

# STEM in the Garden

How to monitor plants with IoT and Grafana



# whoami



testingwithmarie.com

- Senior Developer Advocate @ Grafana Labs
- 10+ years in Software Testing
- Co-author of Contract Testing in Action
- Mum of 2 (+2 furry babies)



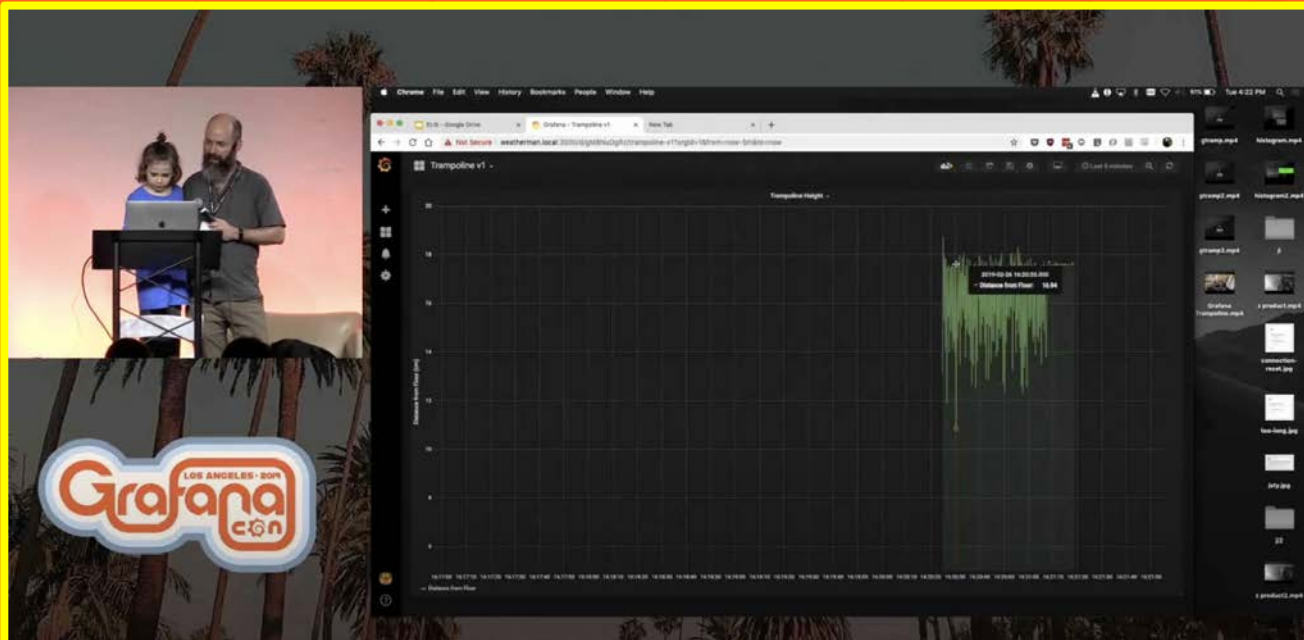
Contract Testing in Action



**Our mission: Look after our  
plants!**

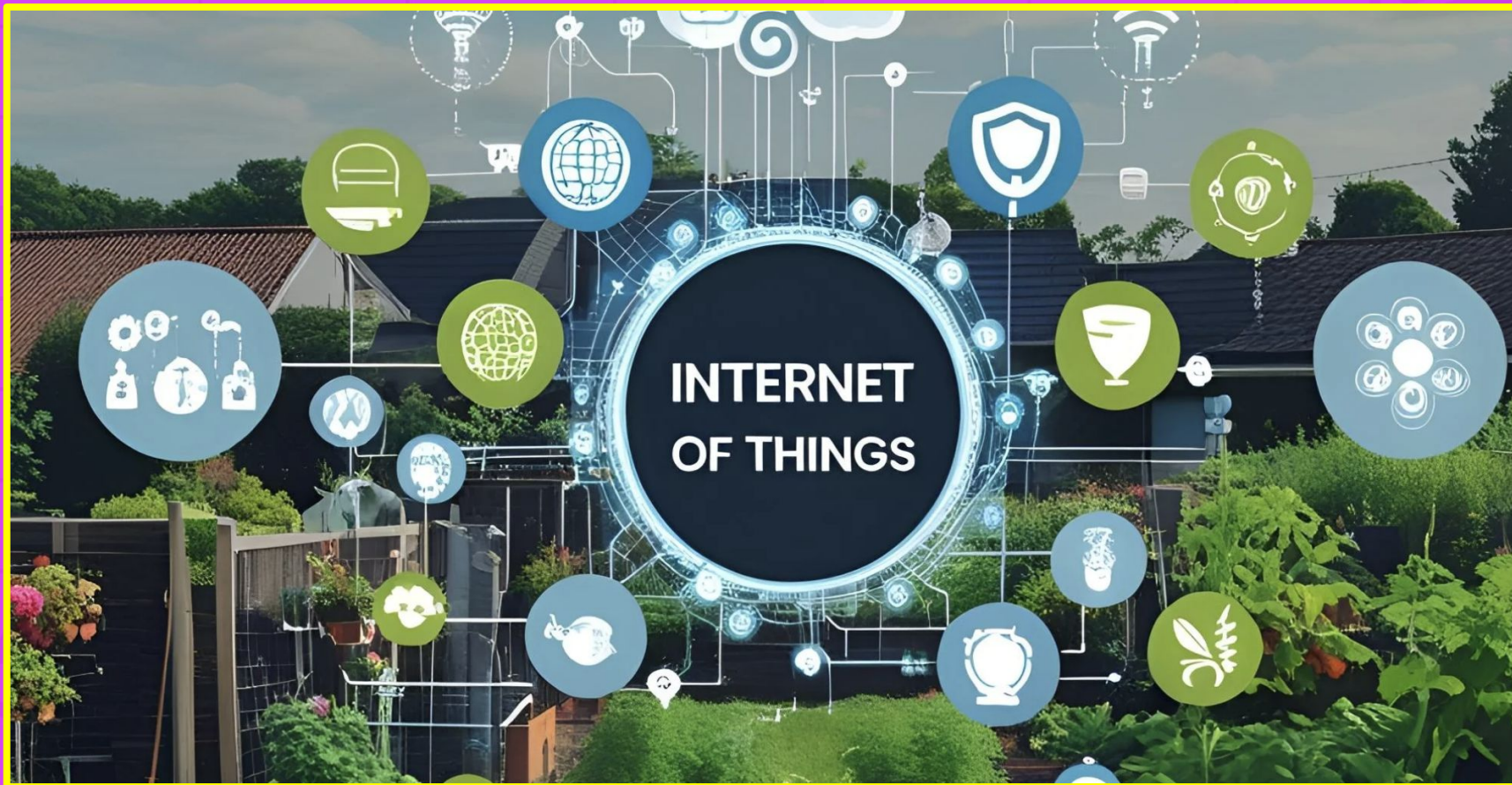
My mission: Let's make **STEM**  
learning fun!





**ELI5 (or 8?)**









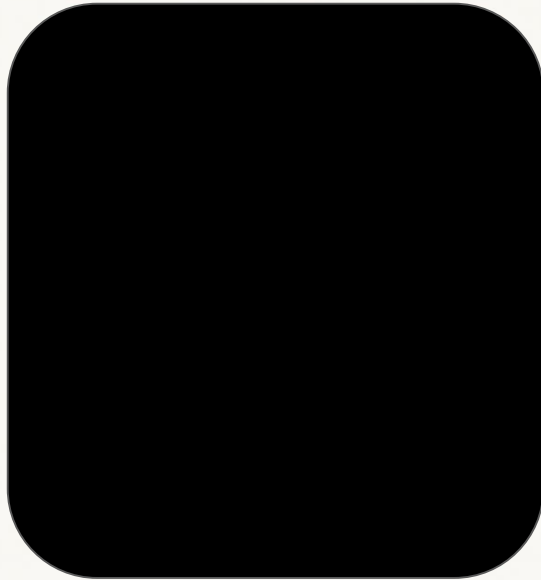
Mum, what's  
**monitoring**?

**Monitoring** keeps  
an eye on our  
plant.

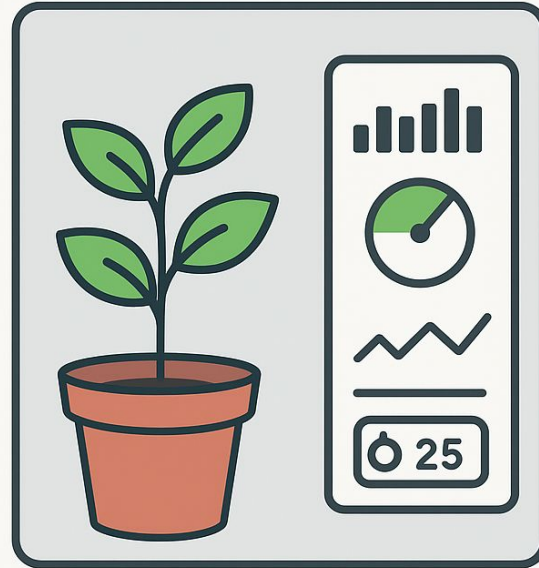
Ok, so what's  
**observability**?

**Observability**  
allows us to  
understand why  
our plant's health  
changes.

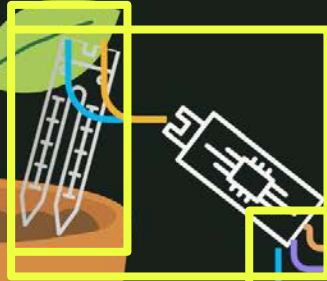
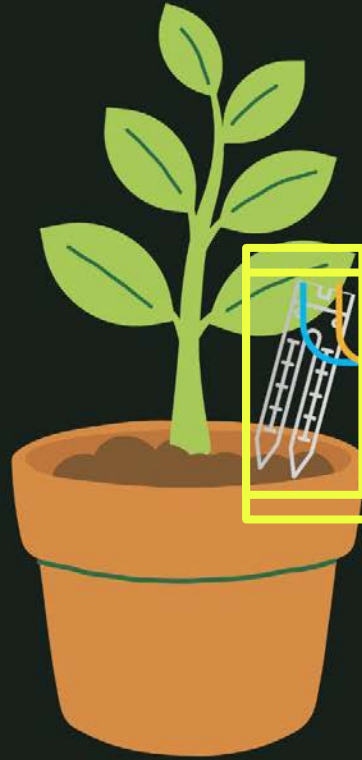
## Without Observability



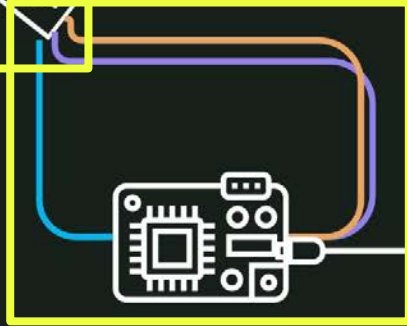
## With Observability



# **Part 1: Project setup**

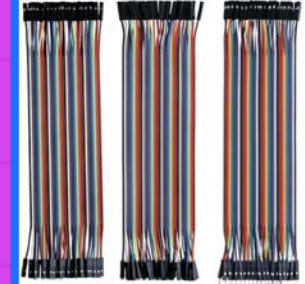
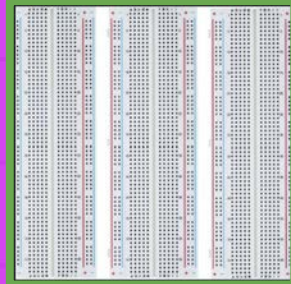
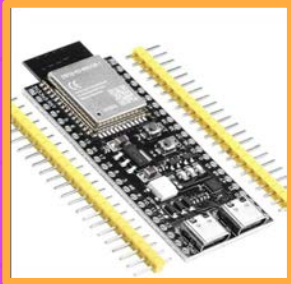


IOT





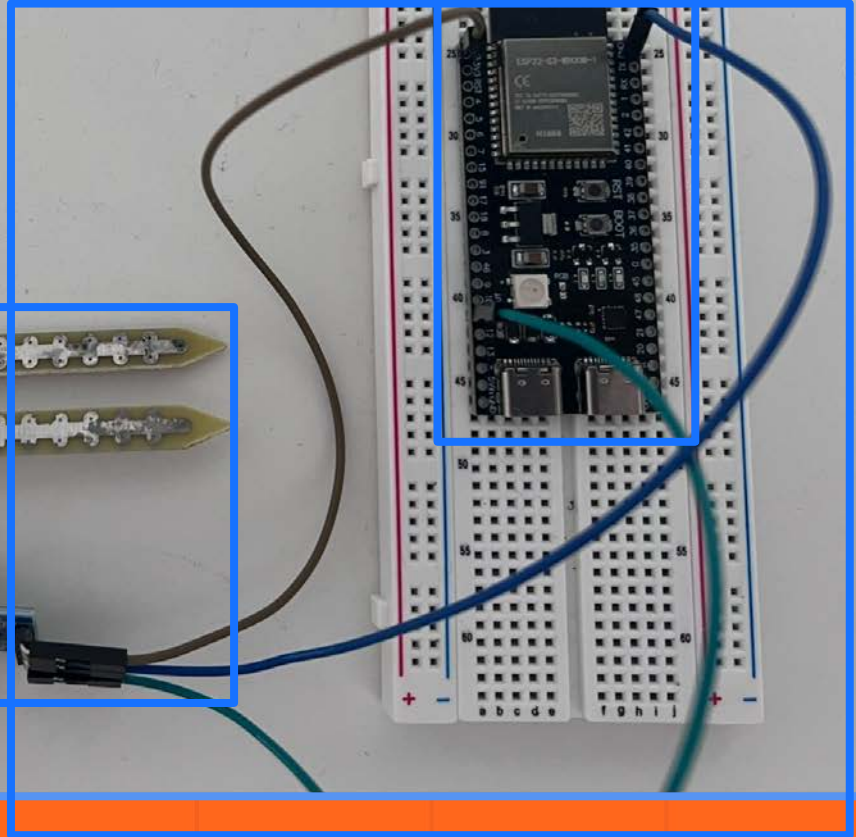
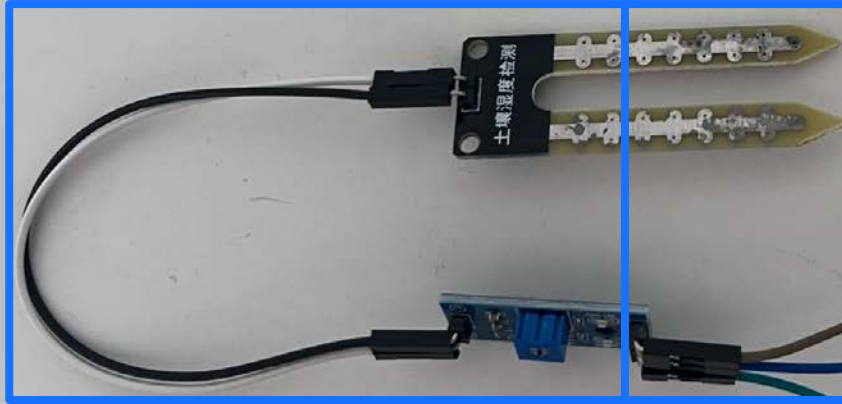
# Getting the crops - hardware



# Getting the crops - software



- VCC pin to 3.3V pin
- GND pin to GND pin
- AO pin to pin 11





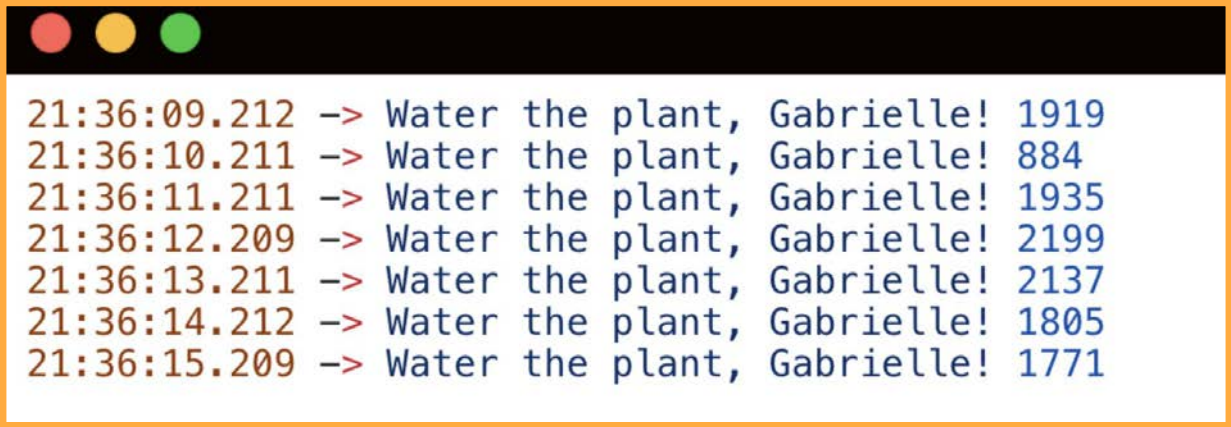
```
// Define the sensor pin and sensor variable
int sensorPin = 11;
int sensorValue = 0;
void setup() {
  // Initialize serial communication at 9600 baud rate
  Serial.begin(9600);
}
void loop() {
  // Read the analog value from the sensor
  sensorValue = analogRead(sensorPin);

  // Check if the soil is dry
  if (sensorValue > 500) {
    Serial.print(sensorValue);
    Serial.println(" - Status: Soil is too dry - time to water!");
  } else {
    Serial.print(sensorValue);
    Serial.println(" - Status: Soil is perfect!");
  }

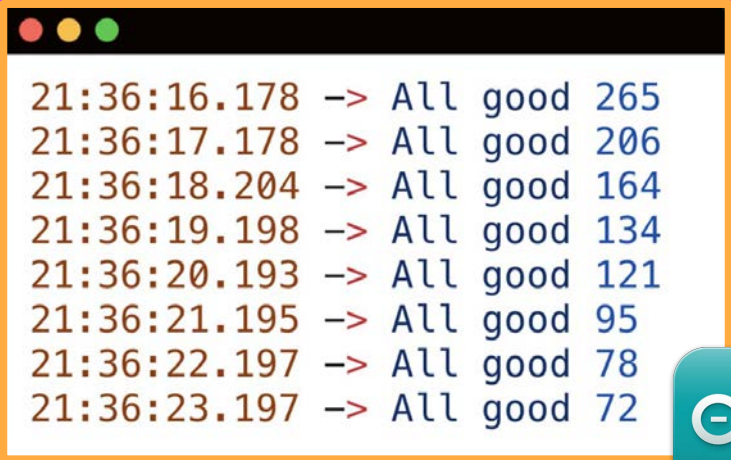
  // Wait 5s before taking another reading
  delay(5000);
}
```







```
21:36:09.212 -> Water the plant, Gabrielle! 1919
21:36:10.211 -> Water the plant, Gabrielle! 884
21:36:11.211 -> Water the plant, Gabrielle! 1935
21:36:12.209 -> Water the plant, Gabrielle! 2199
21:36:13.211 -> Water the plant, Gabrielle! 2137
21:36:14.212 -> Water the plant, Gabrielle! 1805
21:36:15.209 -> Water the plant, Gabrielle! 1771
```



```
21:36:16.178 -> All good 265
21:36:17.178 -> All good 206
21:36:18.204 -> All good 164
21:36:19.198 -> All good 134
21:36:20.193 -> All good 121
21:36:21.195 -> All good 95
21:36:22.197 -> All good 78
21:36:23.197 -> All good 72
```



## **Part 2: Send sensor values to Prometheus**



Now, let me  
introduce  
**Prometheus...**



# Prometheus: The Gardener



Time	Sensor value
2025-09-04 17:30:00	180
2025-09-04 17:35:00	200
2025-09-04 18:00:00	270

```
soil_moisture{monitoring_type="gabby_plant"}
```

**metric**

**Label filters**





# github.com/grafana/diy-iot

## PromLokiTransport by Ed Welch

0.2.2 installed

Abstracts a bunch of hardware and connectivity away from the GrafanaLoki and PrometheusArduino libraries Provides a simple client interface for GrafanaLoki and PrometheusArduino to be able to communicate over a network including SSL support

0.2.2

REMOVE

## PrometheusArduino by Ed Welch

0.4.0 installed

Send data to Prometheus via the remote write protocol. Lets you connect to any Prometheus server or any Prometheus compatible remote write endpoint like Cortex,Thanos,M3DB etc.

0.4.0

REMOVE

## SnappyProto by Ed Welch

0.1.2 installed

A library which includes a snappy compressor and nanopb for compressed protobufs. Primarily built to support GrafanaLoki and Prometheus both build snappy compressed protobufs.

0.1.2

REMOVE


to Grafana Cloud

Update labels

Launch

### Manage your stack

A stack groups together individual instances of all the services Grafana Labs offers for observing your software and infrastructure.




#### Grafana

Go to Grafana Cloud

Launch Details

Active Users: 0




#### Prometheus

Set up and manage your Prometheus metrics service.

Send Metrics Details

Active Users: 0 95th Percentile: 0 Current Active Series: 0




#### Loki

Set up and manage your Loki logging service.

Send Logs Details

Ingest Rate: 0 bytes/hr



#### OpenTelemetry

Connect using OTLP protocol.

Configure



```
// Prometheus details
#define GC_PROM_URL "prometheus-prod-13-prod-us-east-0.grafana.net"
#define GC_PROM_USER "" // username
#define GC_PROM_PASS "" // token
#define GC_PROM_PATH "/api/prom/push"

#define GC_PORT 443

// Wifi details
#define WIFI_SSID "" // WIFI name
#define WIFI_PASSWORD "" // WIFI password
```



```
#ifndef certificates_h
#define certificates_h

static const char grafanaCert[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIDjjCCAnagAwIBAgIQAzrx5qcRqaC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBH
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEwB3
d3cuZGlnaWlnaUNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
MjAeFw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTA1VT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEEdsb2JhbCBSc290IEcyMIIBIjANBgkqhkiG
9w0BAQEFAA0CAQ8AMIIBCgKCAQEAuzfNnN7a8myaJCtSnX/RrohCgiN9RlUyfuI
2/0u8jqJkTx65qsGGmvPrC3oXgkRRLpimn7Wo6h+4FR1IAWsULecYxpsMNzaHmx
1x7e/dfgy5SDN67sH0N03Xss0r0upS/kqbit0tSZpLYl6ZtrAGCSYP9PIuKY92e0
q2EGnI/yuum06ZIya7XzV+hdG82MHauVBjVJ8zUtlunJbd134/tJS7SsVQepj5V
tC07TG1F8PapsPuwtp1MVYwnSlcUfIKdzX0S0xZKBgyMUNGPHgm+F6HmIcr9g+l
```



[github.com/grafana/prometheus-arduino](https://github.com/grafana/prometheus-arduino)

plant-monitoring.ino certificates.h config.h README.md

```
1  #include <PromLokiTransport.h>
2  #include <PrometheusArduino.h>
3
4  #include "config.h"
5  #include "certificates.h"
6
7  // Define sensor values
8  int sensorPin = 11;
9  int sensorValue = 0;
10
11 // Prometheus client and transport
12 PromLokiTransport transport;
13 PromClient client(transport);
14
15 // Create a write request
16 WriteRequest req(1, 3074);
17
18 // Define TimeSeries
19 TimeSeries ts(5, "soil_moisture", "{monitoring_type=\"gabby_plant\",board_type=\"esp32\",room=\"living_room\"}");
20
21 int loopCounter = 0;
22
23 void setup() {
24     // Start the serial output at 9,600 baud
25     Serial.begin(9600);
26
27     uint8_t serialTimeout;
28     while (!Serial || serialTimeout < 50) {
```



[github.com/mdcruz/plant-monitoring](https://github.com/mdcruz/plant-monitoring)

```
21:34:50.851 -> Sending To Prometheus  
21:34:50.916 -> Connection already open  
21:34:50.916 -> Sent, waiting for response  
21:34:52.081 -> Prom Send Succeeded
```





# **Part 3: Visualise to Grafana**



It is time to  
know **Grafana**,  
the visualiser.



Add ▾

Settings

Save dashboard



Last 6 hours

UTC ▾



Refresh



## Start your new dashboard by adding a visualization

Select a data source and then query and visualize your data with charts, stats and tables or create lists, markdowns and other widgets.

+ Add visualization

### Import panel

Add visualizations that are shared with other dashboards.

+ Add library panel

### Import a dashboard

Import dashboards from files or [grafana.com](https://grafana.com) ↗.

📁 Import dashboard



Data source



grafanacloud-mariecruz



> Query options

MD = auto = 1139

Interval = 1m

Query inspector



A

(grafanacloud-mariecruz-prom)



Kick start your query

Explain



Run queries

Builder

Code

Metric

Label filters

soil\_moisture



monitoring\_type



=



gabby\_plant

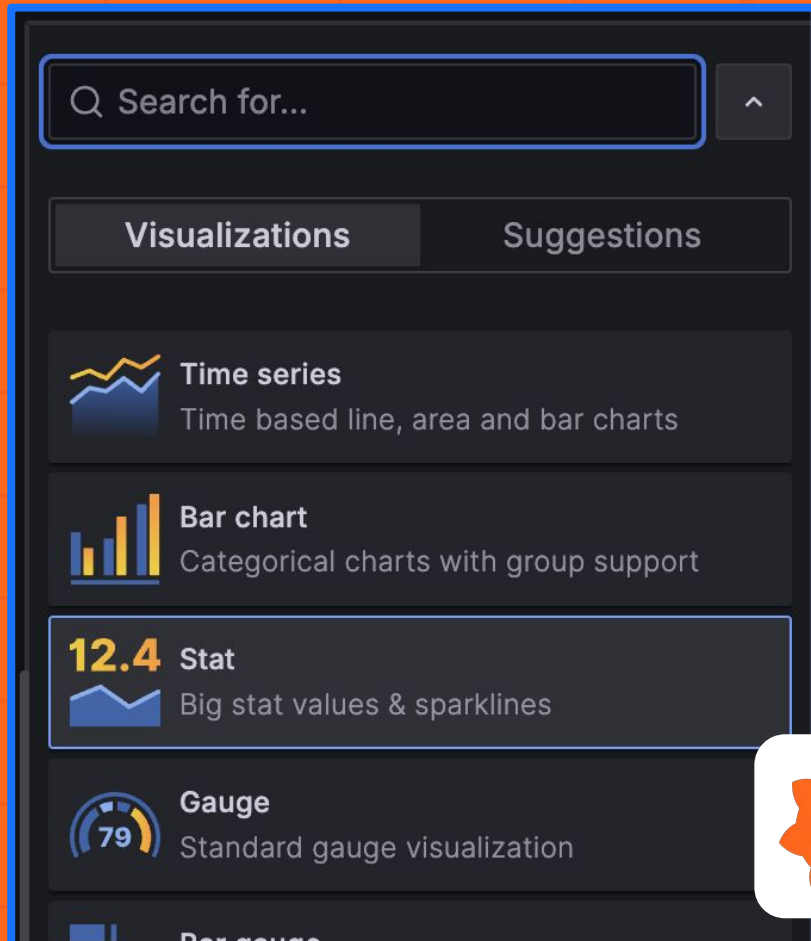


+ Operations

```
soil_moisture{monitoring_type="gabby_plant"}
```



# Visualizations



# Text

# Stat

# Time series



# Gauge



## Gabrielle's Plant Dashboard

This is an example dashboard to show how we can use Grafana to monitor the health of our plant.

Snake plant 🐍



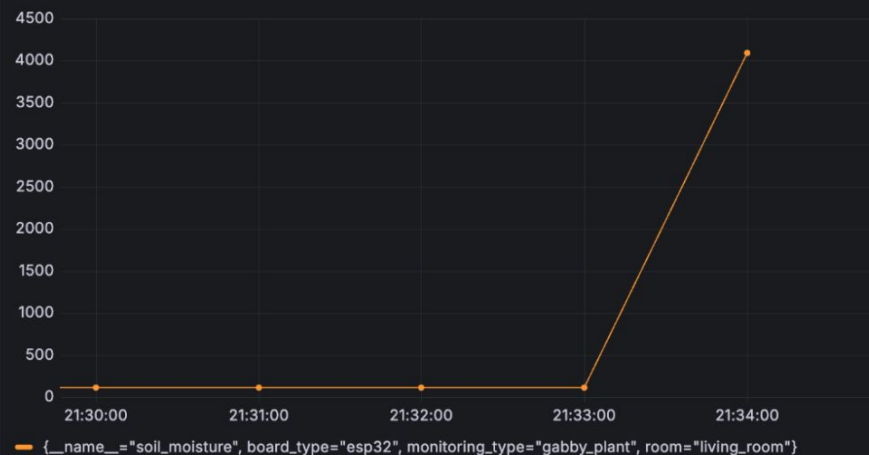
Plant status

Plant is thirsty! 🌵

Soil moisture 🌿



Soil moisture values





## Value mappings



Condition		Display text	Color
⋮ Range	500	To	Plant is thirsty! 🌵
⋮ Range	0	499	Plant is healthy! 🌿

+ Add a new mapping

Cancel

Update

## 2. Define query and alert condition


Define query and alert condition ⓘ [Need help?](#)

 grafanacloud-mariacruz ▾ ⓘ Options ▾ 10 minutes, MD = 43200, Min. Interval = 1s

Kick start your query

Explain ☐

Metric

soil\_moisture ▾ 

Label filters

monitoring\_type ▾

= ▾

gabby\_plant ▾

×

+

+ Operations

```
soil_moisture{monitoring_type="gabby_plant"}
```

> Options Legend: Auto Format: Time series Step: auto Type: Instant

Alert condition

WHEN QUERY

IS ABOVE ▾

500

 Preview alert rule condition



## 5. Configure notifications

Select who should receive a notification when an alert rule fires.

grafana-default-email

email

plant-alert

telegram

plant-alert

[Need help?](#)

[View or create contact points](#)

P

plantbot

Gabrielle, the plant needs to be watered! 🌵 14:39



plantbot

Gabrielle, the plant needs to be watered! 🌵 14:45

Gabrielle, the plant needs to be watered! 🌵 14:46

P

Gabrielle, the plant needs to be watered! 🌵 14:47

## 🌱 Gabrielle's Plant Dashboard 🌱

Monitor your green friends with style!

### 🔍 Dashboard Features

- 💧 Real-time Moisture Monitoring - Track soil hydration levels
- 🌿 Plant Health Status - Visual health indicators with emojis
- 📊 Historical Trends - Track moisture patterns over time
- ⚡ Smart Alerts - Get notified when your plant needs attention

🔗 Explore Grafana Play for more dashboard inspiration!

#### 🌿 Your Snake Plant ⓘ



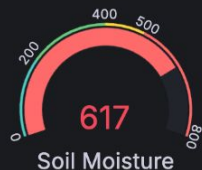
🌐 IoT Enabled

#### 🌿 Plant Health Status ⓘ

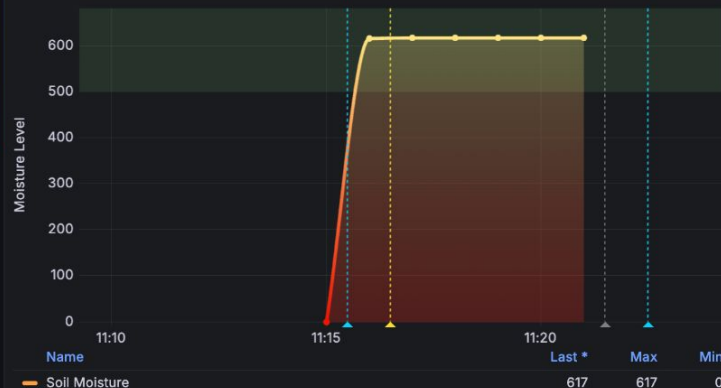


# Thirsty Plant!

#### 💧 Moisture Level Gauge ⓘ



#### 📊 Moisture Trends Over Time ⓘ ❤️



Can you provide visual enhancements to this dashboard

I'll help you enhance the visual aspects of Gabrielle's Plant Dashboard. Let me first examine the current dashboard configuration to understand what panels and visualizations are already in place.

✓ read dashboard information

Perfect! I can see this is a charming plant monitoring dashboard with several panels. Let me enhance the visual aspects by improving colors, layouts, thresholds, and adding some visual polish. I'll focus on:

1. Improving the visual hierarchy and layout
2. Enhancing color schemes and thresholds
3. Adding visual consistency across panels
4. Improving readability and aesthetics

✓ update dashboard panels


@ Add context

Spark a conversation, @ to fuel it with context

🗨 Give feedback

🔒 Limits ▾

# Takeaways

- Learning about monitoring & observability is fun.
- Getting started with IoT and Grafana is great for STEM learning.
- Our plants are healthy! 

**Get all  
the info!**



# STEM in the Garden

How to monitor plants with IoT and Grafana



**Thank you!**

Questions specific to Grafana?

Ask us at [community.grafana.com](https://community.grafana.com)



Connect with me!