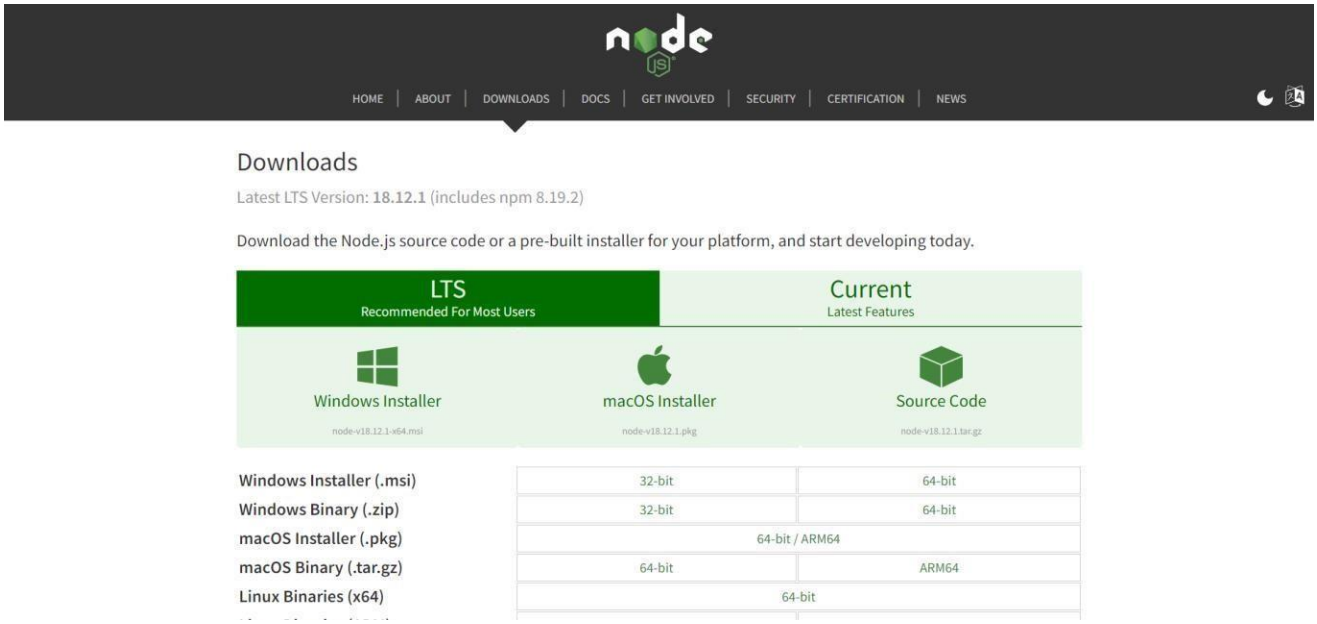
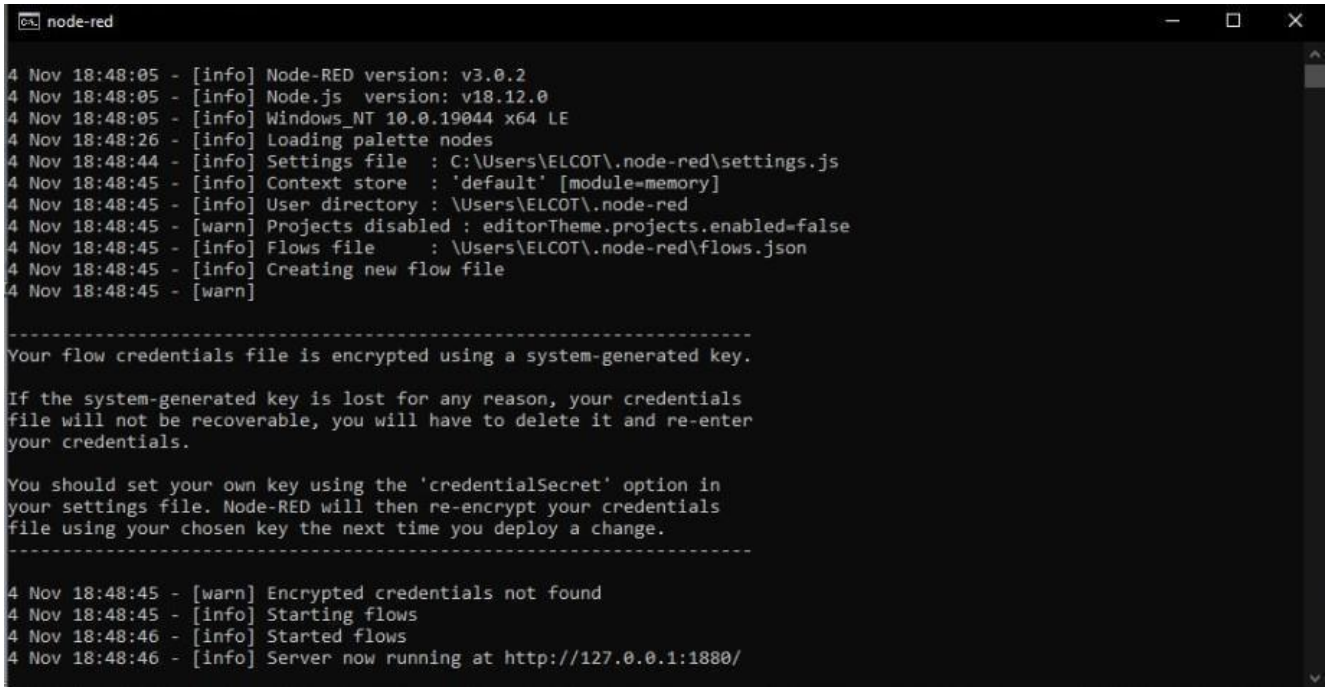
SPRINT 2

TEAM ID	PNT2022TMID04076
Project Name	IoT Based smart crop Protection system for agriculture
Maximum mark	20 marks

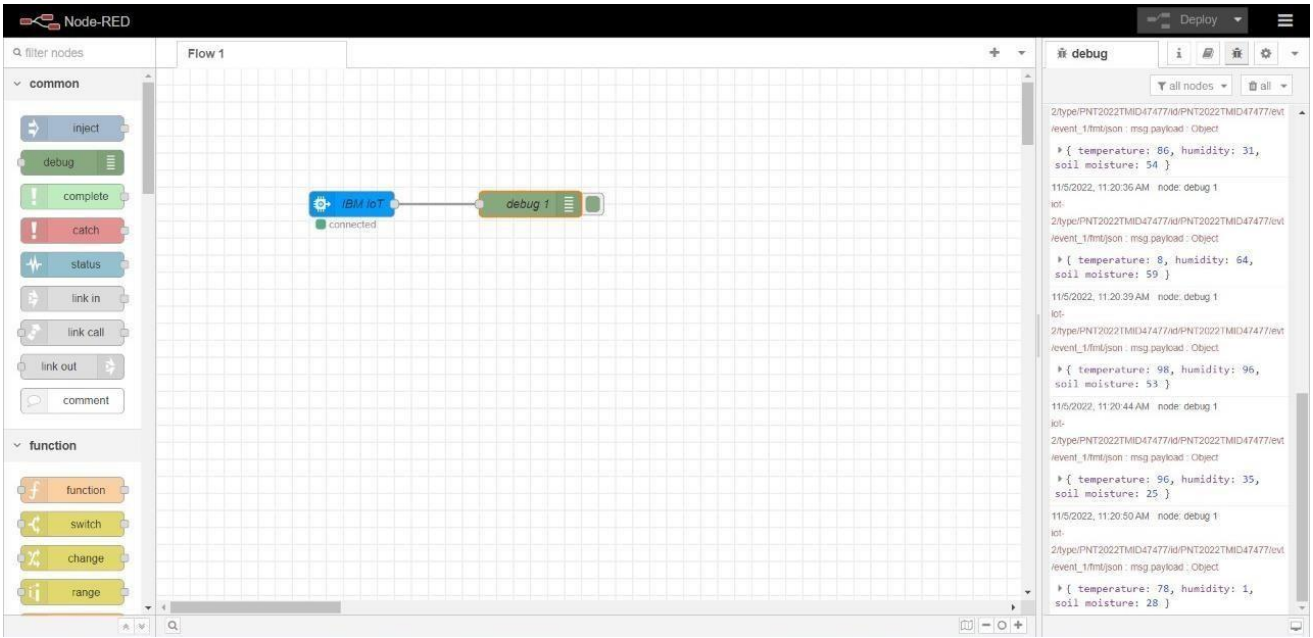
STEP1: Download and Install NODE JS.



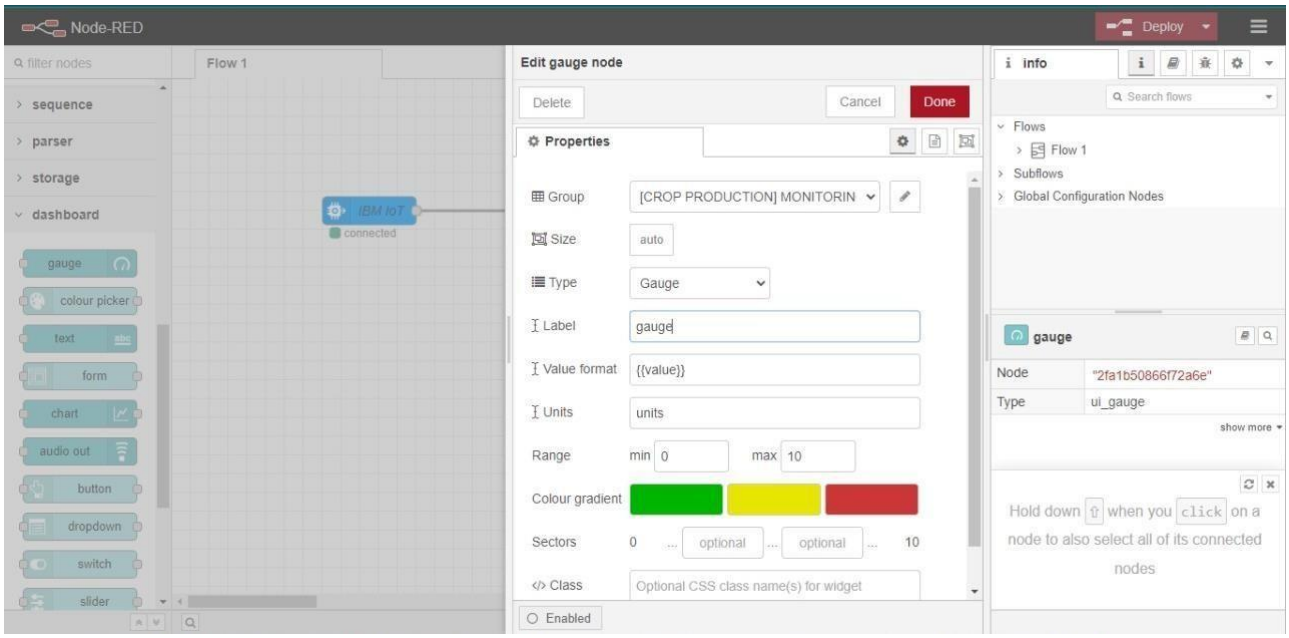
STEP2: Setup node.js and configure command prompt for error check.open node-red from the generated link.

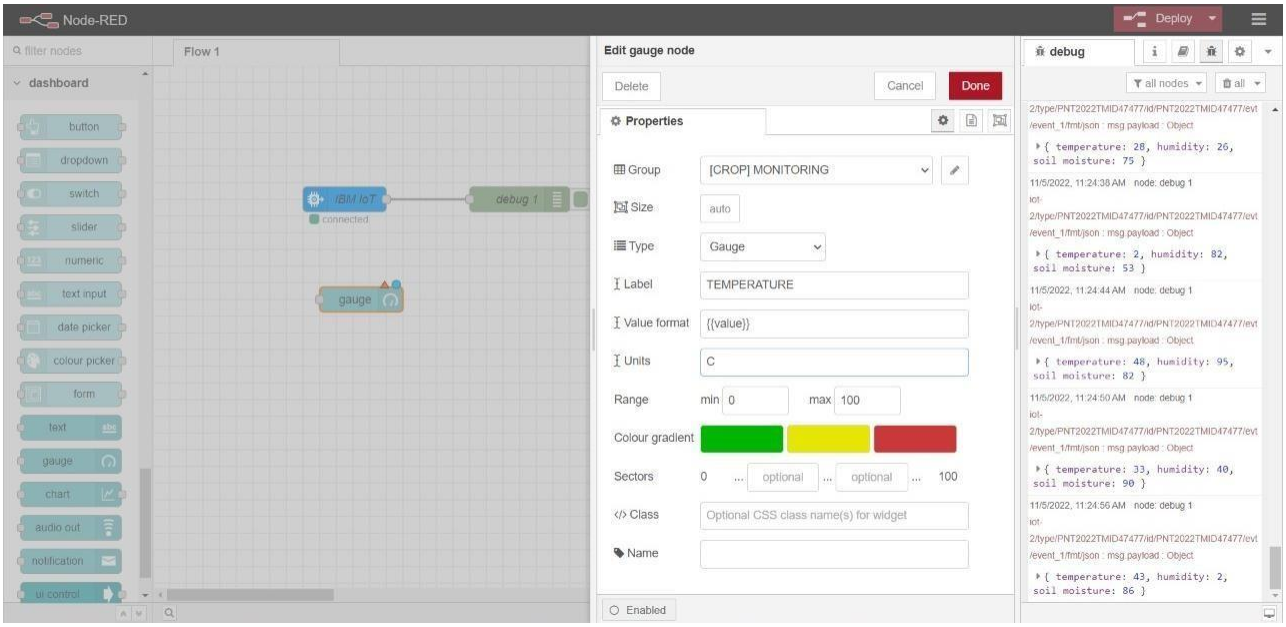


STEP3: Connect IBM IOT in and Debug 1 and Deploy.



STEP4: Edit gauge node (Here the gauge nodes are named as Temperature, Humidity and Soil moisture).





STEP 5: **PYTHON CODE:**

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
import cv2
import numpy as np
import wiot.sdk.device
import playsound
import random
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError

#CloudantDB
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
stub = service_pb2_grpc.V2Stub(clarifaiChannel.get.grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resource_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

#This is how you authenticate
metadata = (('authorization', 'key 0620e202302b4508b90eab7efe7475e4'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "g5d4qO8EIgv4TWUCJj4hfEzgalqEjrDbE82AJDWlAOHo"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN =
"crn:v1:bluemix:public:cloud-object-storage:global:a/c2fa2836eaf3434bbc8b5b58fefff3f0:62e450fd-4c82-4153-ba4
1-ccb53adb8111::"
clientdb = cloudant("apikey-W2njldnwtjO16V53LAVUCqPwc2aHTLmlj1xXvtdGKJBn",
"88cc5f47c1a28afbfb8ad16161583f5a",
url="https://d6c89f97-cf91-48b7-b14b-c99b2fe27c2f-bluemix.cloudantnosqldb.appdomain.cloud")
clientdb.connect()

#Create resource
cos = ibm_boto3.resource("s3",
            ibm_api_key_id=COS_API_KEY_ID,
            ibm_service_instance_id=COS_RESOURCE_CRN,
            ibm_auth_endpoint=COS_AUTH_ENDPOINT,
            config=Config(signature_version="oauth"),
            endpoint_url=COS_ENDPOINT
            )
def = multi_part_upload(bucket_name, item_name, file_path):
  try:
    print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
    #set 5 MB chunks
    part_size = 1024 * 1024 * 5
    #set threadhold to 15 MB
    file_threshold = 1024 * 1024 * 15
    #set the transfer threshold and chunk size
    transfer_config = ibm_boto3.s3.transfer.TransferConfig(
      multipart_threshold=file_threshold,
      multipart_chunksize=part_size
      )
    #the upload_fileobj method will automatically execute a multi-part upload
    #in 5 MB chunks size
    with open(file_path, "rb") as file_data:
      cos.Object(bucket_name, item_name).upload_fileobj(
        Fileobj=file_data,
        Config=transfer_config
        )
    print("Transfer for {0} Complete!\n".format(item_name))
  except ClientError as be:
    print("CLIENT ERROR: {0}\n".format(be))
```

STEP 6: **OUTPUT**