

The background of the entire page is a dark navy blue. It is decorated with numerous thin, light blue lines that flow and curve across the space, creating a sense of motion and depth. These lines are more densely packed on the left and right sides, framing the central text area.

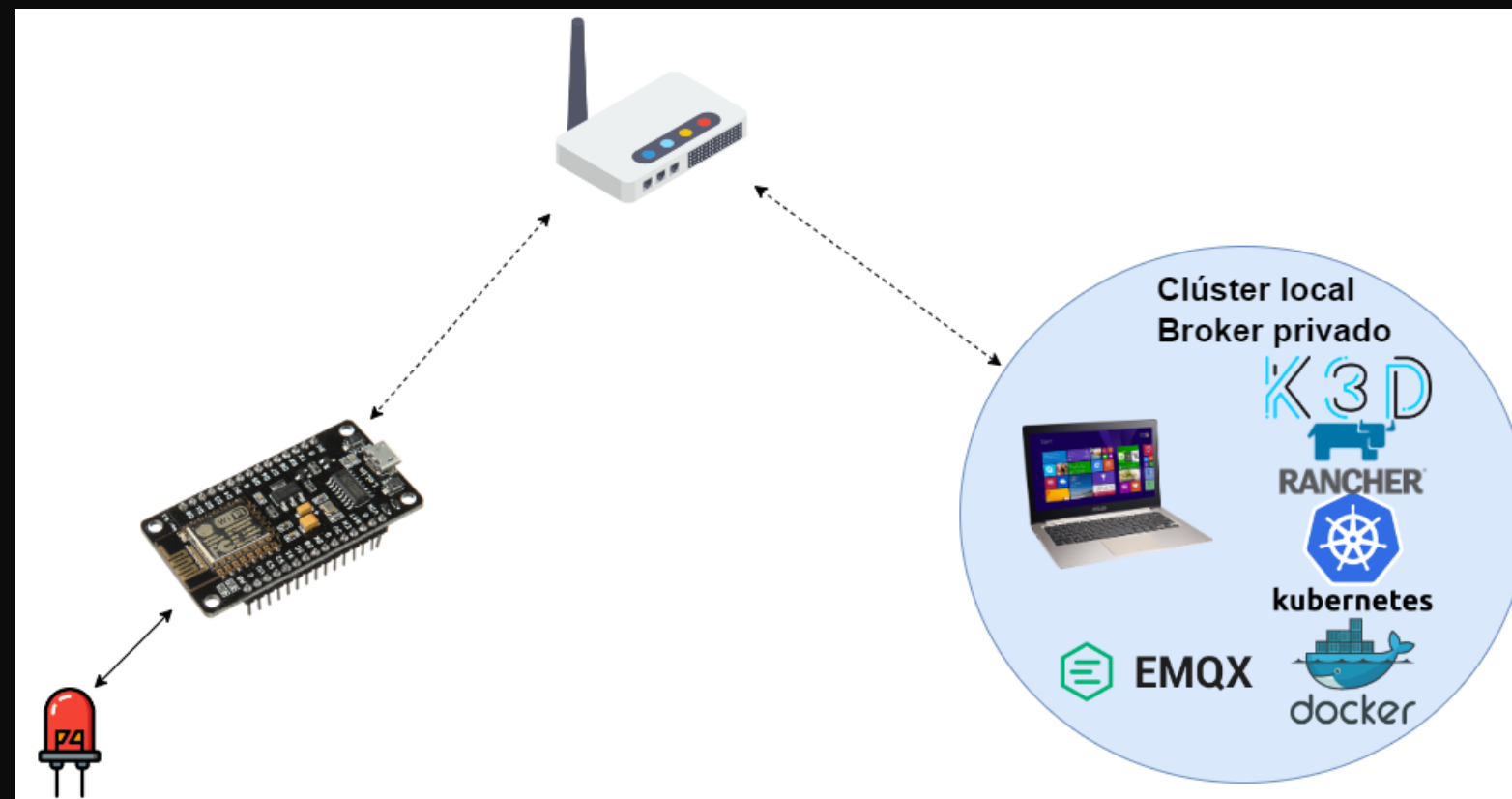
VICTOR MANUEL VELASCO HERNÁNDEZ
JEANETTE MURILLO CORTES

COMPUTACIÓN TOLERANTE A FALLAS
CALENDARIO 2022A
DR. MICHEL EMMANUEL LÓPEZ FRANCO

PROYECTO - IoT

INTRODUCCIÓN

COMPUTACIÓN TOLERANTE



Conexión de una placa NodeMCU8266 a un clúster de Kubernetes para dar una primera etapa a un sistema IoT.

Avances

CÓDIGO

```
NodeMCU
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
int var;
String resultS;
// WiFi
const char *ssid = "dlink-D82C"; // Enter your WiFi name
const char *password = "qpmws68912"; // Enter WiFi password

// MQTT Broker
const char *mqtt_broker = "192.168.0.101";
const char *topic = "esp8266/test";
const char *mqtt_username = "admin";
const char *mqtt_password = "public";
const int mqtt_port = 1883;

WiFiClient espClient;
PubSubClient client(espClient);
void startWifi(){
  // connecting to a WiFi network
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
}

void setup() {
  // Set software serial baud to 115200;
  pinMode(2, OUTPUT);
  Serial.begin(9600);
  startWifi();
  //connecting to a mqtt broker
  client.setServer(mqtt_broker, mqtt_port);
  client.setCallback(callback);
  if(!client.connected()) {
    reconnect();
  }
}
```

```
void reconnect() {
  while (!client.connected()) {
    String client_id = "esp8266-client";
    client_id += String(WiFi.macAddress());
    Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
    if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
      Serial.println("Public emqx mqtt broker connected");
    } else {
      Serial.print("failed with state ");
      Serial.print(client.state());
      delay(2000);
    }
  }
  // publish and subscribe
  client.publish(topic, "hello emqx");
  client.subscribe(topic);
}

void callback(char *topic, byte *payload, unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");
  char payload_string[length + 1];

  int resultI;

  memcpy(payload_string, payload, length);
  payload_string[length] = '\0';
  resultI = atoi(payload_string);

  var = resultI;

  resultS = "";

  for (int i=0;i<length;i++) {
    resultS= resultS + (char)payload[i];
  }
  Serial.println();
  Serial.println("-----");
}
```

```
void loop() {
  if(!client.connected()) {
    reconnect();
  }
  client.loop();
  if(resultS!="")
    Serial.println(resultS);
  if(var == 1)
  {
    digitalWrite(2, LOW);
  }
  else if (var == 0)
  {
    digitalWrite(2, HIGH);
  }
  //delay(5000);
}
```

Dashboard

EMQ

Dashboard

Monitor

Clients

Topics

Subscriptions

Rule Engine

Analysis

Plugins

Modules

Tools

Settings

General

admin

Overview

emqx-deployment-64d6bdc847

Broker

System Name

EMQ X Broker

Version

4.1-rc.1

Uptime

2 hours, 13 minutes, 19 seconds

System Time

2022-05-20 09:09:41

Nodes(1)

Name	Erlang/OTP Release	Erlang Processes (used/available)	CPU Info (1load/5load/15load)	Memory Info (used/total)	MaxFds	Status
emqx-deployment-64d6bdc847-6cq74@10.42.0.12	R22/10.7.1	397 / 2097152	0.07 / 0.20 / 0.18	113.24M / 154.59M	1048576	Running

Stats(1)

Name	Connections	Topics	Retained	Sessions	Subscriptions	Subscriptions Shared
------	-------------	--------	----------	----------	---------------	----------------------

Dashboard

localhost:18083/#/websocket

EMQ

Dashboard

Topics

Subscriptions

Rule Engine

Analysis

Plugins

Modules

Tools

Websocket

HTTP API

Settings

General

admin

Subscribe

Messages

Topic

testtopic

Messages

{ "msg": "Hello, World!" }

QoS

0

Retained

send

Messages already sent

Messages received

Messages	Topic	QoS	Time	Messages	Topic	QoS	Time
No Data							

Dashboard

localhost:18083/#/clients

EMQ

Dashboard

Monitor

Clients

Topics

Subscriptions

Rule Engine

Analysis

Plugins

Modules

Tools

Settings

General

admin

Clients

emqx-deployment-64d6bdc847

Client ID

Username

Search

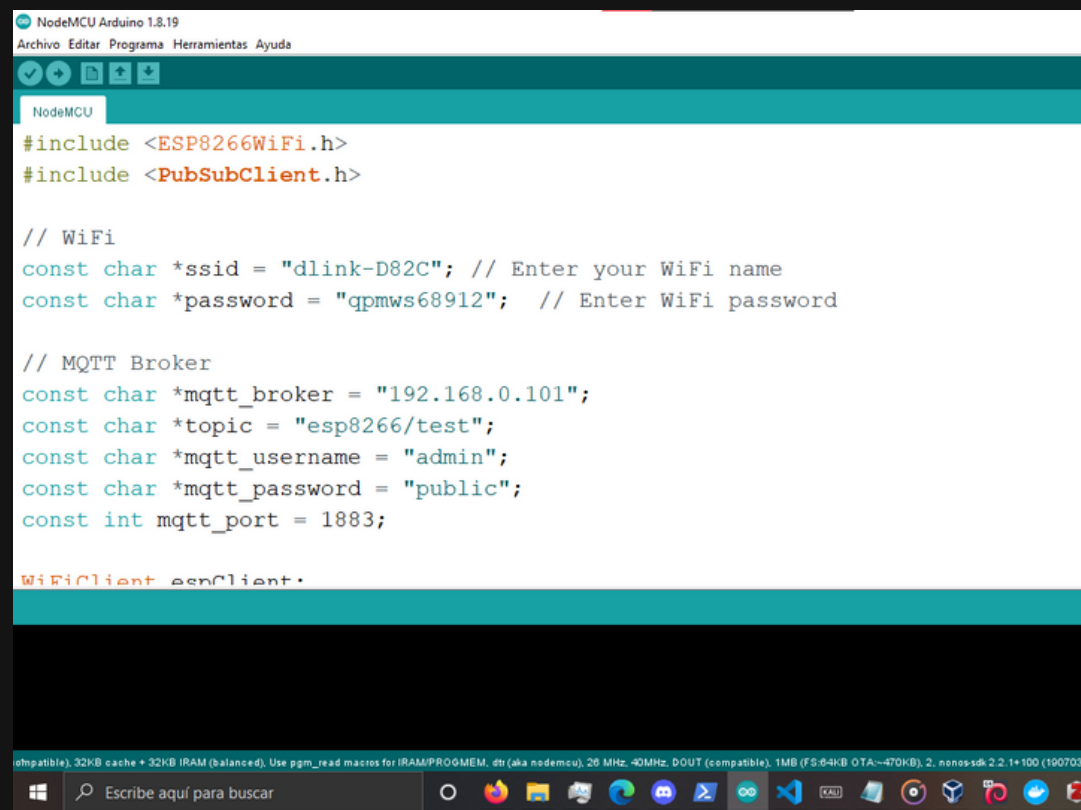
Reset

Expand

Client ID	Username	IP Address	Keepalive (s)	Expiry Interval(s)	Subscriptions Count	Connect Status	Create	Operation
No Data								

Avances

VIDEO



```
NodeMCU Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

NodeMCU
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// WiFi
const char *ssid = "dlink-D82C"; // Enter your WiFi name
const char *password = "qpmws68912"; // Enter WiFi password

// MQTT Broker
const char *mqtt_broker = "192.168.0.101";
const char *topic = "esp8266/test";
const char *mqtt_username = "admin";
const char *mqtt_password = "public";
const int mqtt_port = 1883;

WiFiClient espClient;
```

Link:

https://drive.google.com/file/d/1FMGfbisNaSB2YlhmFSA-l6AZJ_EX42pp/view?usp=sharing