



Ben-Gurion University  
of the Negev

# Course: Internet of Things

Lecturer: Guy Tel-Zur

Presentors: Tal Adoni, Omri Aviram

Subject: Smart Irrigation system

# My Smart Plant

My Smart  
Plant





The Story – Thailand & the Basil Plant



# Literature Review



Contents lists available at ScienceDirect

Environmental Research

journal homepage: [www.elsevier.com/locate/envres](http://www.elsevier.com/locate/envres)



# Comparative analysis of IoT-based controlled environment and uncontrolled environment plant growth monitoring system for hydroponic indoor vertical farm <sup>☆</sup>

Gaganjot Kaur <sup>a,\*</sup>, Prashant Upadhyaya <sup>a</sup>, Paras Chawla <sup>b</sup>

<sup>a</sup> Chandigarh University, Chandigarh-Ludhiana Highway, Mohali, Punjab, India

<sup>b</sup> Zonal Head- IT Academics, Ajeenkyaa D Y Patil University, Pune, Maharashtra, (iNurture Education Solution, Bangalore), India



# Automated vs Manual

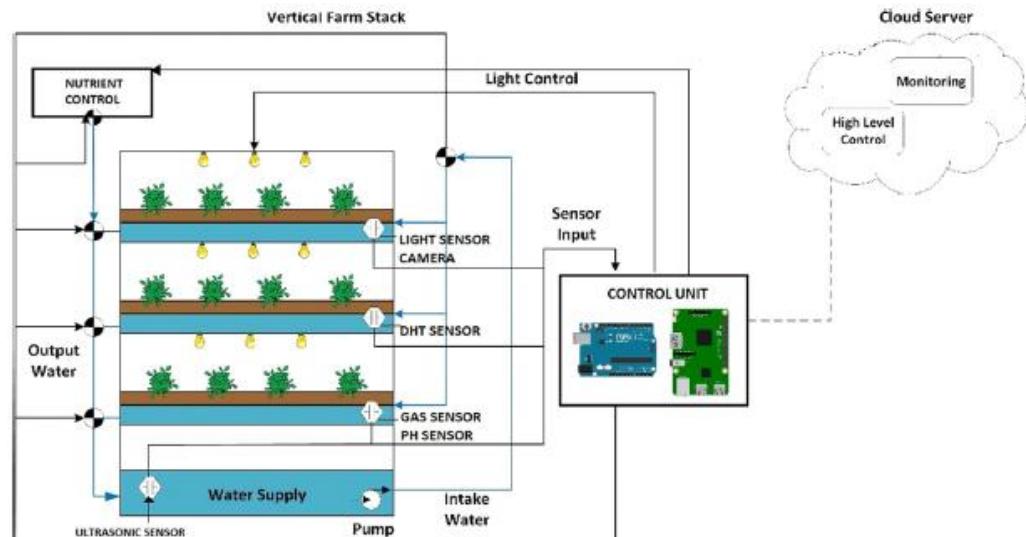


Fig. 2. An automated vertical hydroponic farming setup.

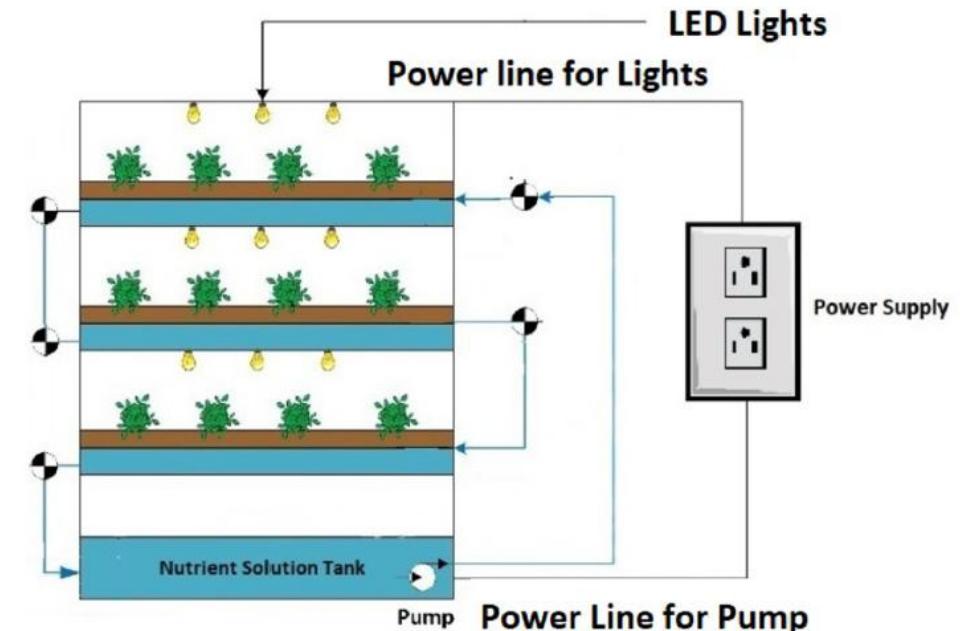
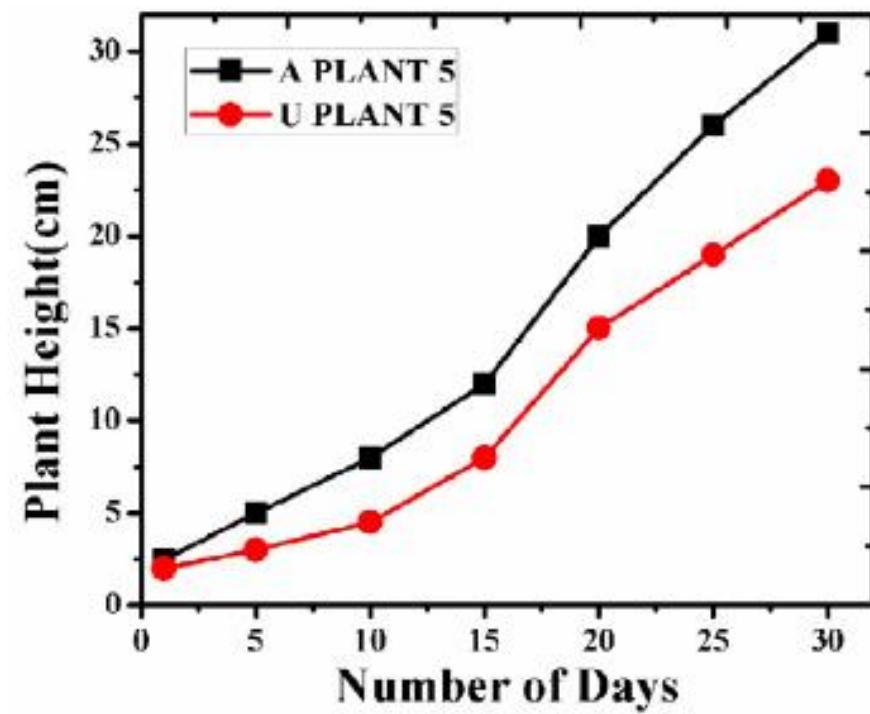


Fig. 1. An unautomated vertical hydroponic farming setup.

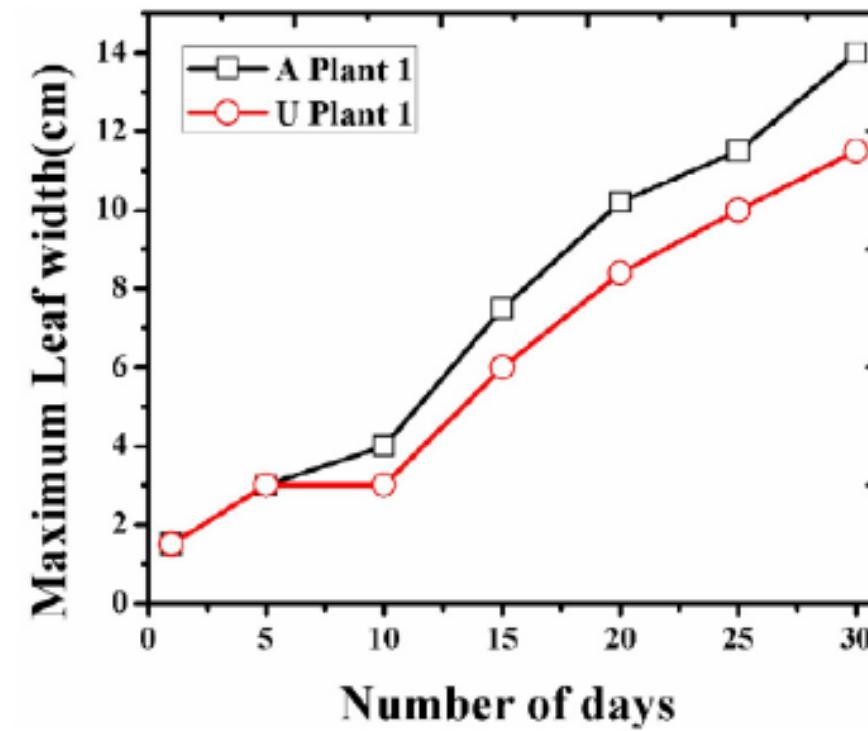
- Active components – Arduino Mega and Raspberry Pi
- Contain Cameras, Flow sensors, Gas Sensors and Water-level sensors

- Passive components for monitoring the data
- Done by the Grower

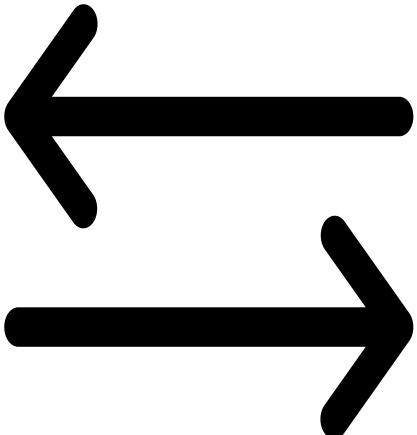
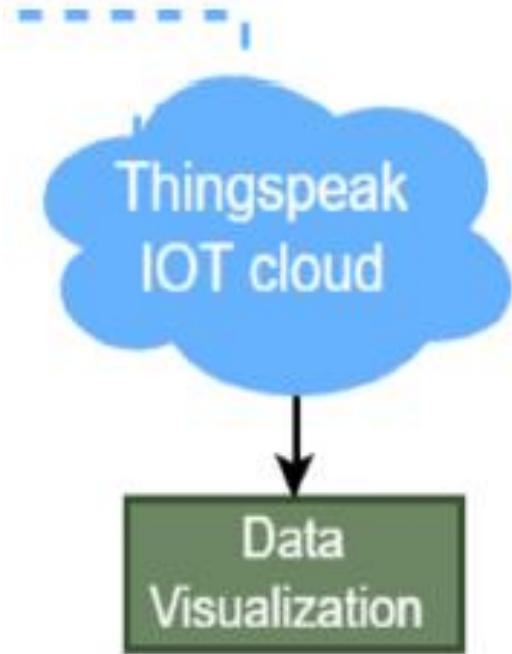
# Automated vs Manual



(e) Plant 5



(a) Plant 1

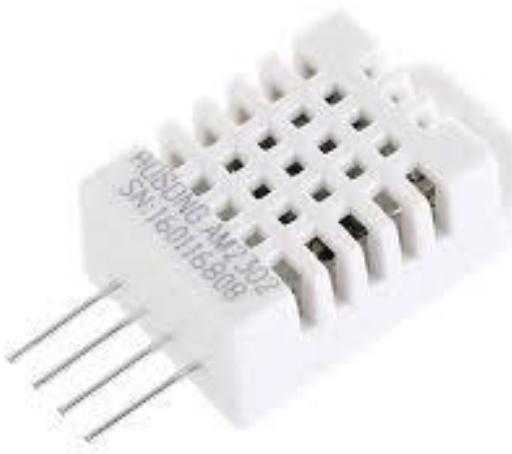


 **QuestDB**

 **Grafana**

**aws**  

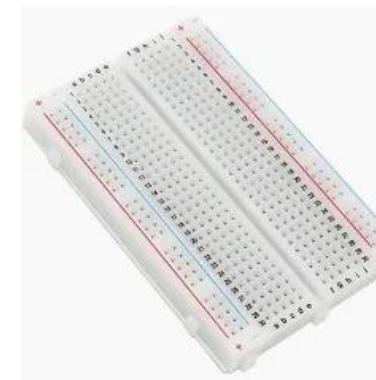

# System Components



DHT22 – Temperature  
and humidity sensor



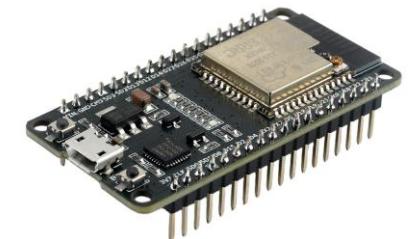
LED



Breadboard



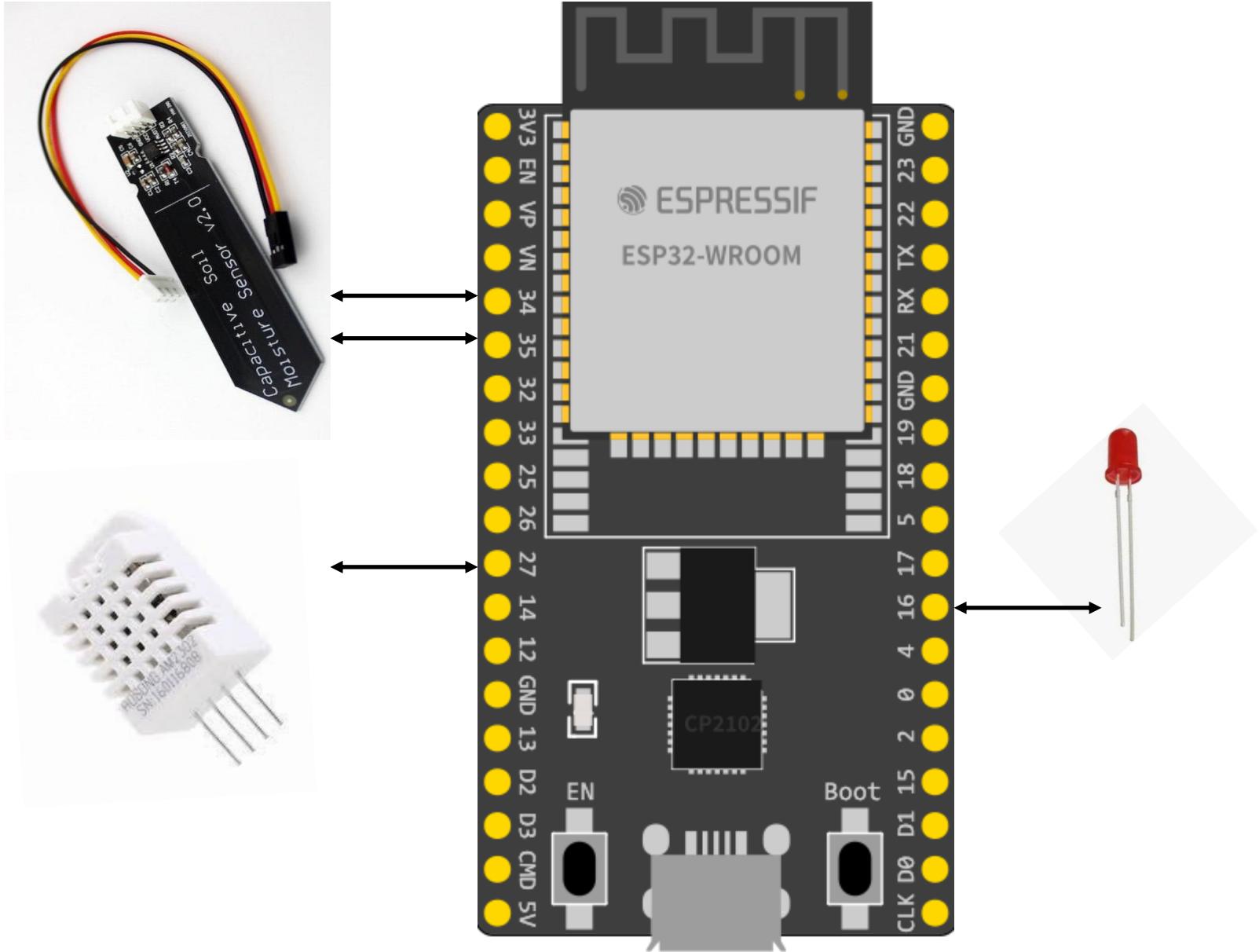
Capacitive Soil  
Moisture Sensor v2.0



ESP32 –  
Microcontroller

# Hardware Connections & Local Control

- Capacitive soil moisture sensors connected to ESP32 ADC pins (Ports 34, 35) – 500 averaged samples
- Local temperature and humidity sensing using a DHT22 sensor (GPIO 17)
- LED used as a local status indicator, designed to operate independently of internet connectivity (GPIO 16)



# Network Connectivity & Cloud Ingestion



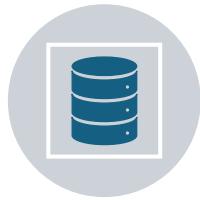
Wi-Fi connection from ESP32 to local network.



Secure MQTT communication over TLS.



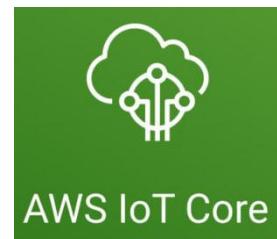
Device authentication using X.509 certificates.



Data ingestion via AWS IoT Core.



IoT Rules engine for routing incoming telemetry.



AWS IoT Core



# Data Storage – Amazon S3

- Sensor data stored as JSON files in Amazon S3
- Each plant has its own telemetry folder
- Inside the telemetry folder latest.json holds the most recent measurement
- S3 used as a simple, scalable, and low-cost data store

Amazon S3 > Buckets > esp32-hamacheshir-telemetry > plant1/

**plant1/**

**Objects** **Properties**

**Objects (3)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#)

<input type="checkbox"/> Name	Type
<input type="checkbox"/> <a href="#">images/</a>	Folder
<input type="checkbox"/> <a href="#">index.html</a>	html
<input type="checkbox"/> <a href="#">telemetry/</a>	Folder

▼ iot/irrigation/ESP32\_Hamacheshir/telemetry/plant1

```
{
  "thing": "ESP32_Hamacheshir",
  "plantId": "plant1",
  "tempC": 21.1,
  "humRH": 52.9,
  "soilRaw": 2080,
  "wateringPct": 30,
  "thirsty": false,
  "state": "NEEDS_WATER",
  "timestamp": "2026-01-11T12:39:47+02:00",
  "epochMs": 1768127987000
}
```

**Json format**

**Objects (1)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get:

**Find objects by prefix**

**Name** **Type**

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	<a href="#">latest.json</a>	json

# Initial Prototype – Single Plant

S3



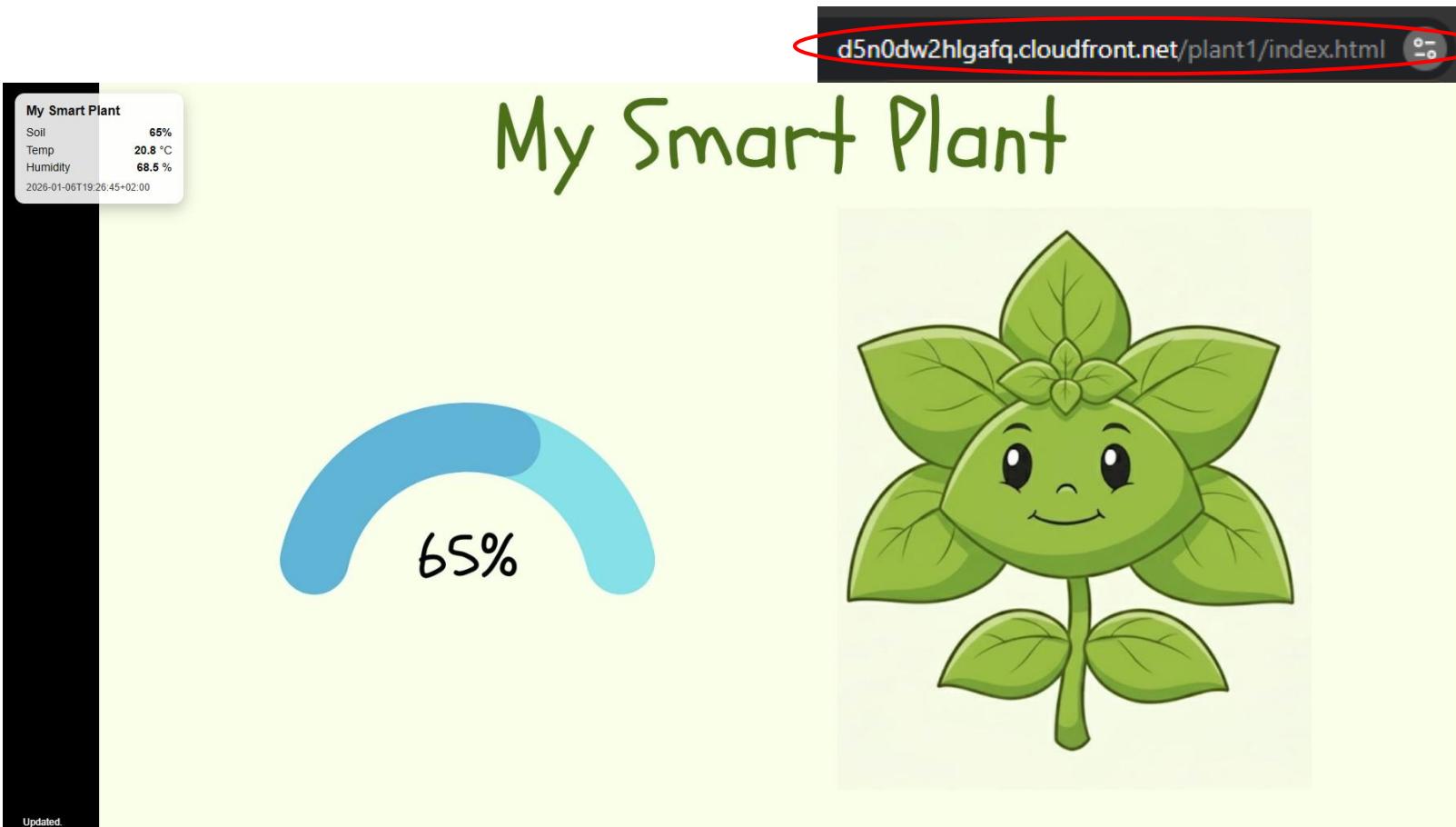
- One plant monitored
- Data stored in S3
- Static website hosted directly from S3
- Public, non-secure access

The screenshot shows a web page titled "My Smart Plant". At the top left is a small sidebar with the title "My Smart Plant" and three data points: Soil (100%), Temperature (20.9 °C), and Humidity (66.5%). Below this, a timestamp reads "2026-01-04T13:30:52+02:00". The main content area features a large blue gauge chart showing "100%" completion. To the right of the gauge is a cartoon illustration of a happy green plant with large leaves and a smiling face. At the top of the page, the URL "esp32-hamacheshir-telemetry.s3-website-us-east-1.amazonaws.com" is displayed, with the ".amazonaws.com" part circled in red. A small Hebrew note "ללא מסמכה" (without certificate) with a warning triangle icon is also present.

# Secure Visualization – CloudFront



- CloudFront used as a secure content delivery layer
- HTTPS enabled with Amazon certificate
- Controlled access to S3
- Improved performance and security



A screenshot of a web-based visualization for a "Smart Plant". The URL in the browser bar is highlighted with a red oval, showing `d5n0dw2hlqafq.cloudfront.net/plant1/index.html`. The dashboard features a title "My Smart Plant" in green, a cartoon plant character with a smiling face, and a large blue gauge meter indicating "65%". To the left, there's a small card with real-time sensor data: Soil (65%), Temp (20.8 °C), and Humidity (68.5 %), updated at 2026-01-06T19:26:45+02:00.

My Smart Plant

65%

20.8 °C

68.5 %

My Smart Plant

65%

Updated: 2026-01-06T19:26:45+02:00



**מציג האישורים: \*cloudfront.net.\***

**כללי** **פרטים**

**הונפק ל**

שם נפוץ (CN) cloudfront.net.\*  
 ארגון (O) <לא חלק מהאישור>  
 יחידה ארגונית (OU) <לא חלק מהאישור>

**הונפק על ידי**

שם נפוץ (CN) Amazon RSA 2048 M01  
 ארגון (O) Amazon  
 יחידה ארגונית (OU) <לא חלק מהאישור>

**תקופת תוקף**

יום שני, 5 במאי 2025 בשעה 3:00:00  
 יומ שישי, 24 באפריל 2026 בשעה 2:59:59  
 הונפק בתאריך  
 בתוקף עד

סכימות אצבע דיגיטליות מסוג SHA-256

אישור  
 מפתח ציבורי  
 60389d249e418f23acd9145ca3477eaf07db9f2d6a8c0d08e  
 9248a8e49a94d28  
 eb36f1d3bc60f9b7ec0e4faf26130c3caf49856acb46a63485  
 29842023e98be9

**אבטחה** **d5n0dw2hlqafq.cloudfront.net**

**החיבור מאובטח**

הפרטים שלך (כמו סיסמות או מספרי כרטיסי אשראי) נשלחים לאתר זהה במחבץ פרטני. **למידע**

**האישור תקין**

# Scalability – Multiple Plants



- Additional soil moisture sensors added
- Each plant has its own data pipeline
- Separate secure website per plant
- QR code for easy accessss

plant2/

Objects Properties

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get

Find objects by prefix

<input type="checkbox"/> Name	Type
<input type="checkbox"/> <a href="#">images/</a>	Folder
<input type="checkbox"/> <a href="#">index.html</a>	html
<input type="checkbox"/> <a href="#">telemetry/</a>	Folder

Updated.

The interface shows a header with the URL `d5n0dw2hlqafq.cloudfront.net/plant2/index.html` and a red oval highlighting it. The main title is "My Smart Plant". On the left, there's a small card with sensor data: Soil 0%, Temp 20.9 °C, Humidity 69.0%. Below the card is a large blue gauge with "0%" in the center. To the right is a cartoon illustration of a brown, shriveled plant with sad eyes, standing in dry, cracked ground.



plant 2



Plant 1

My Smart Plant

Soil 90%  
Temp 21.1 °C  
Humidity 68.4 %  
2026-01-06T20:20:08+02:00

90%

Updated.



My Smart Plant

My Smart Plant

Soil 85%  
Temp 21.1 °C  
Humidity 68.2 %  
2026-01-06T20:19:43+02:00

85%

Updated.



All Plants Data

My Smart Plant

My Smart Plant

All Plants Data Link

 All Plants Data

**Plant 1**

Temperature 21.1 °C

Humidity 68.2 %

Soil Moisture 90%

Status 

Last Cloud Update: 20:19:38

**Plant 2**

Temperature 21.1 °C

Humidity 68.2 %

Soil Moisture 85%

Status 

Last Cloud Update: 20:19:43

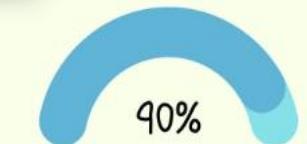
**My Smart Plant**

Soil 90%

Temp 21.1 °C

Humidity 68.4 %

2026-01-06T20:20:08+02:00



90%



Updated.

**My Smart Plant**

Soil 85%

Temp 21.1 °C

Humidity 68.2 %

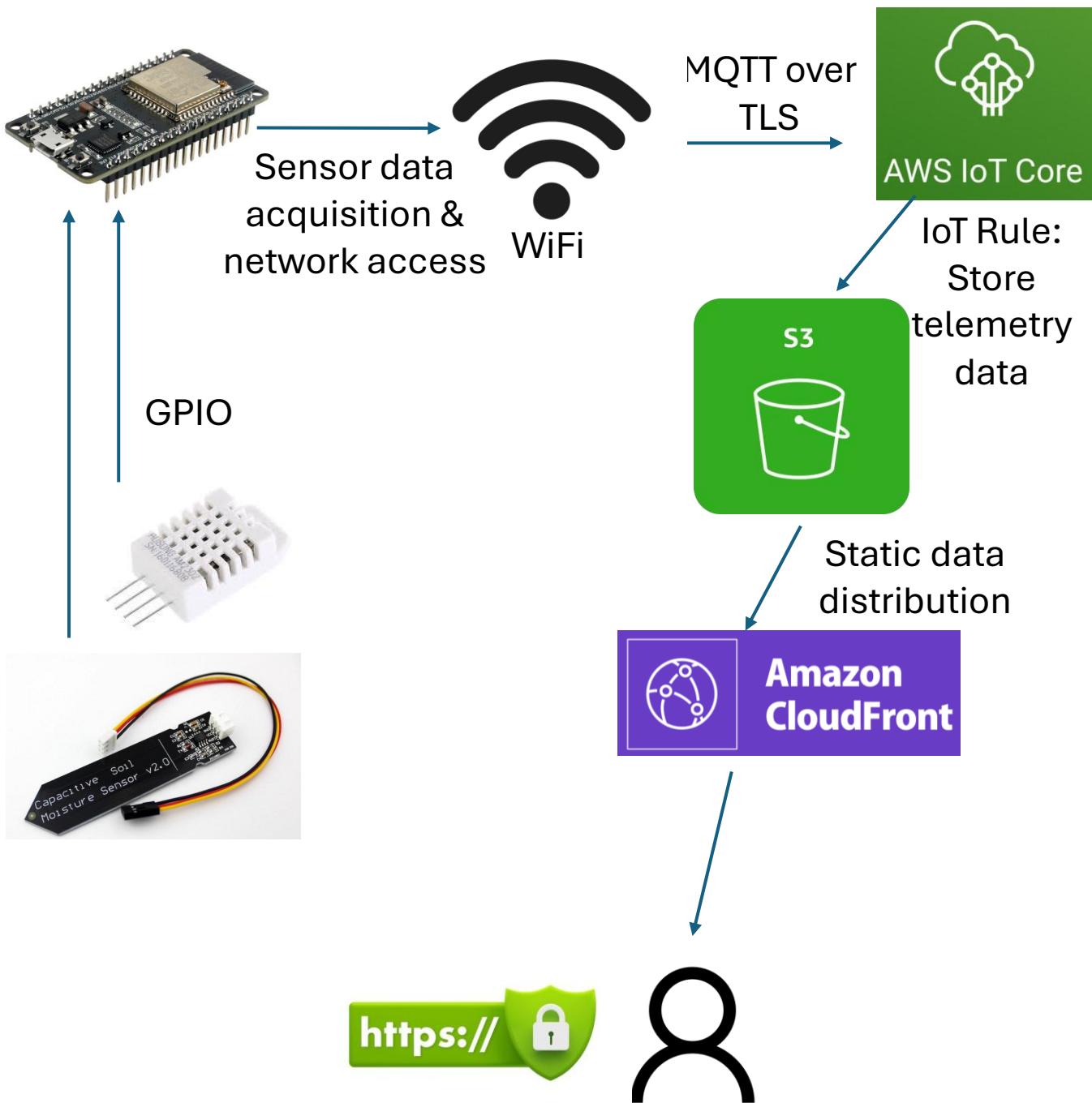
2026-01-06T20:19:43+02:00



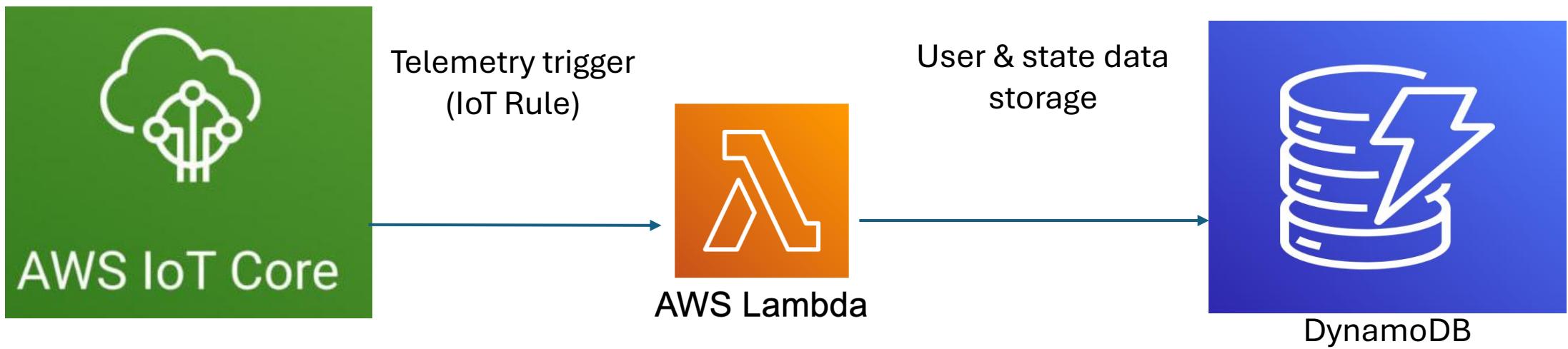
85%



Updated.



# Cross-Account IAM Role



# 1. Role

Screenshot of the AWS IAM Roles page showing a list of 7 roles.

**Roles (7) Info**

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
<a href="#">aws-iot-rul</a>	AWS Service: iot	9 days ago
<a href="#">AWSServiceRoleForResourceExplorer</a>	AWS Service: resource-explorer-2 (Service-Linker)	12 minutes ago
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	-
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-
<a href="#">ESP32_Receiver_Role</a>	Account: 805171368357	21 hours ago
<a href="#">iot_to_dynamodb_role</a>	AWS Service: iot	9 days ago
<a href="#">Telegram_Plant_Alert-role-rqfnzohq</a>	AWS Service: lambda	16 minutes ago

**Roles Anywhere** Info

Authenticate your non AWS workloads and securely provide access to AWS services.

**Access AWS from your non AWS workloads**

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

**X.509 Standard**

Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.

**Temporary credentials**

Use temporary credentials with ease and benefit from the enhanced security they provide.

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie pre

# 2. Permissions Policy

The screenshot shows the AWS IAM console with the policy details for 'CrossAccountDynamoWrite'. The policy is customer-managed and was created on January 02, 2026, at 16:02 UTC+02:00. It was last edited on January 02, 2026, at 20:01 UTC+02:00. The ARN is arn:aws:iam::096938402672:policy/CrossAccountDynamoWrite. The 'Permissions' tab is selected, showing the JSON code for the policy:

```
1 [ { "Version": "2012-10-17", "Statement": [ 2 { "Effect": "Allow", "Action": [ 3 "dynamodb:PutItem", 4 "dynamodb:DescribeTable" ], 5 "Resource": "arn:aws:dynamodb:us-east-1:096938402672:table/PlantData" } ] } ]
```

A red oval highlights the 'Action' section of the JSON code, specifically the 'dynamodb:PutItem' and 'dynamodb:DescribeTable' actions. Another red oval highlights the 'Resource' section, which is the ARN of the DynamoDB table.

**Identity and Access Management (IAM)**

Search IAM

Dashboard

**Access Management**

- User groups
- Users
- Roles
- Policies**
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests

**Access reports**

- Access Analyzer
- Resource analysis New
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies

Comparative analysis of IoT-based... | Google Search - סדרת נסיעות צבאי | ESP32\_Receiver\_Role | IAM | Gl... | CrossAccountDynamoWrite | IA... | +

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/policies/details/arn%3Aaws%3Aiam%3A096938402672%3Apolicy%2FCrossAccountDynamoWrite?section=permissions&view=json

Search [Alt+S]

Global Account ID: 0969-3840-2672 omriavi

IAM Policies CrossAccountDynamoWrite

**CrossAccountDynamoWrite** Info

**Policy details**

Type: Customer managed Creation time: January 02, 2026, 16:02 (UTC+02:00) Edited time: January 02, 2026, 20:01 (UTC+02:00) ARN: arn:aws:iam::096938402672:policy/CrossAccountDynamoWrite

**Permissions** Entities attached Tags Policy versions (2) Last Accessed

**Permissions defined in this policy** Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Copy Edit Summary JSON

```
1 [ { "Version": "2012-10-17", "Statement": [ 2 { "Effect": "Allow", "Action": [ 3 "dynamodb:PutItem", 4 "dynamodb:DescribeTable" ], 5 "Resource": "arn:aws:dynamodb:us-east-1:096938402672:table/PlantData" } ] } ]
```

# 3.Trust Relationship

The screenshot shows the AWS IAM console interface. The left sidebar is collapsed, and the main area displays the details for the role 'ESP32\_Receiver\_Role'. The 'Trust relationships' tab is selected, showing the JSON policy document. A purple oval highlights the 'AWS' field in the policy statement, which contains the ARN of the root user.

**ESP32\_Receiver\_Role** Info

**Summary**

**Creation date**  
January 02, 2026, 16:11 (UTC+02:00)

**Last activity**  
21 hours ago

**ARN**  
arn:aws:iam::096938402672:role/ESP32\_Receiver\_Role

**Maximum session duration**  
1 hour

**Link to switch roles in console**  
[https://signin.aws.amazon.com/switchrole?roleName=ESP32\\_Receiver\\_Role&account=096938402672](https://signin.aws.amazon.com/switchrole?roleName=ESP32_Receiver_Role&account=096938402672)

**Permissions** **Trust relationships** **Tags** **Last Accessed** **Revoke sessions**

**Trusted entities**

Entities that can assume this role under specified conditions.

```
1 [ {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Principal": "  
7         AWS": "arn:aws:iam::805171368357:root"  
8     },  
9     {"Action": "sts:AssumeRole"}  
10    ]  
11  ]  
12 }
```

**Edit trust policy**

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# 4. Cross-Account Data Flow via AWS Lambda

The screenshot shows the AWS Lambda function editor for the 'IoT\_To\_AccountB\_Bridge' function. The code source is 'lambda\_function.py'. The code uses the boto3 library to assume a role and interact with a DynamoDB table named 'PlantData'. A red circle highlights the line 'ROLE\_ARN = "arn:aws:iam::096938402672:role/ESP32\_Receiver\_Role"'. A green circle highlights the 'sts\_client = boto3.client('sts')' line.

```
lambda_function.py
1 import boto3
2 import time
3 import json
4 from decimal import Decimal
5
6 def lambda_handler(event, context):
7     ROLE_ARN = "arn:aws:iam::096938402672:role/ESP32_Receiver_Role"
8
9     sts_client = boto3.client('sts')
10    try:
11        assumed_role = sts_client.assume_role(
12            RoleArn=ROLE_ARN,
13            RoleSessionName="IoTCrossAccountSession"
14        )
15    except Exception as e:
16        print(f"Error assuming role: {e}")
17        raise e
18
19    creds = assumed_role['Credentials']
20    dynamodb = boto3.resource(
21        'dynamodb',
22        aws_access_key_id=creds['AccessKeyId'],
23        aws_secret_access_key=creds['SecretAccessKey'],
24        aws_session_token=creds['SessionToken']
25    )
26
27    table = dynamodb.Table('PlantData')
28
29    item = json.loads(json.dumps(event), parse_float=Decimal)
30
31    if 'timestamp' not in item:
```

**Code source** Info

EXPLORER IOT\_TO\_ACCOUNTB\_BRIDGE lambda\_function.py

DEPLOY Current Deploy (Ctrl+Shift+U) Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED] Create new test event

ENVIRONMENT VARIABLES

Lambda Deployed 0 △ 0 Amazon Q

Line numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31

Open in Visual Studio Code Upload from

Info Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

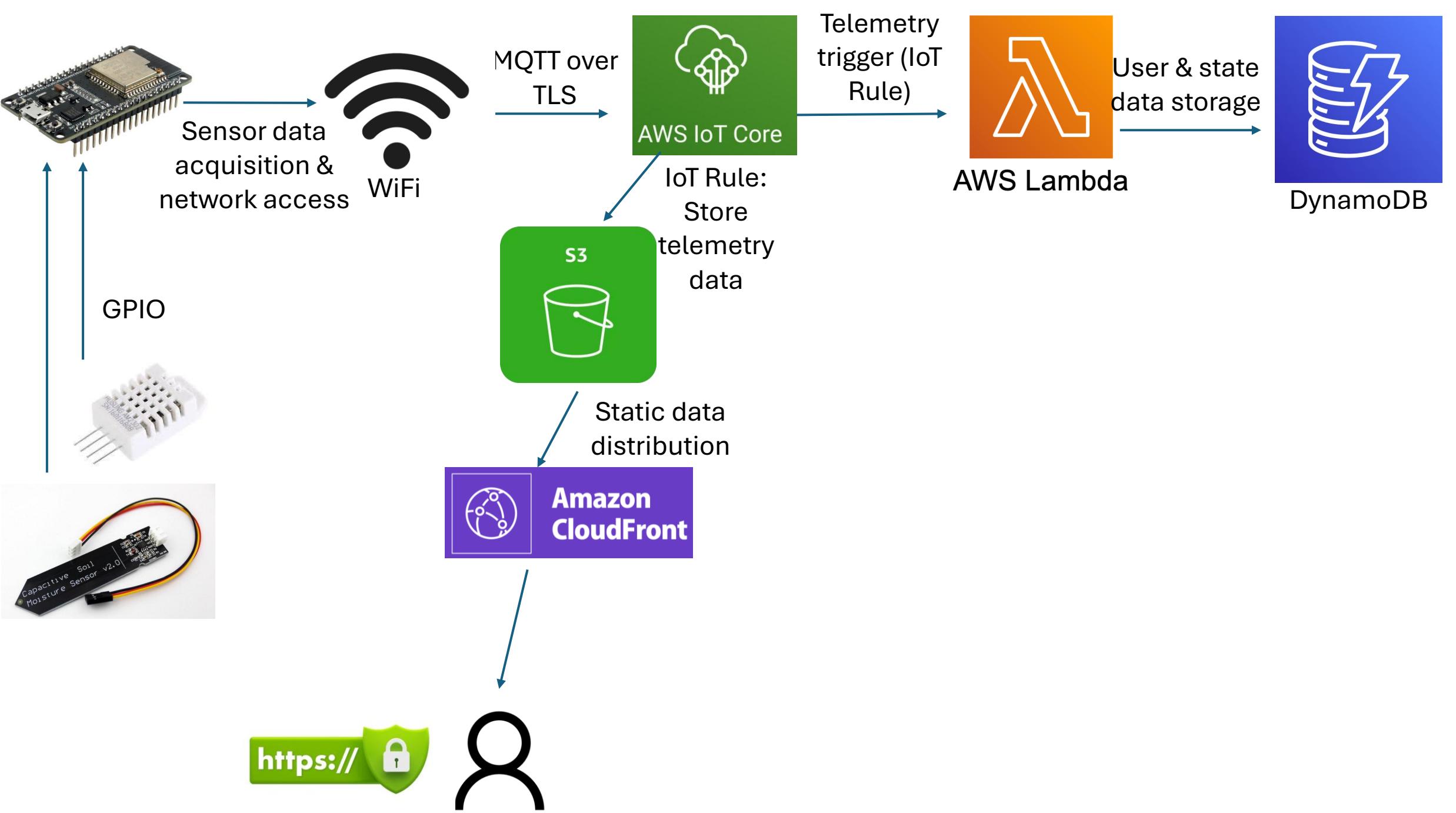
In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

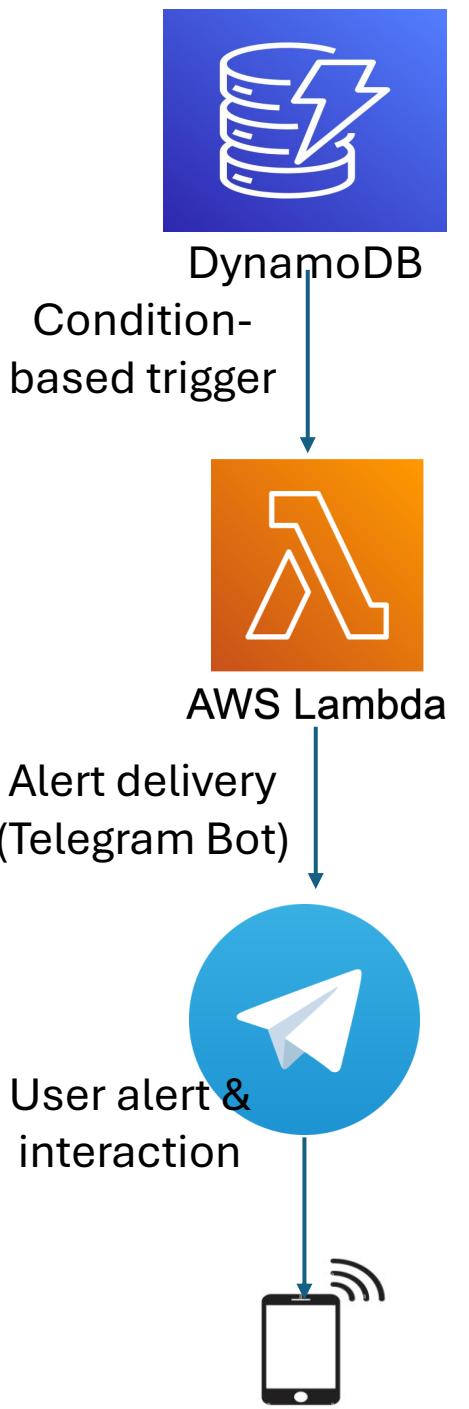
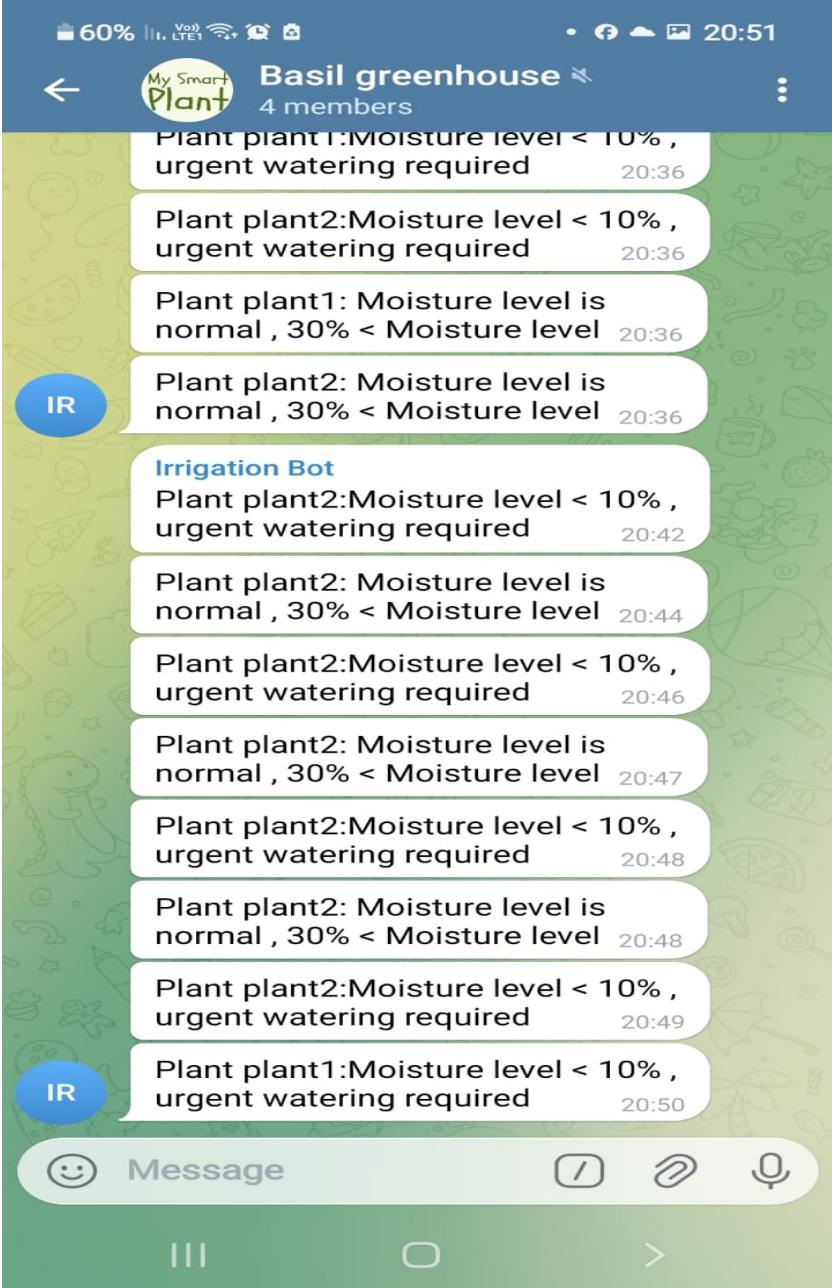
Learn more Start tutorial

CloudShell Feedback Console Mobile App

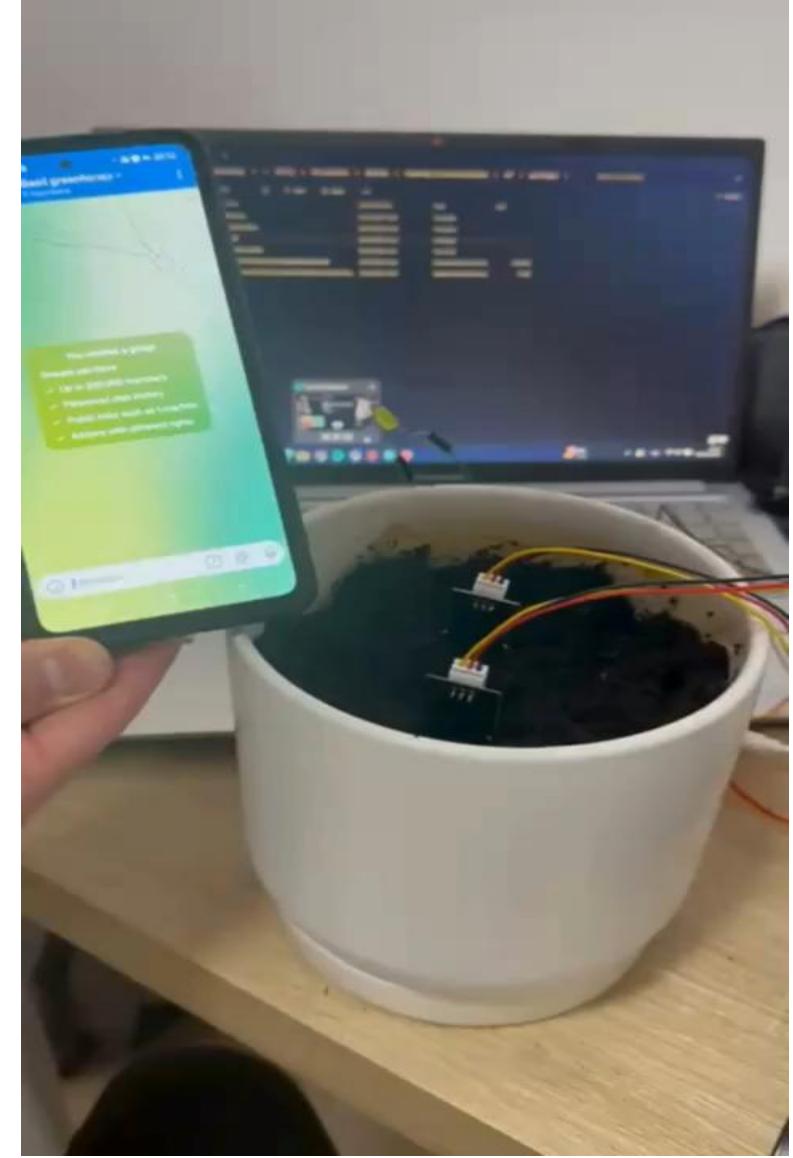
© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

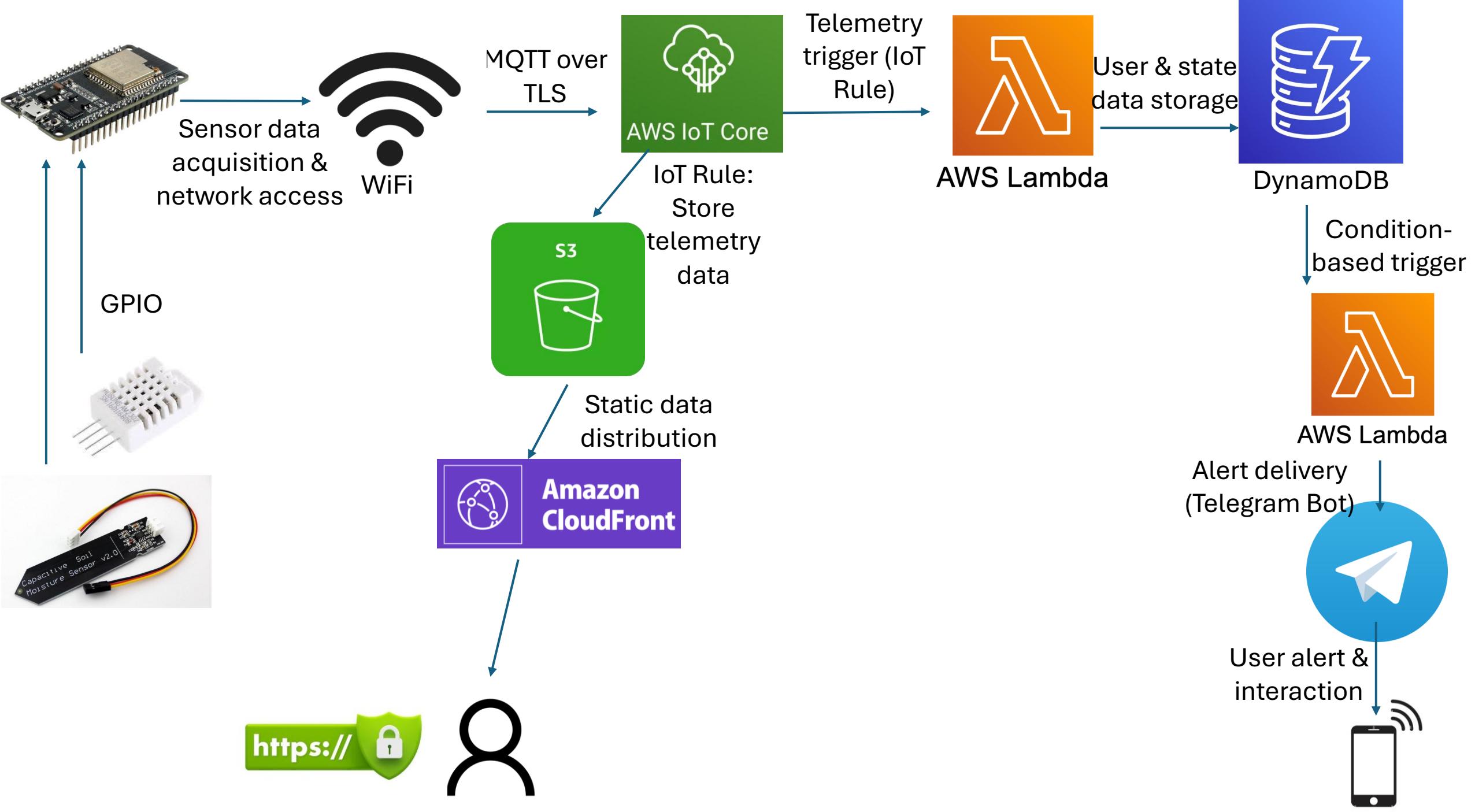


# Alert ! ! !

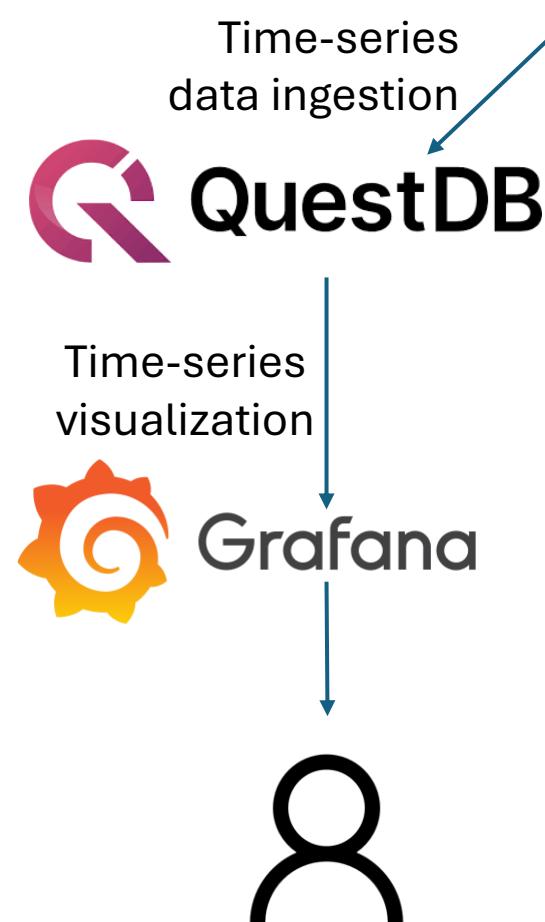


# Demonstration Video

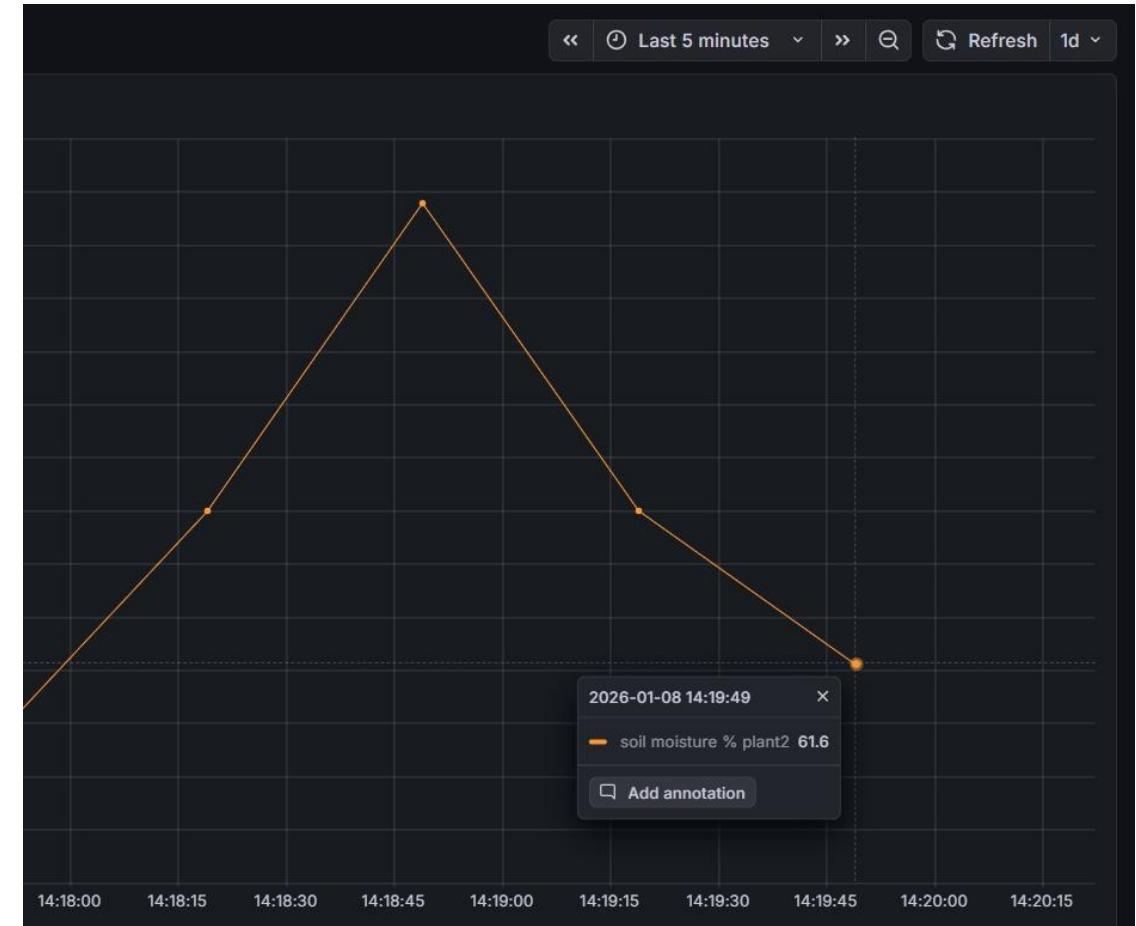
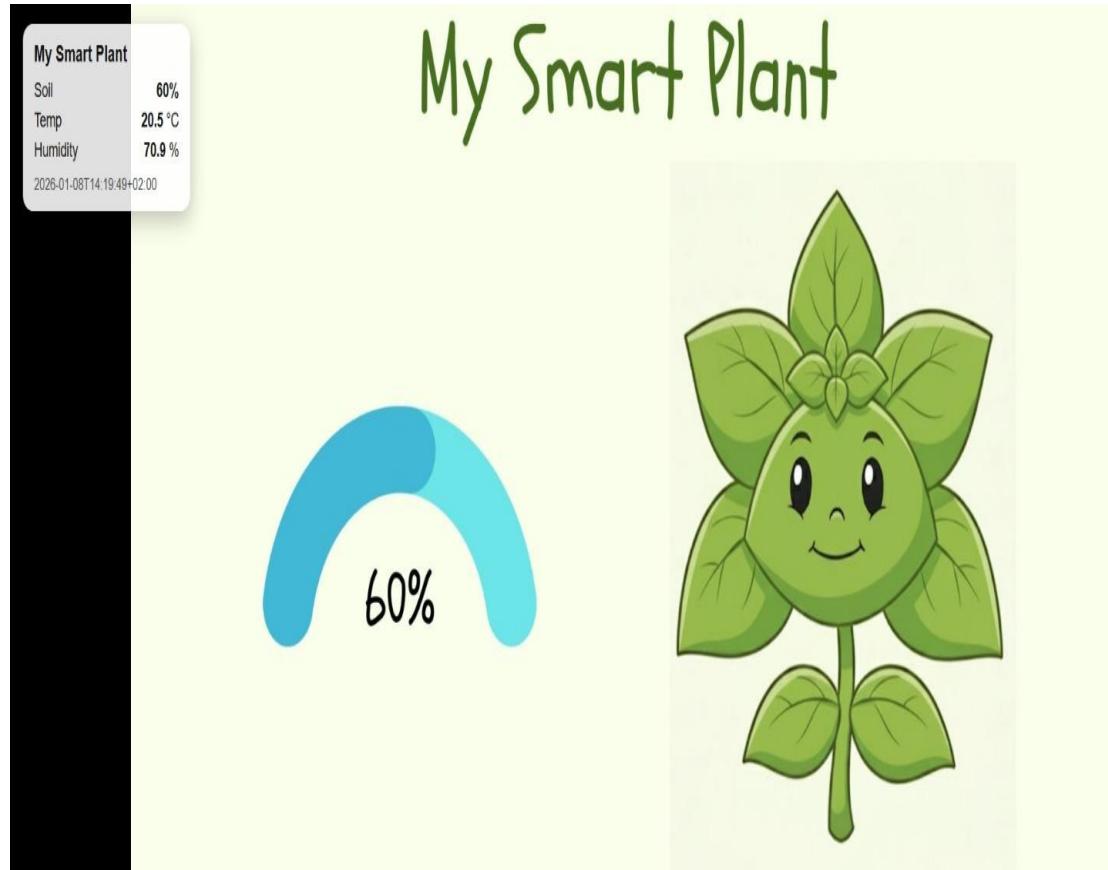


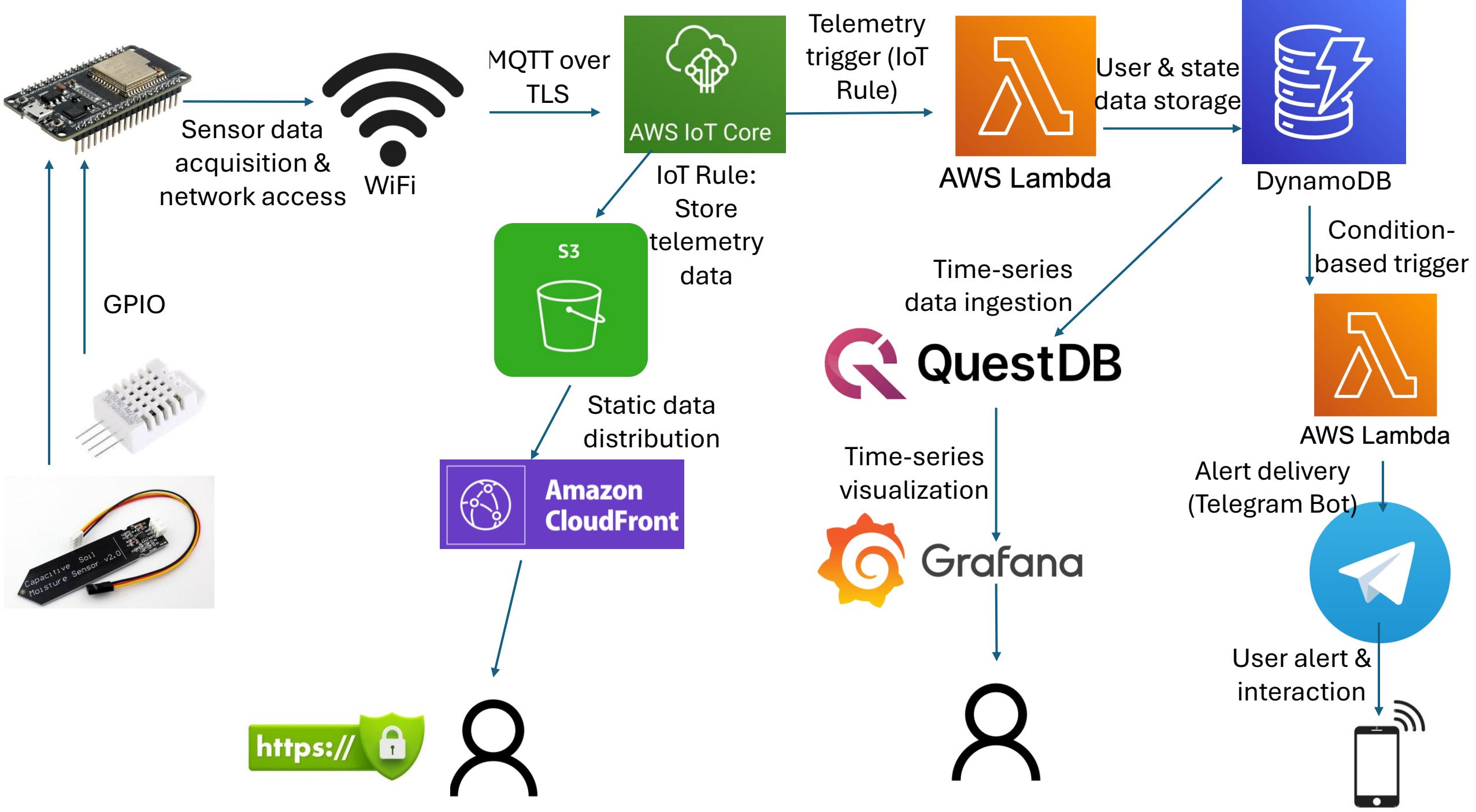


# Real-Time Data Analytics & Visualization



# RT visualization VS. Website





# Challenges

## 1. Limited GPIO / ADC Resources

Solution: Use additional ESP32 controllers that operate in parallel and publish data using separate identities and topics.

## 2. Unstable Wiring and Sensor Disconnections

Solution: Use more reliable physical connections such as wider prototyping boards or dedicated PCBs with fixed wiring.

[Plant1](#) [Plant2](#) [All\\_plants\\_data](#)

## 3. Intermittent WiFi / Cloud Connectivity

Solution: Provide local status indication and implement retry mechanisms with temporary local data buffering.

## 4. Noisy Analog Soil Moisture Measurements

Solution: Apply high-sample averaging and per-sensor calibration to improve measurement stability and accuracy.

# Project Cost – Components

- ESP32 DevKit V4– 16.48 ILS
- DHT22 – 5.12 ILS
- Jumpers – 7.31 ILS
- 5Capacitive soil moisture sensor – 16.48 ILS
- **Components total (USD) - \$14.53**

Completed Order date: Dec 9, 2025  
Order ID: 1117766493712291 [Copy](#) [Order details >](#)

**Choice | Shop1102186190 Store > ...**

**SPCS** 1-10PCS Capacitive Soil Moisture Sensor Module Corrosion Resistant 3.3... **Total: ₪ 16.48**

5PCS ₪ 8.99 x1 **Write a review**

Free returns • Fast delivery **Add to cart**

**Remove**

**EXIKA** 1-10PCS DHT22 With Cable  
Bluetooth WiFi  
CP2102 38PIN 4MB SPI Flash Low Power V4 Expansion Board

**CHANZON** 10x 5mm LED Diodes Red Green Blue Yellow Orange White

**MB-102 Mb102 Choice** -%50 ₪ 14.61 ₪ 7.31  
+900 | 4.7 ★★★★☆  
+5,000 | 4.9 ★★★★☆  
משלוח חינם  
תוספת 2% הנחה עם מטבעות AliExpress  
בראש דירוגי המכירות ב-AliExpress

**MB-102 Mb102 Choice** -%29 ₪ 13.25 ₪ 9.36  
+5,000 | 4.8 ★★★★☆  
משלוח חינם  
תוספת 2% הנחה עם מטבעות AliExpress  
בראש דירוגי המכירות ב-AliExpress

**משלוח חינם**

# Project Cost – AWS

Approximate usage for 10 days use (Jan):

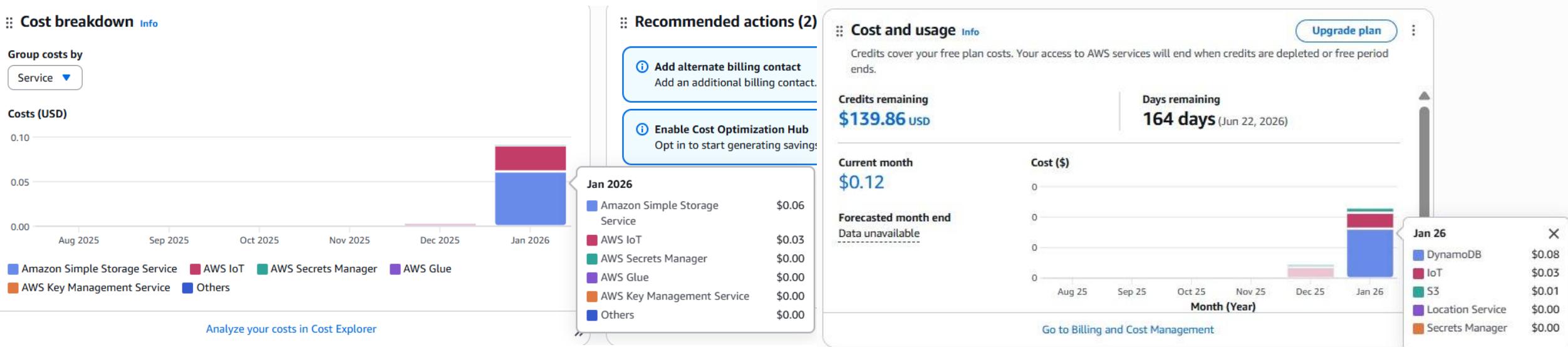
- Amazon Simple Storage Service (S3) – \$0.06
- AWS IoT – \$0.03

User 2:

- AWS IoT - \$0.03
- DynamoDB – \$0.08

Approximate usage per month:

- Amazon Simple Storage Service (S3) – \$0.18
- AWS IoT – \$0.18
- DynamoDB - \$0.24



# Project Cost – Total Cost (2 plants)

- Components - \$14.53
- AWS Services (per month) – \$0.6
- **Total - \$15.13**

\* 3.15 ILS = 1 USD (11/01/2026)



# Summary

- Designed and implemented an IoT-based smart irrigation monitoring system using ESP32
- Enabled local decision making with offline indication and cloud-based monitoring
- Implemented secure data ingestion, visualization, user notification and data analysis using AWS IoT, S3, CloudFront AWS Lambda, DynamoDB, Grafana, QuestDB and Telegram
- Demonstrated system scalability by supporting multiple plants with independent data pipelines
- Combined low hardware cost with minimal cloud expenses, suitable for home and small-scale use

# References

1. Kaur, G., Upadhyaya, P., & Chawla, P. (2023). Comparative analysis of IoT-based controlled environment and uncontrolled environment plant growth monitoring system for hydroponic indoor vertical farm. *Environmental Research*, 222, Article 115313.  
<https://doi.org/10.1016/j.envres.2023.115313>
2. <https://he.aliexpress.com>
3. <https://www.google.com/finance/>

My Smart  
Plant



**Thank You For Listening**

My Smart  
Plant



# Q&A