Project Development Phase

Gas Leakage Monitoring and Alerting System

TEAM ID: PNT2022TMID39307

TEAM MEMBERS

ROLE	TEAM MEMBERS NAME	ROLL NO
TEAM LEADER	NITHISH KUMAR.P	(422619104031)
TEAM MEMBER 1	SATHYA.R	(422619104038)
TEAM MEMBER 2	SUBASHINI.P	(422619104042)
TEAM MEMBER 3	SURYA.A	(422619104043)
TEAM MEMBER 4	VISHAL GANDHI.G	(422619104302)

Project Development - Delivery of Sprint-2

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Create And Configure IBM Cloud Services	USN- 2	Create Node-RED Service	10	Medium	SURYA A
Sprint-2	Develop A Python Script To Publish And Subscribe To IBM IoT Platform	USN- 3	Develop The Python Code	10	Medium	SUBASHINI P

Delivery

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-2	20	6 Days	31 Oct 2022	5 Nov 2022	20	5 Nov 2022

Hardware / Software requirements

Hardware:

Processor: Intel Pentium

Ram: 512 MB

Storage: 256 MB

Software:

IBM Watson Assistant, Browser

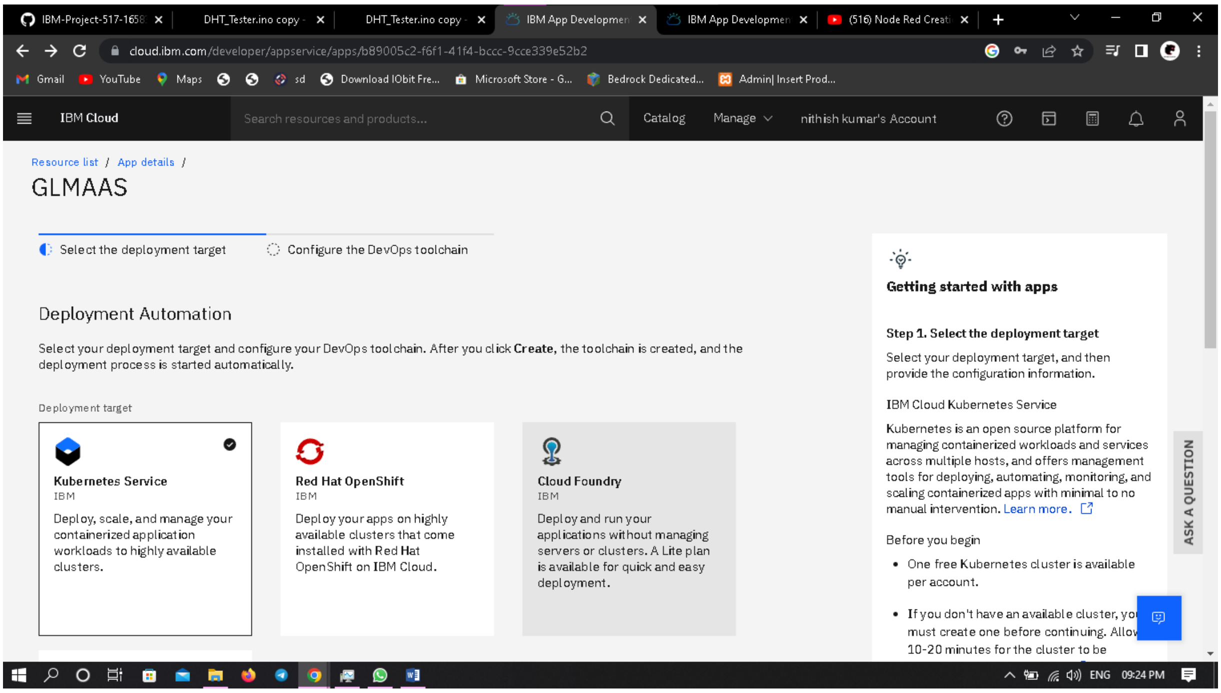
Create Node – Red Service

Task assigned: Surya A

Task started on: 31- 10 - 2022

Task completion date: 05 - 11 - 2022

Progress 1: Creating Node – RED service



Resource list / App details /

GLMAAS

Select the deployment target   Configure the DevOps toolchain

Deployment Automation

Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically.

Deployment target

Kubernetes Service
IBM
Deploy, scale, and manage your containerized application workloads to highly available clusters.

Red Hat OpenShift
IBM
Deploy your apps on highly available clusters that come installed with Red Hat OpenShift on IBM Cloud.

Cloud Foundry
IBM
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

Getting started with apps

Step 1. Select the deployment target

Select your deployment target, and then provide the configuration information.

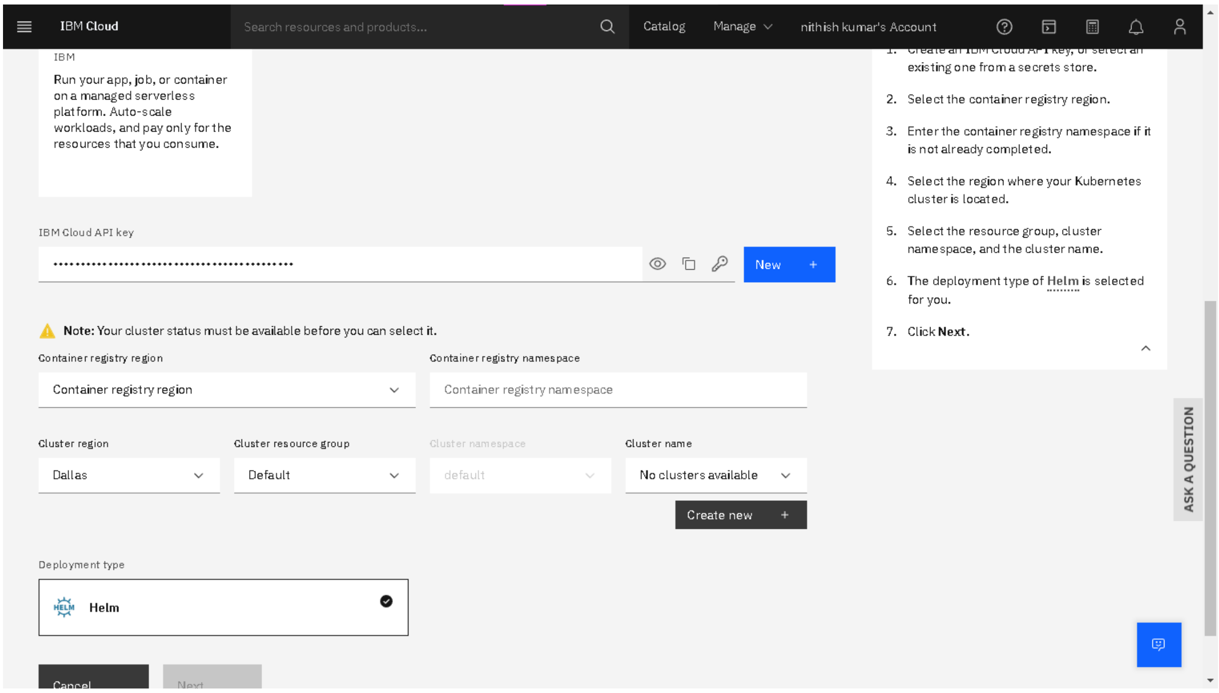
IBM Cloud Kubernetes Service

Kubernetes is an open source platform for managing containerized workloads and services across multiple hosts, and offers management tools for deploying, automating, monitoring, and scaling containerized apps with minimal to no manual intervention. Learn more.

Before you begin

- One free Kubernetes cluster is available per account.
- If you don't have an available cluster, yo must create one before continuing. Allov 10-20 minutes for the cluster to be

Progress 2: Changing configuration of Node – RED service



IBM

Run your app, job, or container on a managed serverless platform. Auto-scale workloads, and pay only for the resources that you consume.

IBM Cloud API key: ••

New +

Note: Your cluster status must be available before you can select it.

Container registry region: Container registry region

Container registry namespace: Container registry namespace

Cluster region: Dallas

Cluster resource group: Default

Cluster namespace: default

Cluster name: No clusters available

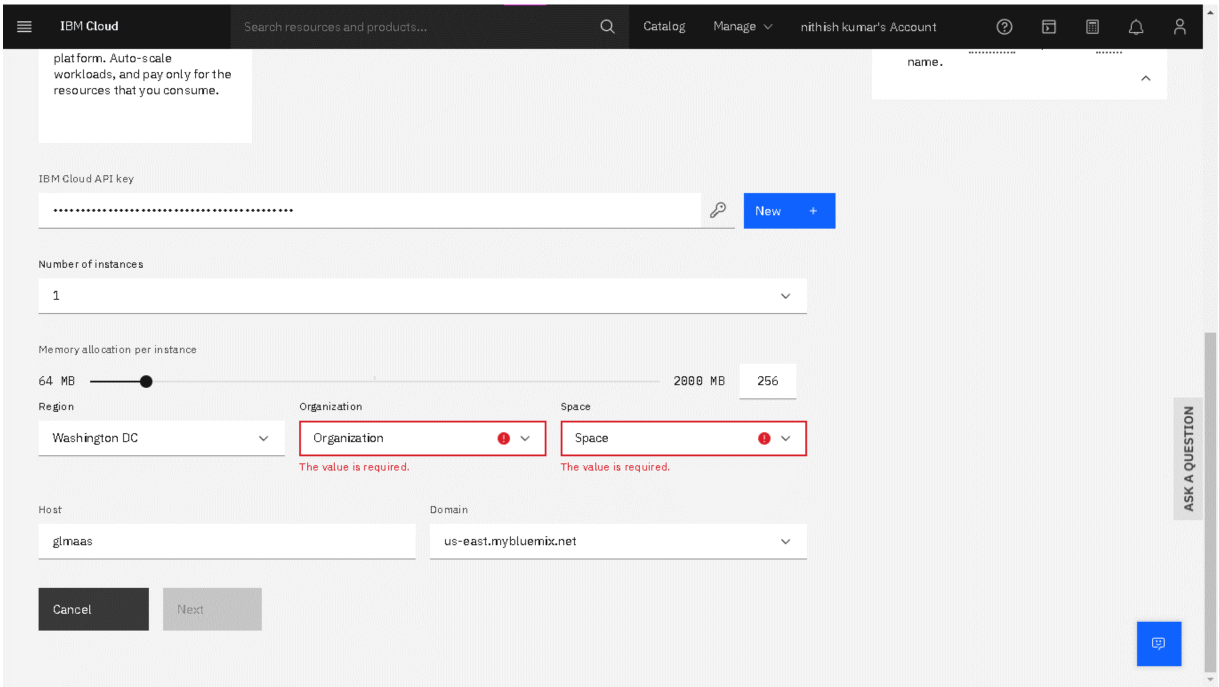
Create new +

Deployment type: Helm

Cancel Next

1. Create an IBM Cloud API key, or select an existing one from a secrets store.
2. Select the container registry region.
3. Enter the container registry namespace if it is not already completed.
4. Select the region where your Kubernetes cluster is located.
5. Select the resource group, cluster namespace, and the cluster name.
6. The deployment type of Helm is selected for you.
7. Click **Next**.

Progress 3: Completing our process



platform. Auto-scale workloads, and pay only for the resources that you consume.

name.

IBM Cloud API key: ••

New +

Number of instances: 1

Memory allocation per instance: 64 MB – 2000 MB: 256

Region: Washington DC

Organization: Organization
The value is required.

Space: Space
The value is required.

Host: glmaas

Domain: us-east.mybluemix.net

Cancel Next

Develop a Python Code

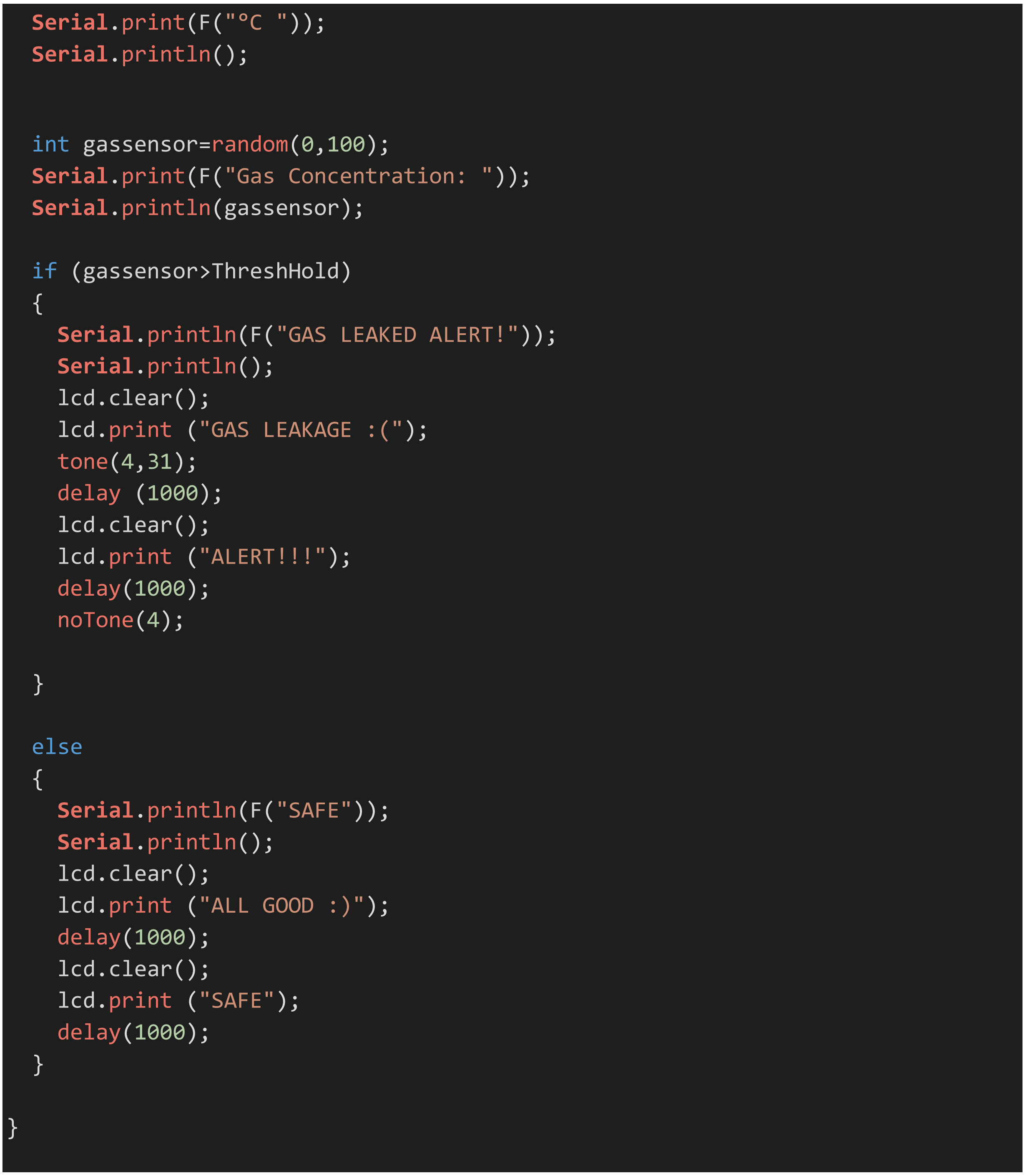
Task assigned: Subashini P

Task started on: 31 - 10 - 2022

Task completion date: 05 - 11 - 2022

sketch.ino

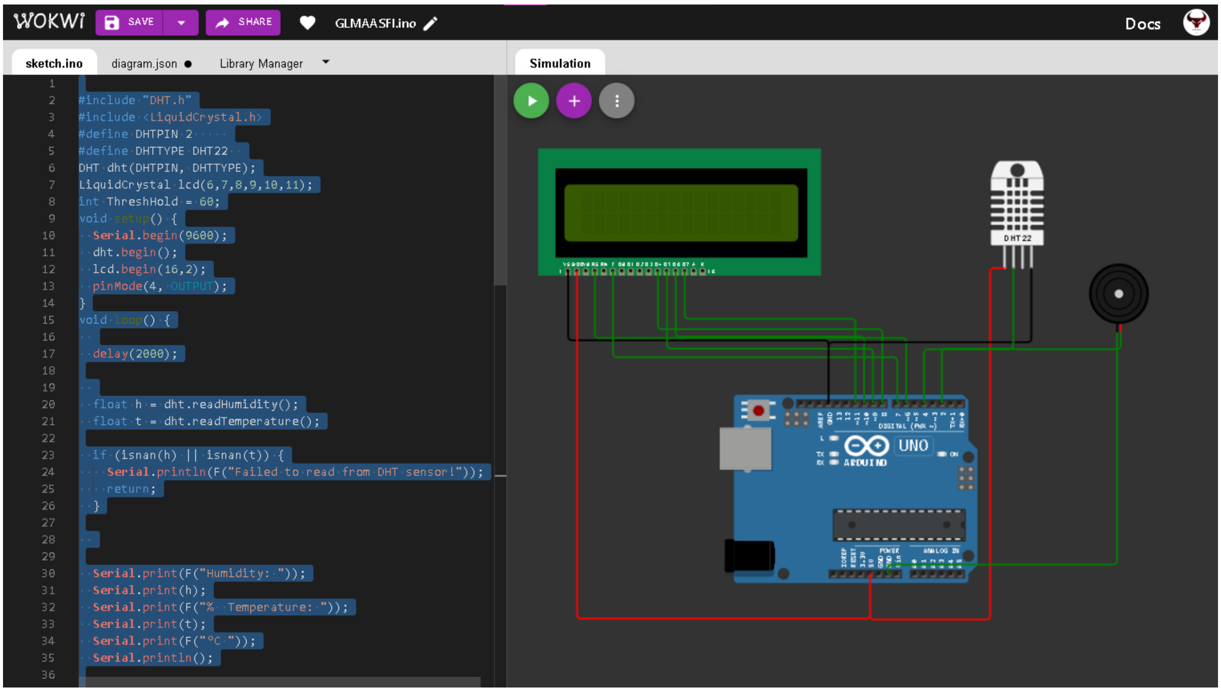
```
#include "DHT.h"
#include <LiquidCrystal.h>
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(6,7,8,9,10,11);
int ThreshHold = 60;
void setup() {
  Serial.begin(9600);
  dht.begin();
  lcd.begin(16,2);
  pinMode(4, OUTPUT);
}
void loop() {

  delay(2000);


  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }



  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
```



```
  Serial.print(F("°C "));
  Serial.println();


  int gassensor=random(0,100);
  Serial.print(F("Gas Concentration: "));
  Serial.println(gassensor);

  if (gassensor>ThreshHold)
  {
    Serial.println(F("GAS LEAKED ALERT!"));
    Serial.println();
    lcd.clear();
    lcd.print ("GAS LEAKAGE :(");
    tone(4,31);
    delay (1000);
    lcd.clear();
    lcd.print ("ALERT!!!");
    delay(1000);
    noTone(4);

  }

  else
  {
    Serial.println(F("SAFE"));
    Serial.println();
    lcd.clear();
    lcd.print ("ALL GOOD :)");
    delay(1000);
    lcd.clear();
    lcd.print ("SAFE");
    delay(1000);
  }

}
```

diagram .json

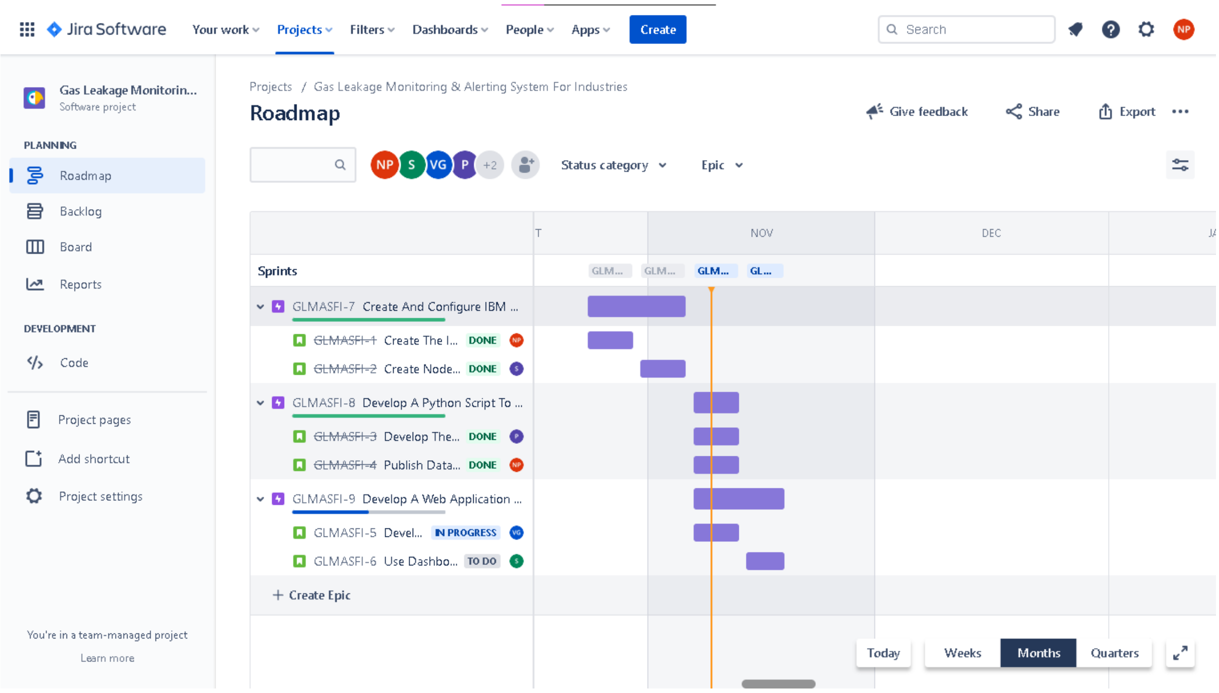
```json
{
  "version": 1,
  "author": "Nithish Kumar",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-arduino-uno", "id": "uno", "top": 96.03, "left": 207.51, "attrs": {} },
    { "type": "wokwi-lcd1602", "id": "lcd1", "top": -129.23, "left": -2.16, "attrs": {} },
    {
      "type": "wokwi-dht22",
      "id": "dht1",
      "top": -116.3,
      "left": 480.78,
      "attrs": { "humidity": "55", "temperature": "25" }
    },
    {
      "type": "wokwi-buzzer",
      "id": "bz1",
      "top": -13.9,
      "left": 586.11,
      "attrs": { "volume": "0.1" }
    }
  ],
  "connections": [
    [ "uno:GND.1", "lcd1:VSS", "black", [ "v-66.86", "h-290.34" ] ],
    [ "uno:5V", "lcd1:VDD", "red", [ "v48.24", "h-328.83" ] ],
    [ "lcd1:RS", "uno:6", "green", [ "v71.5", "h347.08" ] ],
    [ "uno:7", "lcd1:E", "green", [ "v-48.82", "h-327.77" ] ],
    [ "uno:8", "lcd1:D4", "green", [ "v-78.88", "h-255.66" ] ],
    [ "uno:9", "lcd1:D5", "green", [ "v-57.84", "h-232.63" ] ],
    [ "uno:10", "lcd1:D6", "green", [ "v-70.87", "h-214.11" ] ],
    [ "uno:11", "lcd1:D7", "green", [ "v-89.9", "h-198.6" ] ],
    [ "uno:5V", "dht1:VCC", "red", [ "v49.25", "h128.09", "v-67.14" ] ],
    [ "uno:GND.1", "dht1:GND", "black", [ "v-64.85", "h21.28" ] ],
    [ "uno:2", "dht1:SDA", "green", [ "v-57.84", "h57.6" ] ],
    [ "bz1:2", "uno:4", "green", [ "v17.17", "h-195.65" ] ],
    [ "bz1:1", "uno:GND.3", "green", [ "v247.63", "h-225.72" ] ]
  ]
}
```



```

#include "DHT.h"
#include <LiquidCrystal.h>
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(6,7,8,9,10,11);
int ThreshHold = 60;
void setup() {
  Serial.begin(9600);
  dht.begin();
  lcd.begin(16,2);
  pinMode(4, OUTPUT);
}
void loop() {

  delay(2000);



  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }



  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.println();

```

JIRA

Road map



Projects / Gas Leakage Monitoring & Alerting System For Industries

Roadmap

Sprints

GLMASFI-7 Create And Configure IBM ...

~~GLMASFI-1~~ Create The I... DONE NP

~~GLMASFI-2~~ Create Node... DONE S

GLMASFI-8 Develop A Python Script To ...

~~GLMASFI-3~~ Develop The... DONE P

~~GLMASFI-4~~ Publish Data... DONE NP

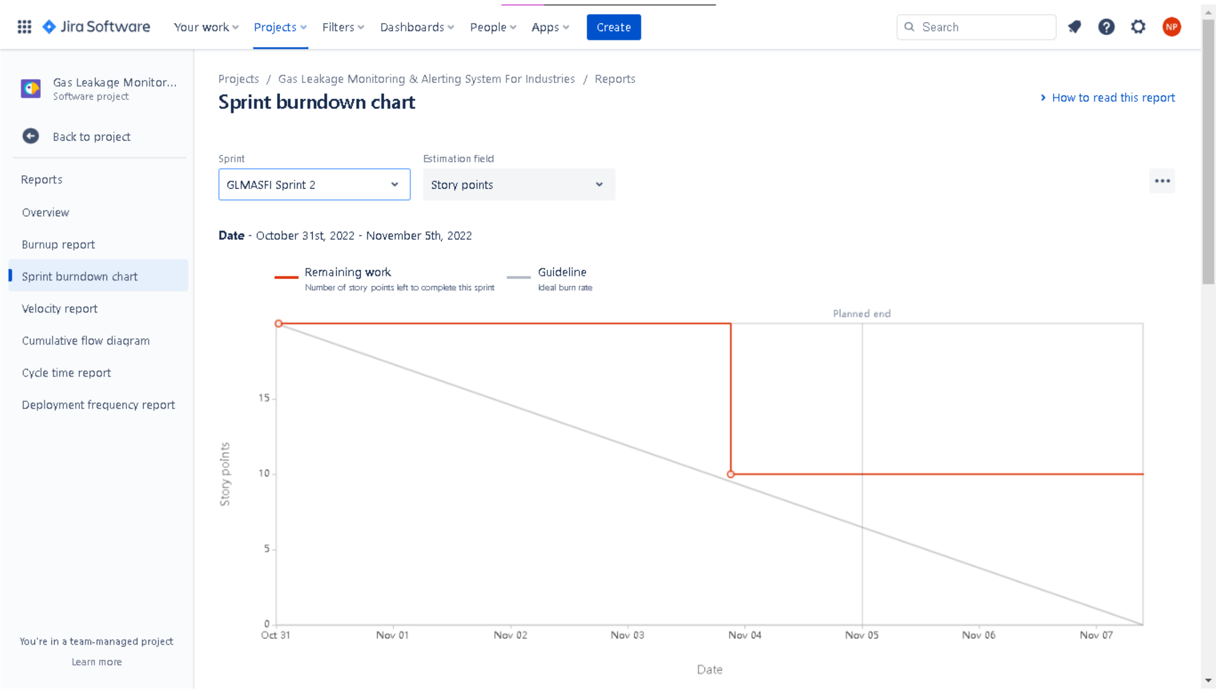
GLMASFI-9 Develop A Web Application ...

GLMASFI-5 Devel... IN PROGRESS VG

GLMASFI-6 Use Dashbo... TO DO S

+ Create Epic

Sprint Burndown Chart



Projects / Gas Leakage Monitoring & Alerting System For Industries / Reports

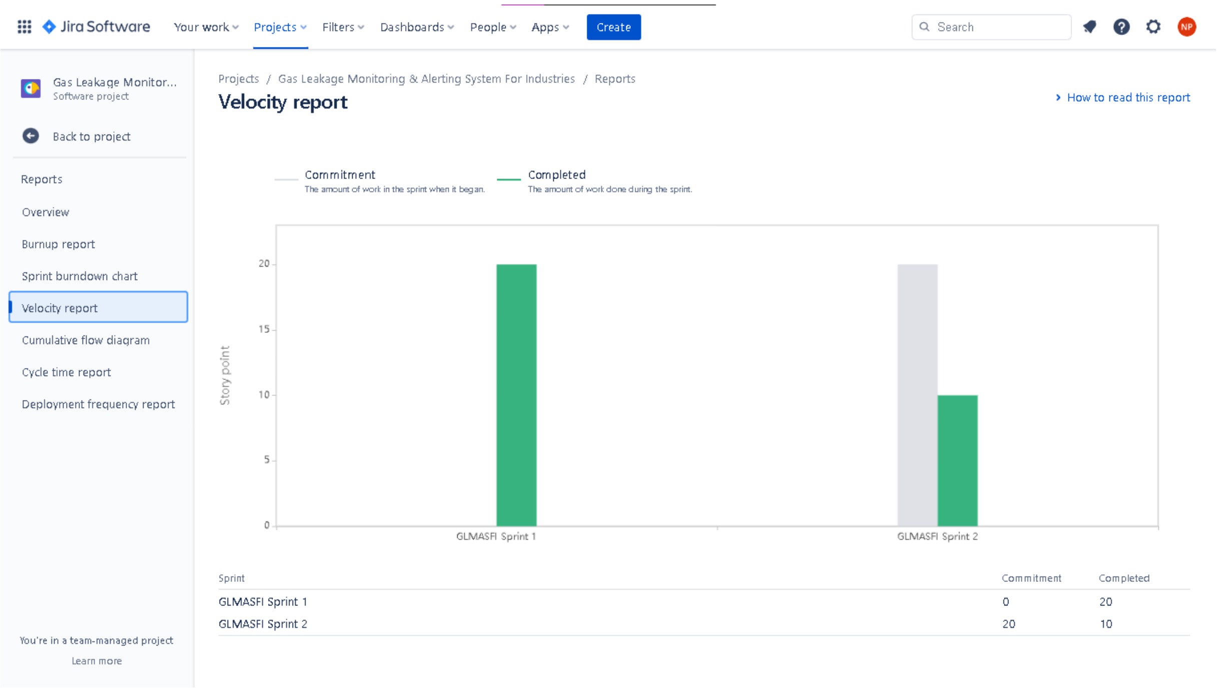
Sprint burndown chart

Sprint: GLMASFI Sprint 2

Estimation field: Story points

Date - October 31st, 2022 - November 5th, 2022

Velocity Report

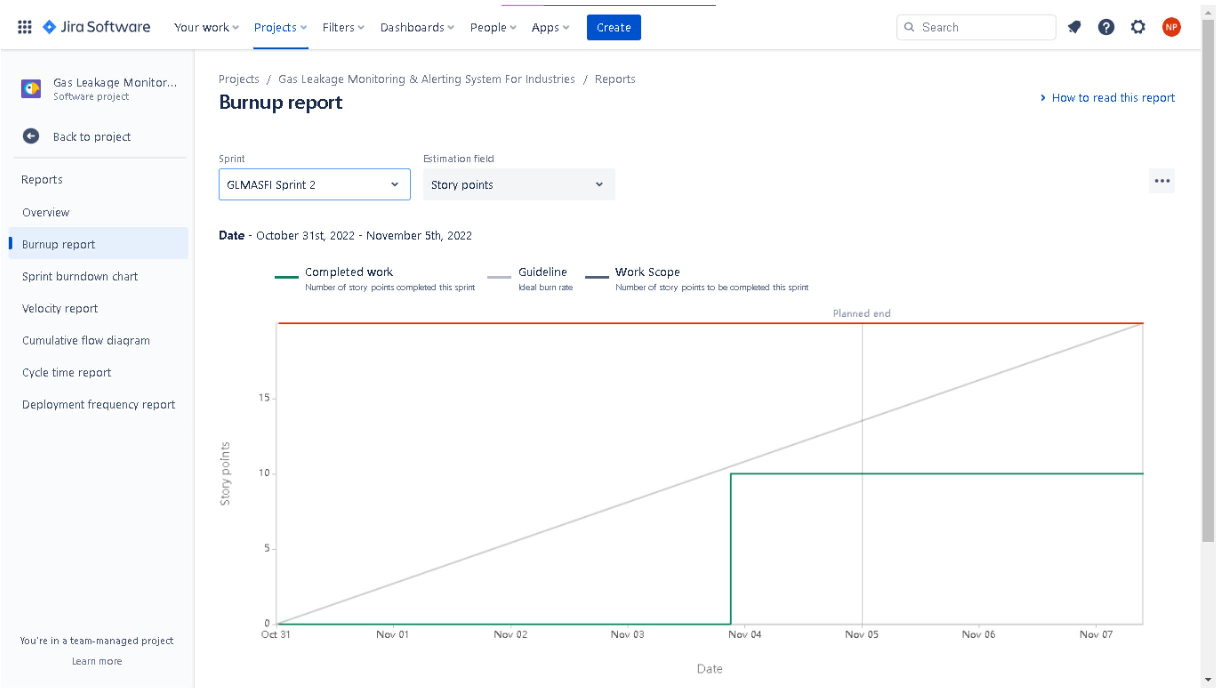


Projects / Gas Leakage Monitoring & Alerting System For Industries / Reports

Velocity report

Sprint	Commitment	Completed
GLMASFI Sprint 1	0	20
GLMASFI Sprint 2	20	10

Burnup report



Projects / Gas Leakage Monitoring & Alerting System For Industries / Reports

Burnup report

Sprint: GLMASFI Sprint 2

Estimation field: Story points

Date - October 31st, 2022 - November 5th, 2022