**Project Overview**

This project focuses on creating an IoT-based environmental monitoring system to collect and visualize environmental data such as temperature and humidity. The system utilizes a Raspberry Pi 5, Pico W, DHT11 sensor, and several software tools such as Docker, Node-RED, MQTT, Grafana, and InfluxDB to build an efficient data collection and monitoring system.

**Hardware Components**

* **Raspberry Pi 5**: Serves as the primary computing unit for handling Docker, running Node-RED, MQTT, and the database services.
* **Raspberry Pi Pico W**: Collects data from the sensor and communicates with the Raspberry Pi 5 using MQTT protocol.
* **DHT11 Sensor (3-pin)**: Measures environmental parameters such as temperature and humidity.
* **Breadboard**: A platform for connecting the Pico W to the DHT11 sensor.
* **Jumper Wires**: Used to connect the Pico W with the DHT11 sensor via the breadboard.
* **27W Power Supply**: Provides power to the Raspberry Pi 5.
* **5W Power Supply for Pico W**: Provides power to the Pico W.
* **Micro USB Cable**: Used to connect the Raspberry Pi Pico W to the power supply or Raspberry Pi.

**Software Tools**

* **MicroPython**: Firmware for programming the Pico W to read sensor data.
* **Python**: Used to manage the overall operation on the Raspberry Pi 5.
* **Docker**: Containerization tool for deploying and managing IoT Stack services.
* **IoT Stack**: A collection of services (Node-RED, MQTT, InfluxDB, Grafana) used for data collection, messaging, storage, and visualization.
* **Portainer**: A tool for managing Docker containers.
* **Node-RED**: A flow-based development tool used to wire up inputs and outputs from the IoT system.
* **Eclipse Mosquitto MQTT**: A lightweight messaging protocol to send sensor data from Pico W to the Raspberry Pi 5.
* **InfluxDB**: A time-series database used to store environmental data.
* **Grafana**: A visualization tool used to create real-time dashboards for viewing temperature and humidity data.

**1. Hardware Setup**

**1.1 Connecting the DHT11 Sensor to the Raspberry Pi Pico W**

**Pin Connections:**

* **DHT11 VCC** → **Pico W 3.3V**
* **DHT11 GND** → **Pico W GND**
* **DHT11 Data** → **Pico W GPIO Pin**

**Steps:**

1. Plug the DHT11 sensor into the breadboard.
2. Use jumper wires to connect the sensor to the appropriate GPIO pins of the Pico W.
3. Ensure that the Pico W is powered via the 5W power supply or through a connection to the Raspberry Pi using a Micro USB cable.

**2. Software Setup**

**2.1 Programming the Raspberry Pi Pico W**

The Pico W will use MicroPython to collect sensor data and send it via MQTT.

**Steps:**

1. Install the MicroPython firmware on the Pico W.
2. Write a MicroPython script to:
   * Initialize the DHT11 sensor.
   * Collect temperature and humidity data.
   * Send the collected data to the MQTT broker (Eclipse Mosquitto running on Raspberry Pi 5).

Example MicroPython code:

|  |
| --- |
| import machine  import time  import dht  from umqtt.simple import MQTTClient  # Pin setup for DHT11  sensor = dht.DHT11(machine.Pin(15))  # MQTT setup  mqtt\_server = 'YOUR\_RPI\_IP'  client\_id = 'pico\_w'  topic = b'environment/sensor'  def connect\_mqtt():      client = MQTTClient(client\_id, mqtt\_server)      client.connect()      return client  client = connect\_mqtt()  while True:      sensor.measure()      temp = sensor.temperature()      humidity = sensor.humidity()      message = f'Temp: {temp}C, Humidity: {humidity}%'        client.publish(topic, message)      time.sleep(10) |

**2.2 Setting Up Docker and IoT Stack on Raspberry Pi**

**Steps:**

1. **Install Docker** on the Raspberry Pi.
2. **Install IoT Stack**:
   * Follow the IoT Stack installation instructions to include Node-RED, InfluxDB, Grafana, and Eclipse Mosquitto MQTT.
3. **Deploy Docker Containers**: Use docker-compose.yml for IoT stack services.

Example docker-compose.yml:

|  |
| --- |
| version: '3'  services:    mqtt:      image: eclipse-mosquitto      ports:        - "1883:1883"        - "9001:9001"      volumes:        - ./mosquitto/config:/mosquitto/config        - ./mosquitto/data:/mosquitto/data        - ./mosquitto/log:/mosquitto/log      influxdb:      image: influxdb:latest      ports:        - "8086:8086"      environment:        - INFLUXDB\_DB=iot\_data        - INFLUXDB\_ADMIN\_USER=admin        - INFLUXDB\_ADMIN\_PASSWORD=password    grafana:      image: grafana/grafana      ports:        - "3000:3000"      nodered:      image: nodered/node-red      ports:        - "1880:1880" |

**2.3 Configuring Node-RED**

**Steps:**

1. **Access Node-RED UI**: Navigate to http://<RaspberryPi\_IP>:1880 in your browser.
2. **Create a Flow**:
   * Use the MQTT input node to subscribe to the topic (e.g., environment/sensor).
   * Store the data in InfluxDB using the InfluxDB node.
   * Visualize the data in Grafana by connecting Grafana to the InfluxDB database.

**3. Visualizing Data with Grafana**

**3.1 Connecting Grafana to InfluxDB**

1. Access the Grafana web interface at http://<RaspberryPi\_IP>:3000.
2. Add a new data source and choose **InfluxDB**.
3. Set the connection details (e.g., database name, username, password).

**3.2 Creating a Dashboard**

1. Create a new dashboard in Grafana.
2. Use the query editor to fetch data (temperature, humidity) from InfluxDB.
3. Display real-time data using line charts or other visualizations.

**4. Conclusion**

By setting up this IoT-based environmental monitoring system, you can continuously monitor environmental data and display it in real-time using Grafana dashboards. This system is scalable, and additional sensors can be added by modifying the MicroPython code on the Pico W.