PROJECT DEVELOPMENT PHASE

DELIVERY OF SPRINT-2

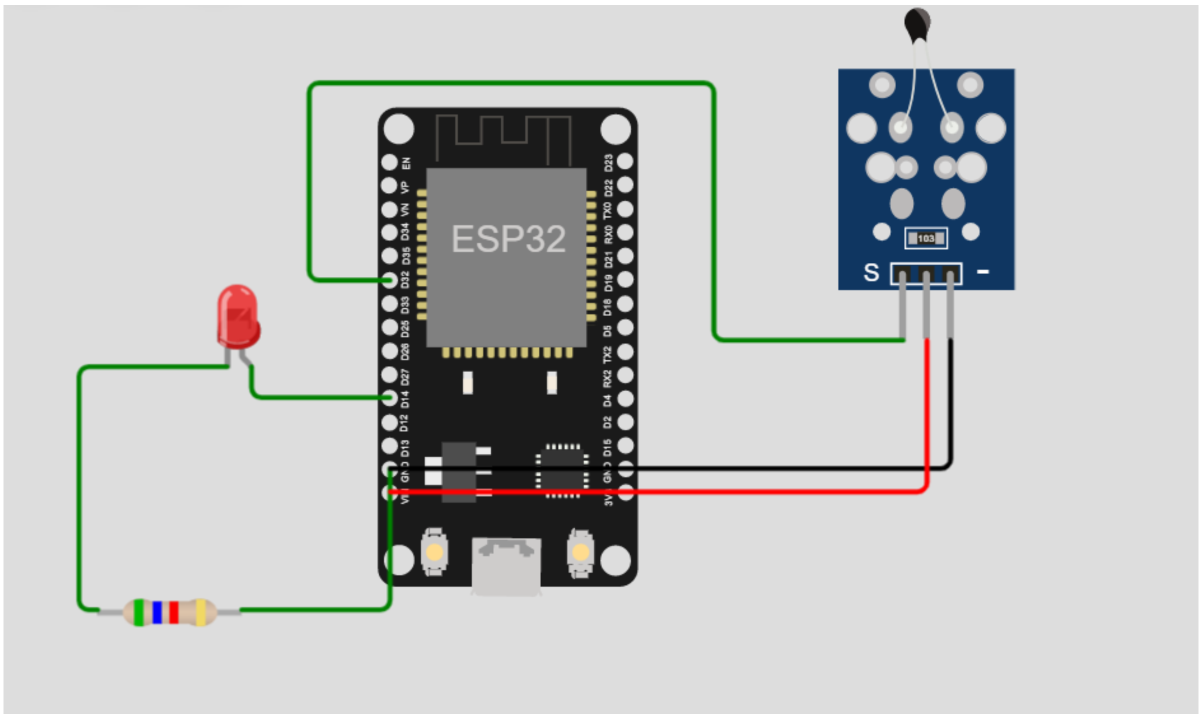
Team ID	PNT2022TMID14349
Project Name	Industry- Specific Intelligent Fire Management System

CODE:

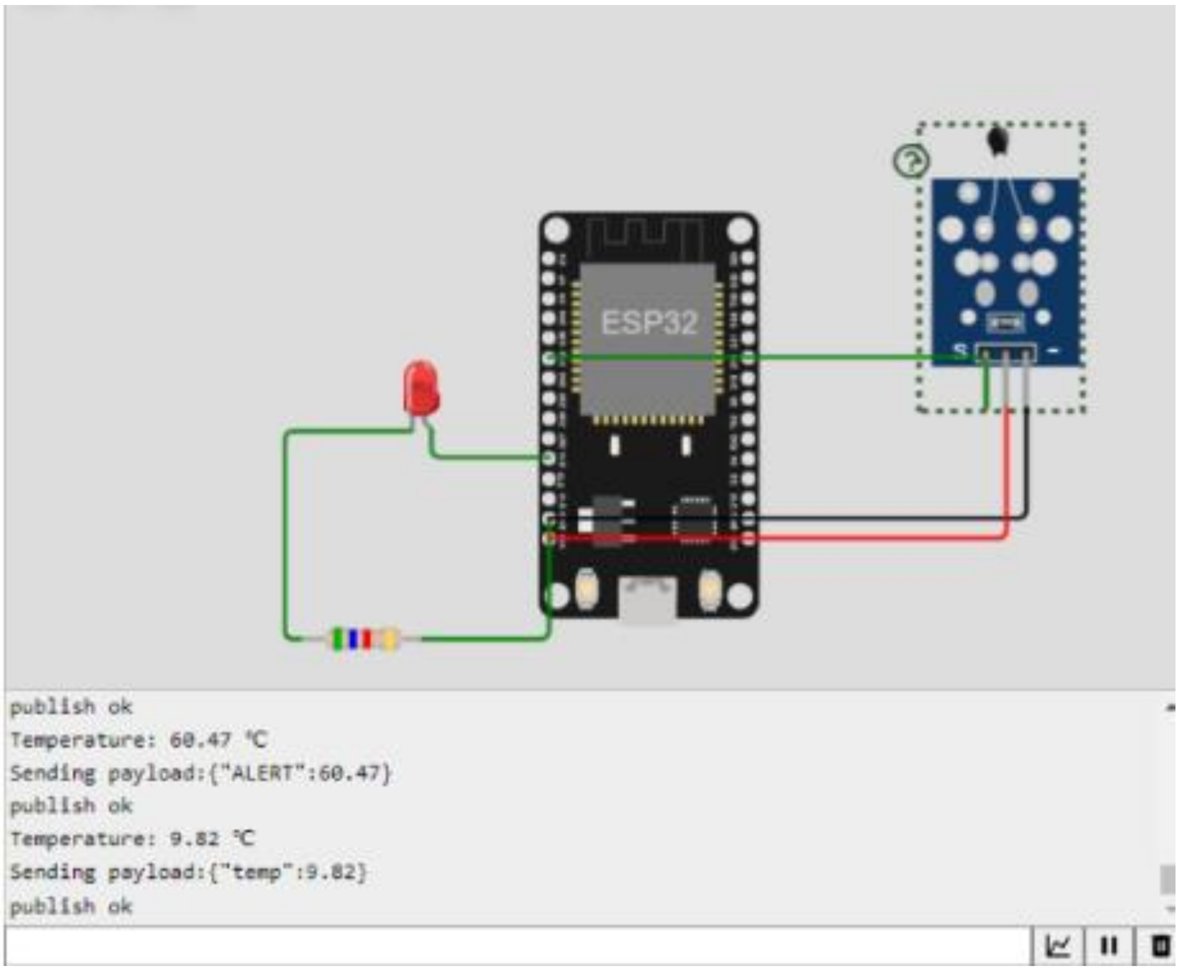
```
#include <WiFi.h>
#include <PubSubClient.h>

#define temp_pin 15

void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);

#define ORG "he8juu"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "12"
#define TOKEN "12345678"
String data3;

char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/Data/fmt/json";
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server,1883,callback,wifiClient);



 // should match the Beta Coefficient of the thermistor

void setup() {
  Serial.begin(9600);
  analogReadResolution(10);
  pinMode(32,INPUT);
  pinMode(14,OUTPUT);

  wificonnect();
  mqttconnect();
```

```
}
void loop() {
  const float BETA = 3950; // should match the Beta Coefficient of the
thermistor
int analogValue = analogRead(A4);
float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) -
273.15;
  //float temp = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 /
298.15) - 273.15;
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" ℃");
  if(temp>=35){
    PublishData2(temp);
    digitalWrite(14, HIGH);
  }else{
    digitalWrite(14, LOW);
    PublishData1(temp);
}
 delay(1000);
  if(!client.loop()){
    mqttconnect();
  }

  //delay(2000);
}
void PublishData1(float tem){
  mqttconnect();
  String payload= "{\"temp\":";
  payload += tem;
  payload+="}";

  Serial.print("Sending payload:");
  Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
void PublishData2(float tem){
  mqttconnect();
  String payload= "{\"ALERT\":";
  payload += tem;
  payload+="}";
  Serial.print("Sending payload:");
  Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
void mqttconnect(){
  if(!client.connected()){
    Serial.print("Reconnecting to");
    Serial.println(server);
    while(!!!client.connect(clientID, authMethod, token)){
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);
  while(WiFi.status()!=WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WIFI CONNECTED");
  Serial.println("IP address:");
  Serial.println(WiFi.localIP());
}

void initManagedDevice(){
  if(client.subscribe(subscribeTopic)){
    Serial.println((subscribeTopic));
    Serial.println("subscribe to cmd ok");
  }else{
    Serial.println("subscribe to cmd failed");
  }
}

void callback(char* subscribeTopic, byte* payload, unsigned int
payloadLength){
  Serial.print("callback invoked for topic:");
  Serial.println(subscribeTopic);
  for(int i=0; i<payloadLength; i++){
    data3 += (char)payload[i];
  }
  Serial.println("data:"+ data3);
  if(data3=="lighton"){
    Serial.println(data3);
    digitalWrite(14,HIGH);
  }else{
    Serial.println(data3);
    digitalWrite(14,LOW);
  }
  data3="";
}
```

CIRCUIT:

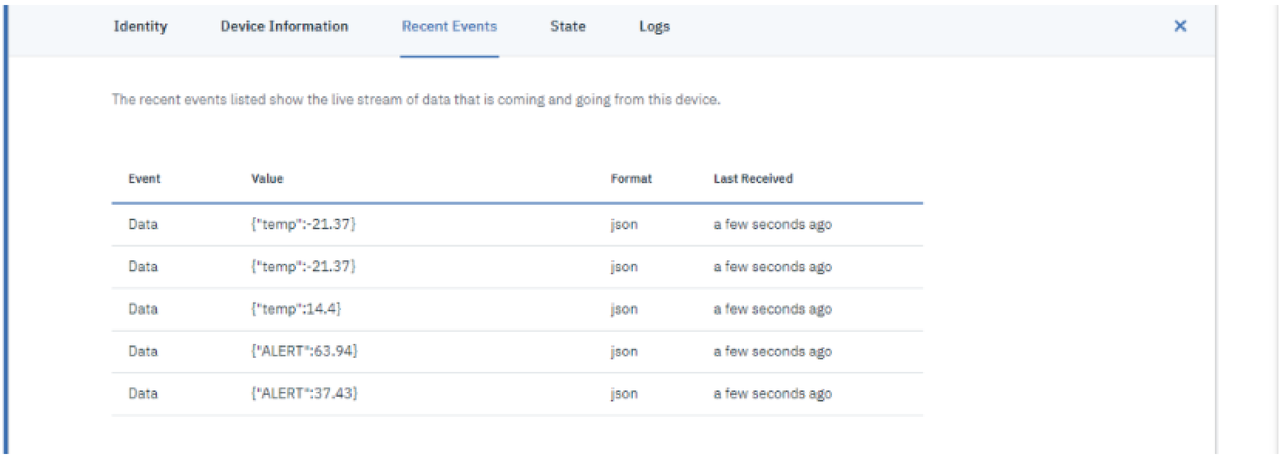


OUTPUT:



```
publish ok
Temperature: 60.47 °C
Sending payload:{"ALERT":60.47}
publish ok
Temperature: 9.82 °C
Sending payload:{"temp":9.82}
publish ok
```

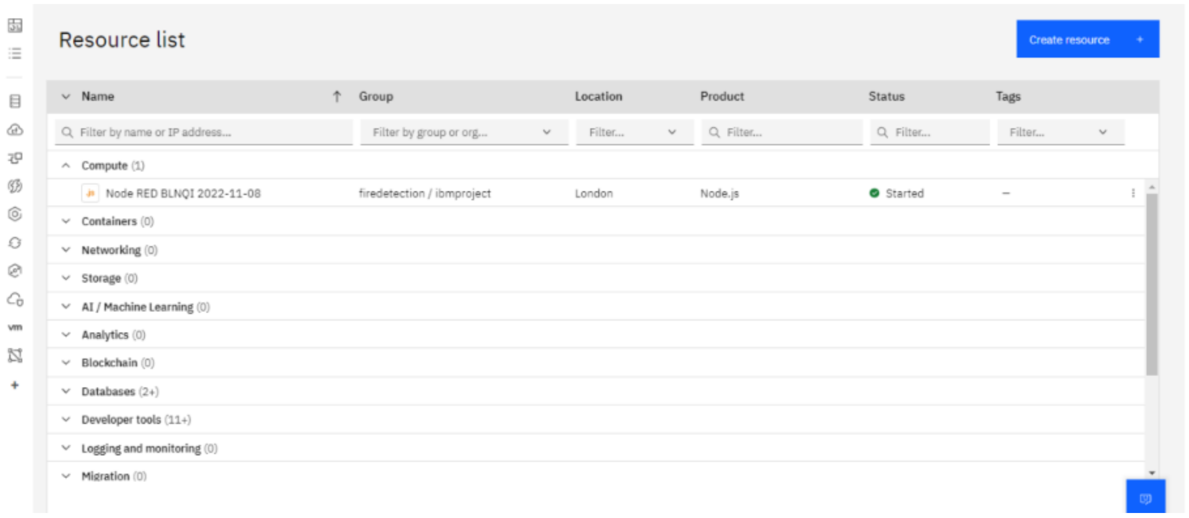
IBM CLOUD DATA:



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temp":-21.37}	json	a few seconds ago
Data	{"temp":-21.37}	json	a few seconds ago
Data	{"temp":14.4}	json	a few seconds ago
Data	{"ALERT":63.94}	json	a few seconds ago
Data	{"ALERT":37.43}	json	a few seconds ago

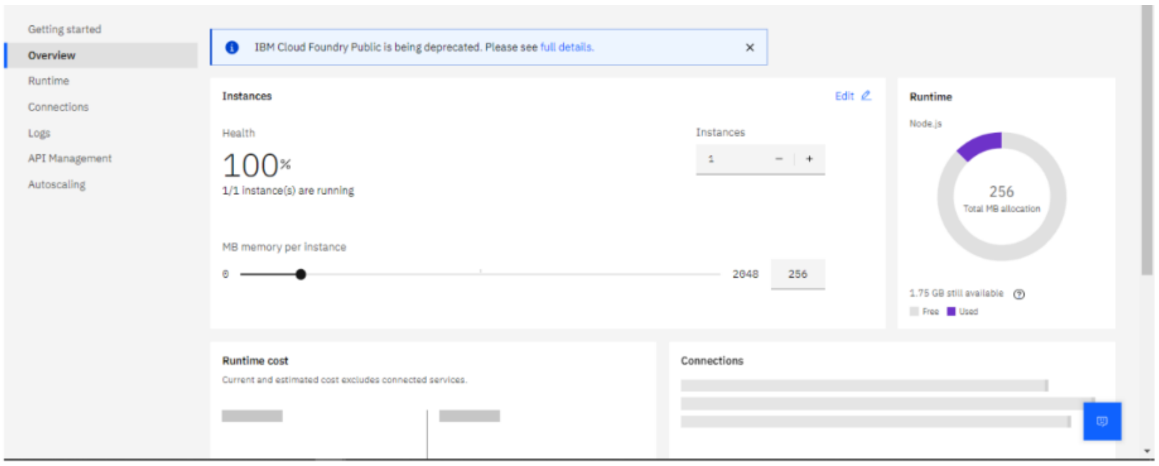
IBM CLOUD NODE-RED SERVICE:



Resource list

Create resource +

Name	Group	Location	Product	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Compute (1)					
Node RED BLNQI 2022-11-08	firedetection / ibmproject	London	Node.js	● Started	–
Containers (0)					
Networking (0)					
Storage (0)					
AI / Machine Learning (0)					
Analytics (0)					
Blockchain (0)					
Databases (2+)					
Developer tools (11+)					
Logging and monitoring (0)					
Migration (0)					



Getting started
Overview
Runtime
Connections
Logs
API Management
Autoscaling

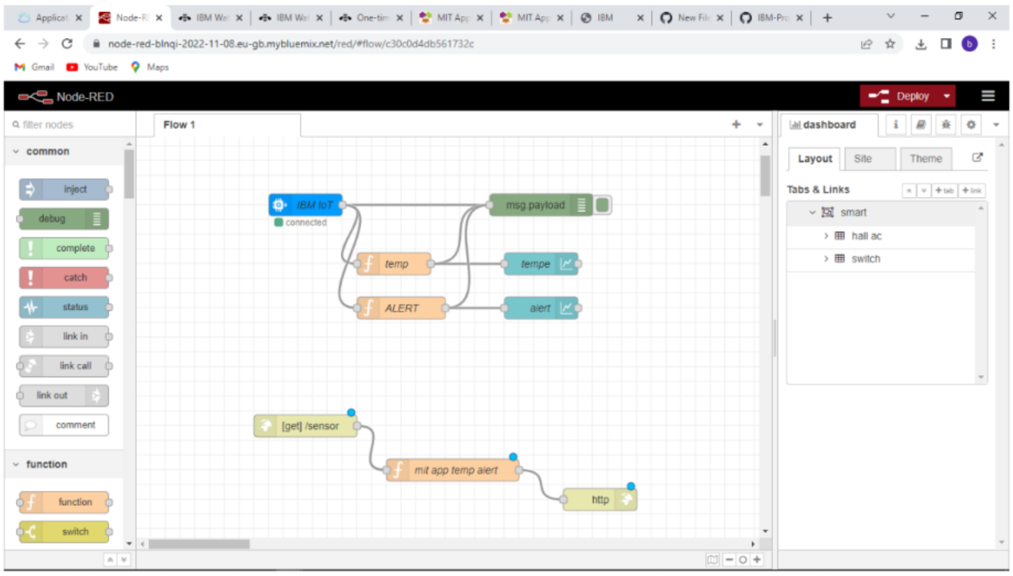
IBM Cloud Foundry Public is being deprecated. Please see full details.

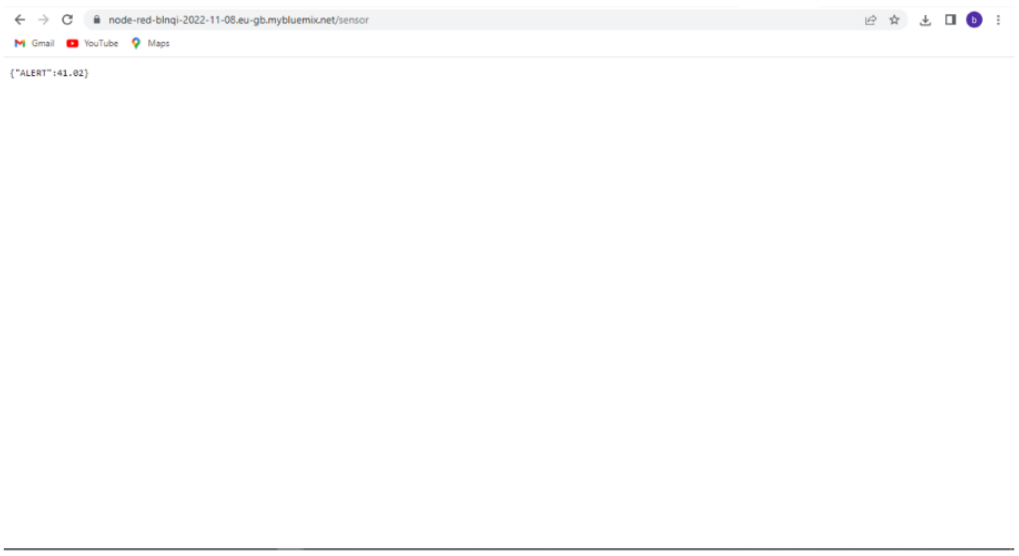
Instances
Edit
Health
100%
1/1 instance(s) are running
Instances
1
MB memory per instance
0 2048 256

Runtime
Node.js
256
Total MB allocation
1.75 GB still available
Free Used

Runtime cost
Current and estimated cost excludes connected services.

Connections




node-red-blnqi-2022-11-08.eu-gb.mybluemix.net/sensor
Gmail
YouTube
Maps

```
{"ALERT":41.02}
```