORIAL



# SQL QUERY

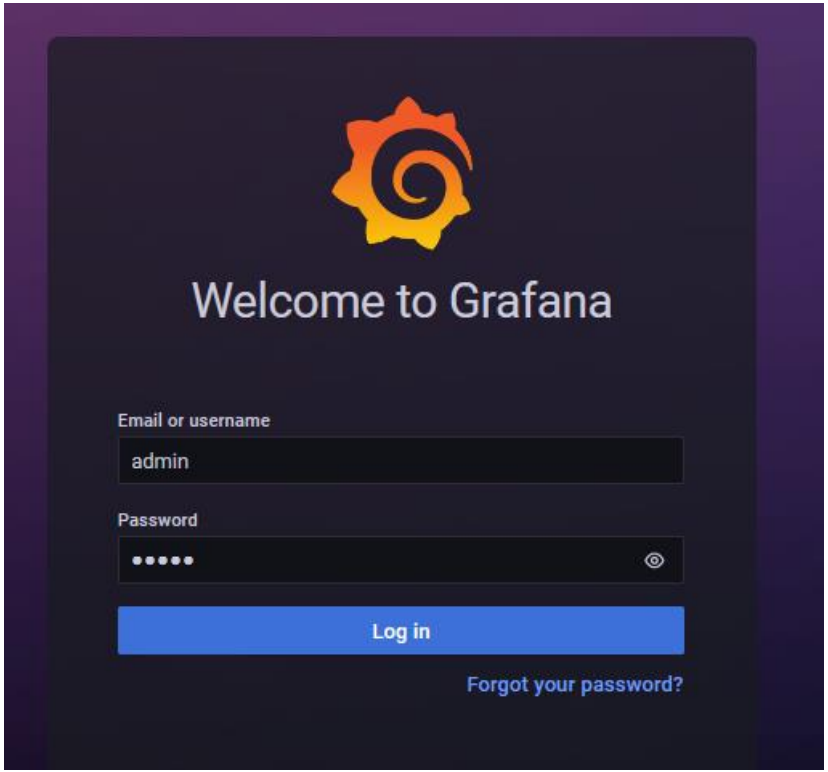
```
PS C:\Users\User\Documents\home\Project\FGD Tel-U 2022\influxdb-1.8.10-1> .\influx.exe
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
>
>
> show databases
name: databases
name
----
_internal
sensordata
> use sensordata
Using database sensordata
> select * from "sensordata"
name: sensordata
time                timestamp     type  value
----                ---------     ----  -----
1655866970748378600 1655866970745 point 14.454086270623922
1655866991794885500 1655866991792 point 6.2147892276364 64
>
```


Data every 2 minutes
1 week
1 year
1 week = 7d x 24h x 30 = 5’040 data sets
1 year = 52 weeks = 52 x 5’040 = 262’080 data sets

# GRAFANA

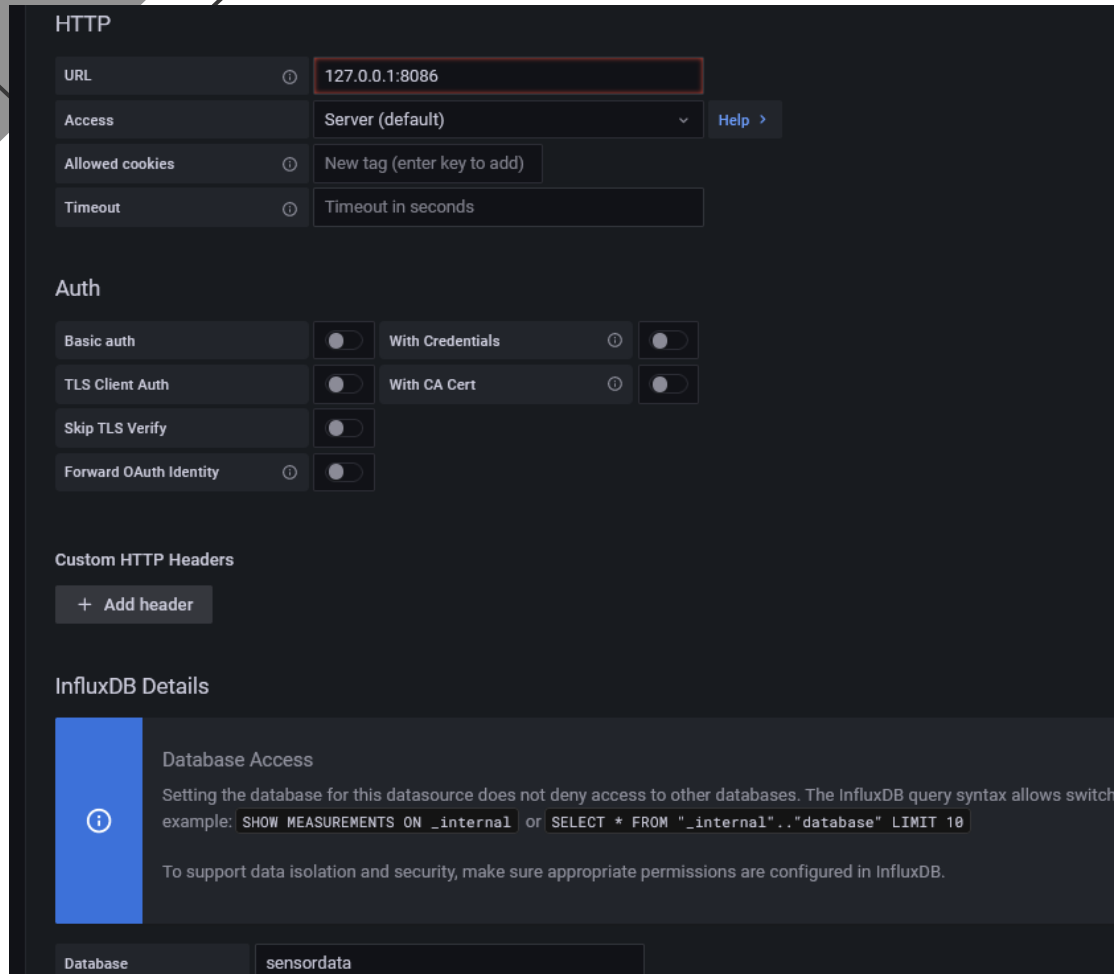
Dashboard

# OPEN HTTP://LOCALHOST:3000

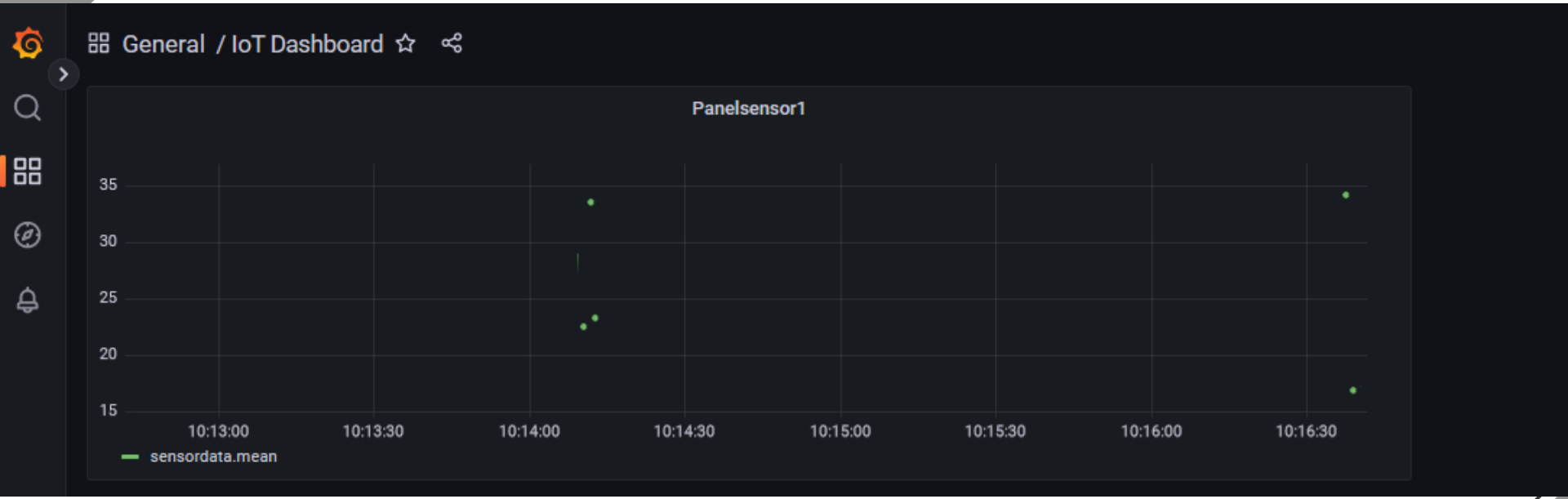



Username : admin
Password : admin
Configure datasource

# Database connection



# DASHBOARD


General / IoT Dashboard
Panelsensor1
35
30
25
20
15
10:13:00
10:13:30
10:14:00
10:14:30
10:15:00
10:15:30
10:16:00
10:16:30
sensordata.mean

# THANKS

Do you have any question?

hasbiida@gmail.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**