

DEVELOP A WEB APPLICATION USING NODE – RED SERVICE

Team ID:PNT2022TMID45173

Create Node – Red Service:

The screenshot displays the IBM Cloud Developer console interface. The browser address bar shows the URL: `cloud.ibm.com/developer/appservice/apps/c68f9905-b8d7-42b9-ab9d-212280f3ff8c`. The page title is "Node RED UKKWK 2022-11-14".

Details

App URL	http://169.51.204.124:30479
Source	https://us-south.git.cloud.ibm.com/angalaparameshwaria/Nod...
Resource group	Default
Deployment target	Kube/Helm
Created	11/14/2022

Services

Cloudant

- [Open dashboard](#)
- [Documentation](#)
- [API reference](#)

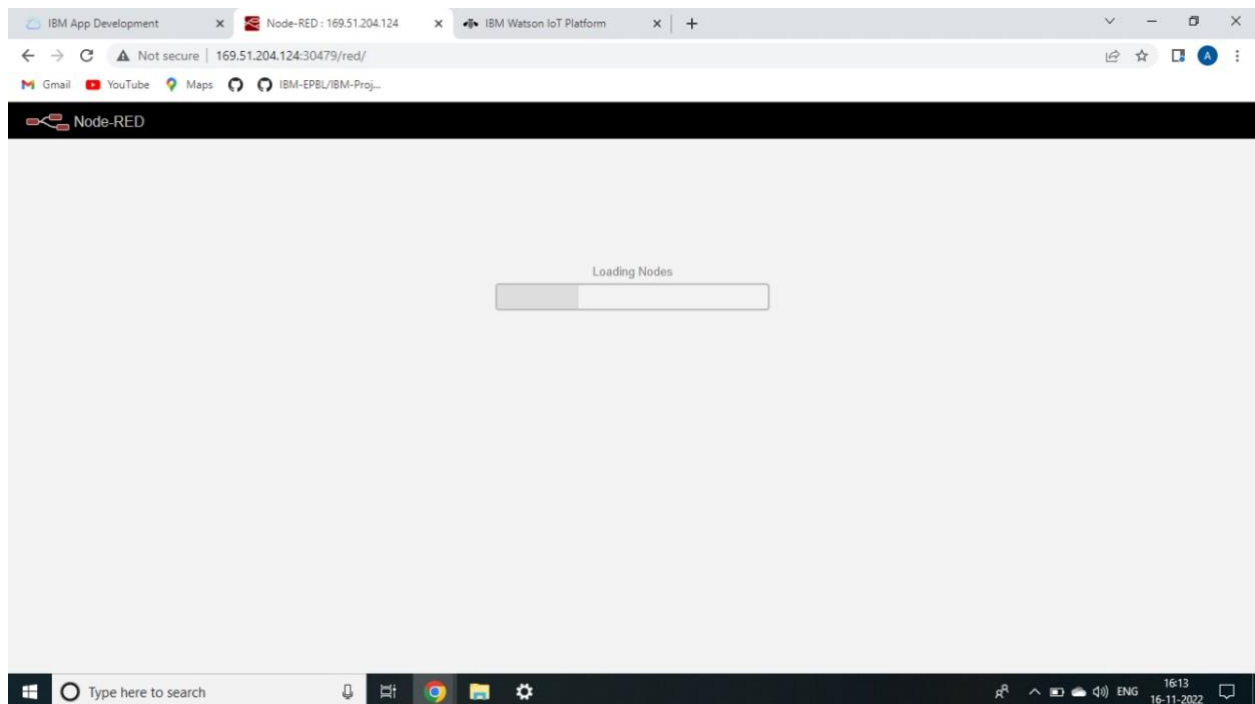
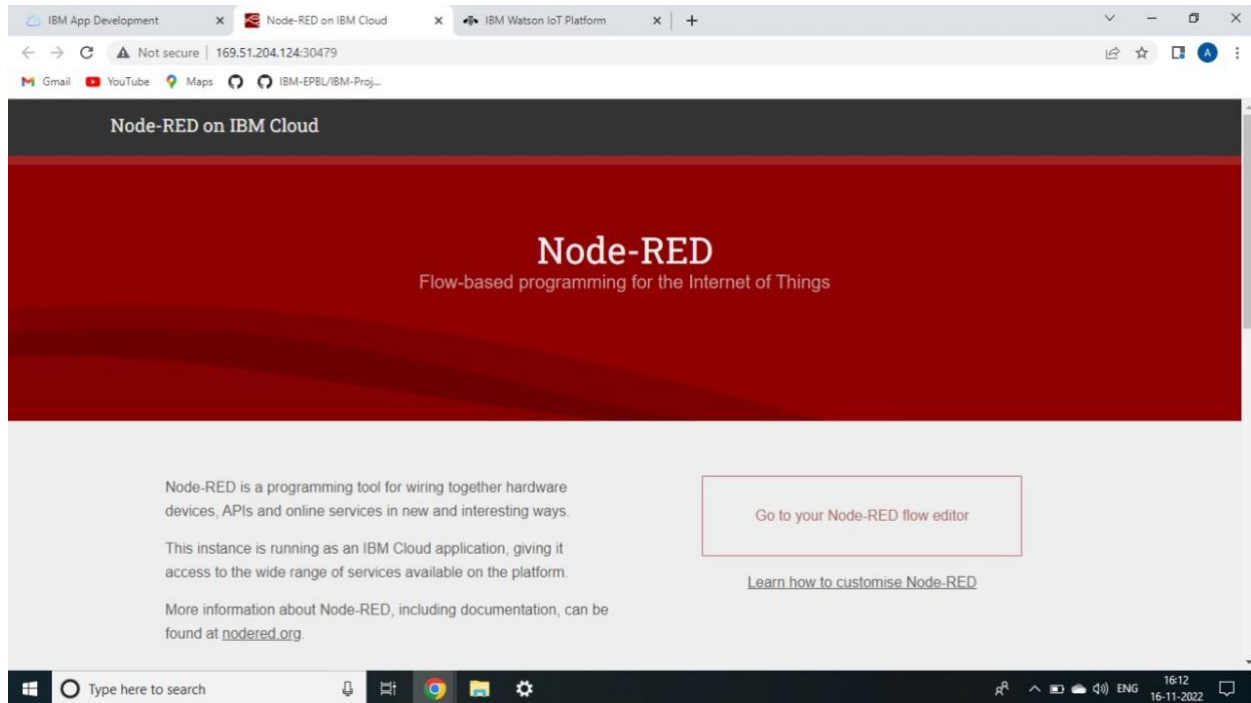
Deployment Automation

Name	NodeREDUKKWK2022-11-14
Location	Dallas
Tool Integrations	

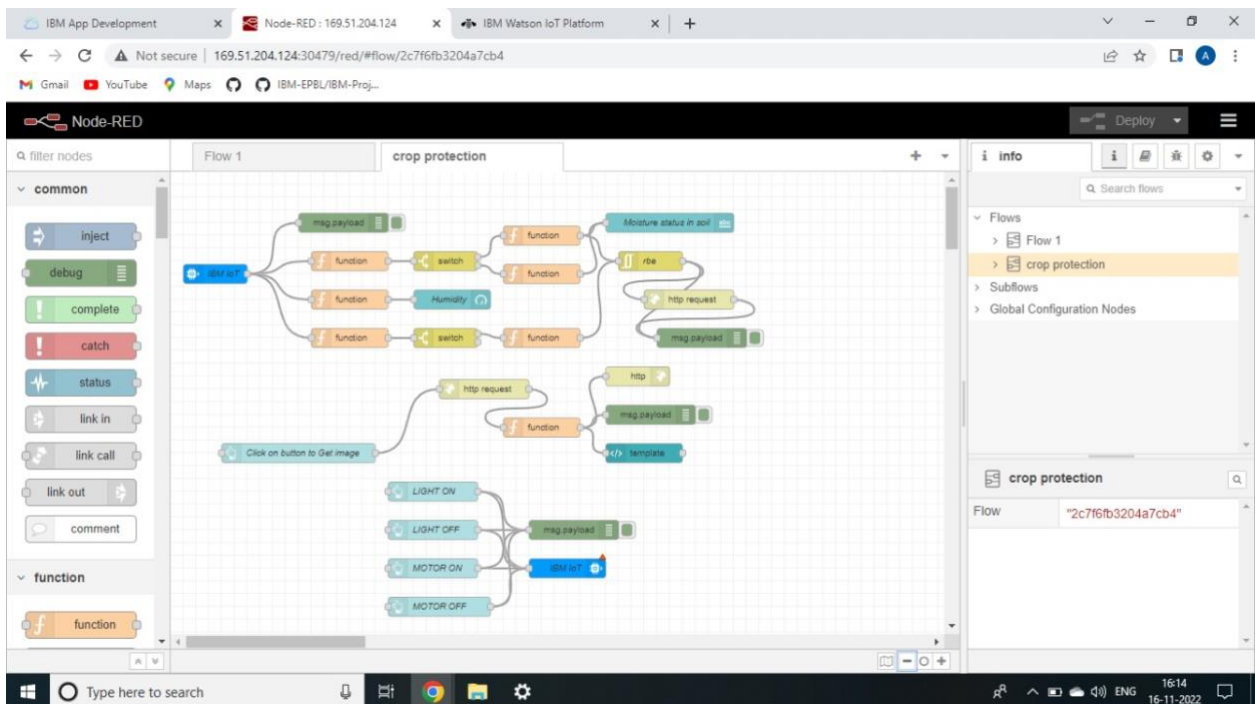
Delivery Pipelines

Name	pr-pipeline
Status	No stages detected
Name	ci-pipeline
Status	Success

The interface includes a top navigation bar with "IBM Cloud", "Catalog", "Manage", and a user profile. A right sidebar contains an "ASK A QUESTION" button. The Windows taskbar at the bottom shows the search bar and system tray.



Develop A Web Application Using Node – RED Service:



```
python ibm.1.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\python ibm.1.py (3.7.0)
File Edit Format Run Options Window Help

import cv2
import numpy as np
import wiotp.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, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
#This is how you authenticate.
metadata= (('authorization', 'key bc885e5165d74ef48f42f6f6a2c9eb87'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list available at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID = "sep-ctim079907XFAE7170cmFLUOQA25" # eg "MOOYSELW4JIT MB-B-2y8TLyBIOQRanco--P3byk"
COS_AUTH_ENDPOINT="https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN="crn:vl:bluemix:public:cloud-object-storage:global:a/6b644a3fda97448c23eeef263ed6:199able5-0d9d-420f-8e4a-98d8604368::" # eg crn:vl:bluemix:public:clo
clientdb = Cloudant({"apikey-v2-16u3crmdpkghhxfdikvpssoh5fwezrmuup5fv5g3ubz", "b0ab119f5d3e6255eabb978e7e2f0e1", url="https://apikey-v2-16u3crmdpkghhxfdikvpssoh5fwezr

#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="auth"),
    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 chunk_size to 10 MB
```

```
"python ibm.1.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\python ibm.1.py (3.7.0)"
File Edit Format Run Options Window Help

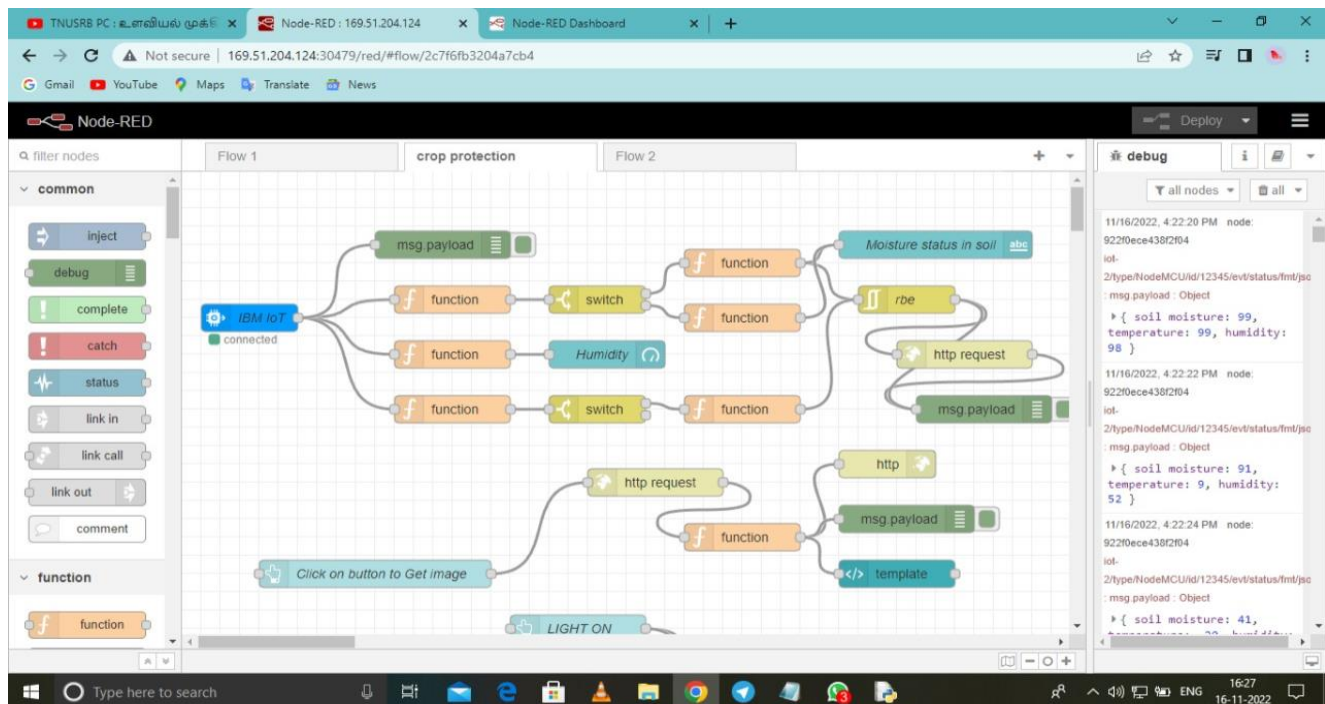
#set 5 MB chunks
part_size 1024 1024 5
#set threshold to 15 MB
file_threshold 1024 1024 15
#set the transfer threshold and chunk size
transfer_config= ibm boto3.s3.transfer.TransferConfig(
    multipart_threshold=file_threshold,
    multipart_chunksize=part_size
)
#the upload fileobj method will automatically execute a multi-part upload
#in 5 chunks for all files over 15 MB
with open(file_path, "b") as file_data:
    cos.Object(bucket_name, item_name).upload_fileobj(
        fileobj=file_data,
        config=transfer_config
    )
print("Transfer for {} Complete!\n".format(item_name))
except clientError as be:
    print("CLIENT ERROR: {}{}\n".format(be))
except Exception as e:
    print("Unable to complete multi-part upload: {}{}\n".format(e))
def myCommandCallback(cmd):
    print ("Command received: %s" % cmd.data)
    command=cmd.data['command']
    print (command)
    if (command == 'lighton'):
        print ('lighton')
    elif (command == 'lightoff'):
        print ('lightoff')
    elif (command == 'motoron'):
        print ('motoron')
    elif (command == 'motoroff'):
        print ('motoroff')
myConfig={
    "identity": {
        "orgId": "xvp7mx",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
Ln: 74 Col: 22
```

```
python ibm final.py - C:\Users\ELCOT\Documents\python ibm final.py (3.7.0)
File Edit Format Run Options Window Help

import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
    "identity": {
        "orgId": "xvp7mx",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect ()

def myCommandCallback (cmd) :
    print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if (m=="motoron"):
        print ("Motor is switched on")
    elif (m=="motoroff"):
        print ("Motor is switched OFF")
    print (" ")
while True:
    soil=random.randint (0,100)
    temp=random.randint (-20, 125)
    hum=random.randint (0, 100)
    myData={'soil moisture': soil, 'temperature':temp, 'humidity':hum}
    client.publishEvent (eventId="status", msgFormat="json",
        data=myData, qos=0 , onPublish=None)
    print ("Published data Successfully: %s", myData)
    time.sleep (2)
    client.commandCallback = myCommandCallback
client.disconnect ()
Ln: 5 Col: 13
```

```
*IDLE Shell 3.8.8*
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/HP/Desktop/crop/crop_protect.py =====
2021-04-06 12:52:19,640 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:hj5fmy:NodeMCU:12345
'sample' successfully created.
File opened
{'Animal': False, 'moisture': 17, 'humidity': 41}
Publish Ok..
{'Animal': False, 'moisture': 84, 'humidity': 16}
Publish Ok..
{'Animal': False, 'moisture': 48, 'humidity': 43}
Publish Ok..
{'Animal': False, 'moisture': 0, 'humidity': 3}
Publish Ok..
{'Animal': False, 'moisture': 73, 'humidity': 68}
Publish Ok..
{'Animal': False, 'moisture': 26, 'humidity': 26}
Publish Ok..
{'Animal': False, 'moisture': 96, 'humidity': 59}
Publish Ok..
```



The screenshot shows a web browser window displaying a Node-RED dashboard. The dashboard has a blue header with the text 'crop protection'. Below the header, there is a section titled 'Crop protection' in blue text. The main content area is white and contains several elements:

- A humidity gauge on the left, labeled 'Humidity' above it. The gauge is a semi-circle with a green segment on the left and a grey segment on the right. A black needle points to the value '11' on a scale from 0 to 100. Below the gauge, the text 'units' is visible.
- A text label 'Moisture status in soil' on the right side.
- Four blue buttons arranged in a 2x2 grid on the right side:
 - Top-left: 'LIGHT ON'
 - Top-right: 'LIGHT OFF'
 - Bottom-left: 'MOTOR ON'
 - Bottom-right: 'MOTOR OFF'
- A large blue button at the bottom center with the text 'CLICK ON BUTTON TO GET IMAGE'.
- Below the large button, there is a checkbox and the text 'Click on the button to get image'.