SPRINT - 4

TEAM ID: PNT2022TMID12045

TRANSMITTING THE DATABASE FROM THE CLOUDANT TO THE NODE RED WEB APP UI

CODE:

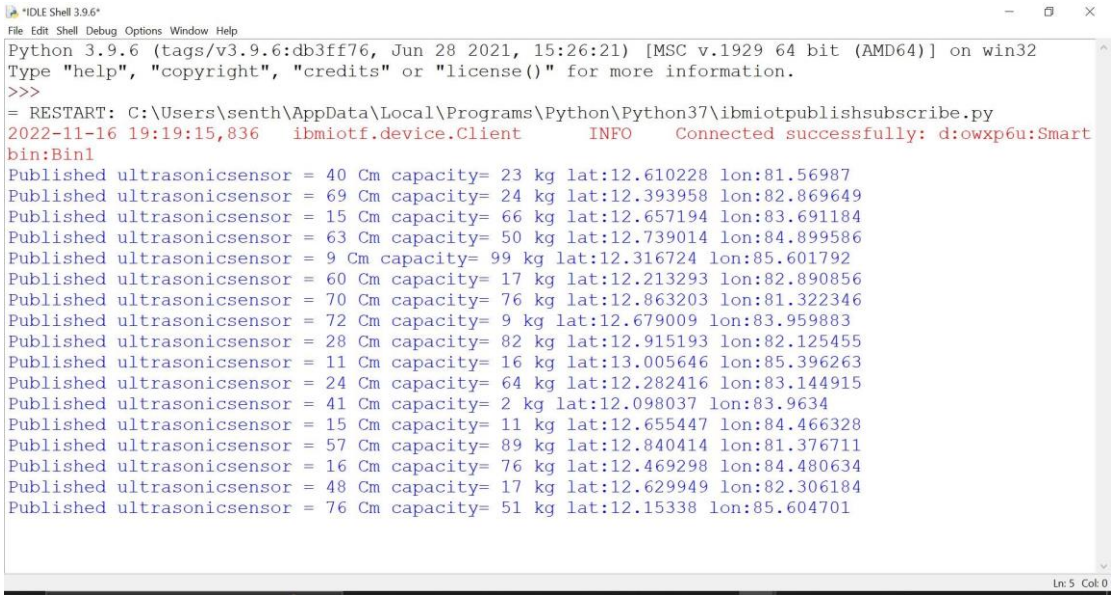
```
import time import
sys import
ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "owxp6u" deviceType =
"Smartbin" deviceId = "Bin1" authMethod =
"token" authToken= "12345678910"
#     Initialize     GPIO     def     myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="lighton":
      print ("led is on")
   else : print ("led is
      off")
try: deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken} deviceCli
   = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
   print("Caught exception connecting device: %s" % str(e))
   sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times
deviceCli.connect()

while True:
      #Get Sensor Data from DHT11
      time.sleep(5)
      ultrasensor=random.randint(0,80)
      capacity=random.randint(0,100)
      lat=round(random.uniform(12.03,13.05),6)
      lon=round(random.uniform(80.80,85.90),6)
      data = { 'ultrasonicsensor' : ultrasensor, 'capacity': capacity,'lat':lat,'lom':lon}
      #print data def
      myOnPublishCallback():
         print ("Published ultrasonicsensor = %s Cm" % ultrasensor, "capacity= %s  kg" %
capacity,"lat:%s"%lat,"lon:%s"%lon)

     success =  deviceCli.publishEvent("IoTSensor",  "json", data,  qos=0,
on_publish=myOnPublishCallback) if not success:
         print("Not connected to IoTF")
```

time.sleep(1) deviceCli.commandCallback =

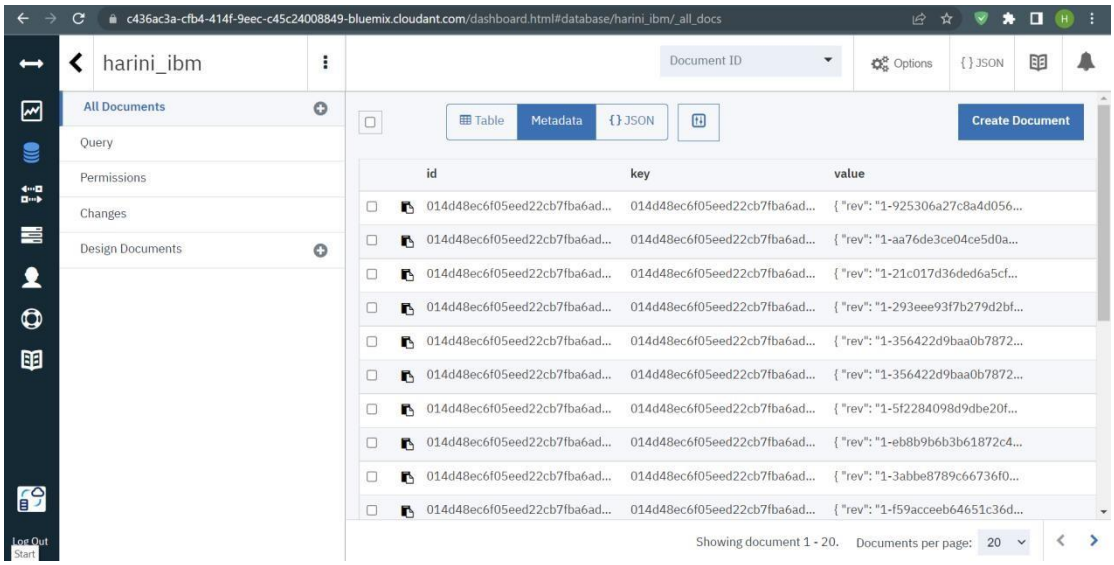
myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

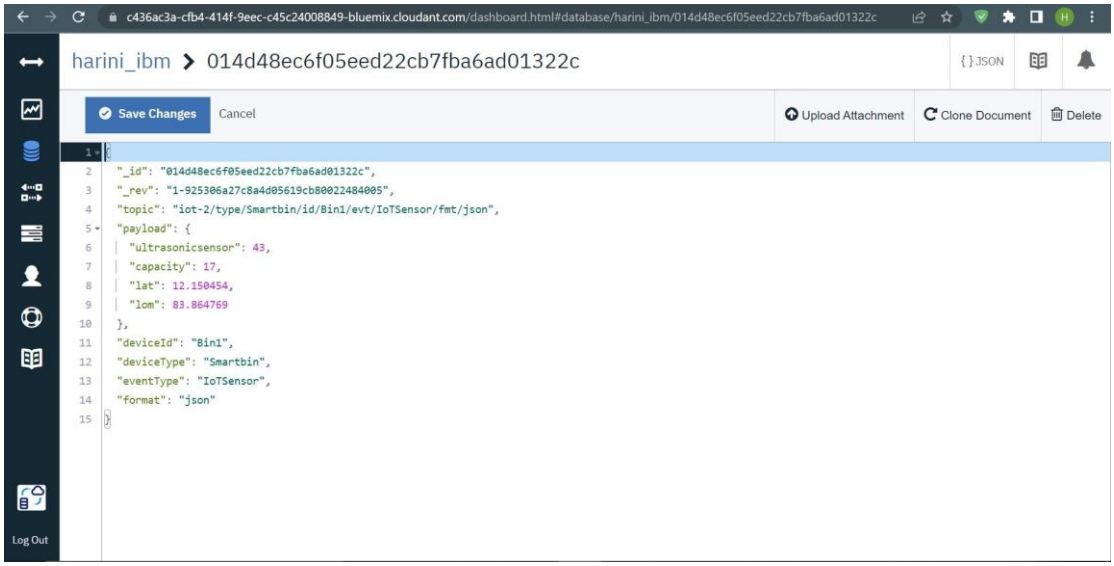
PYTHON OUTPUT:



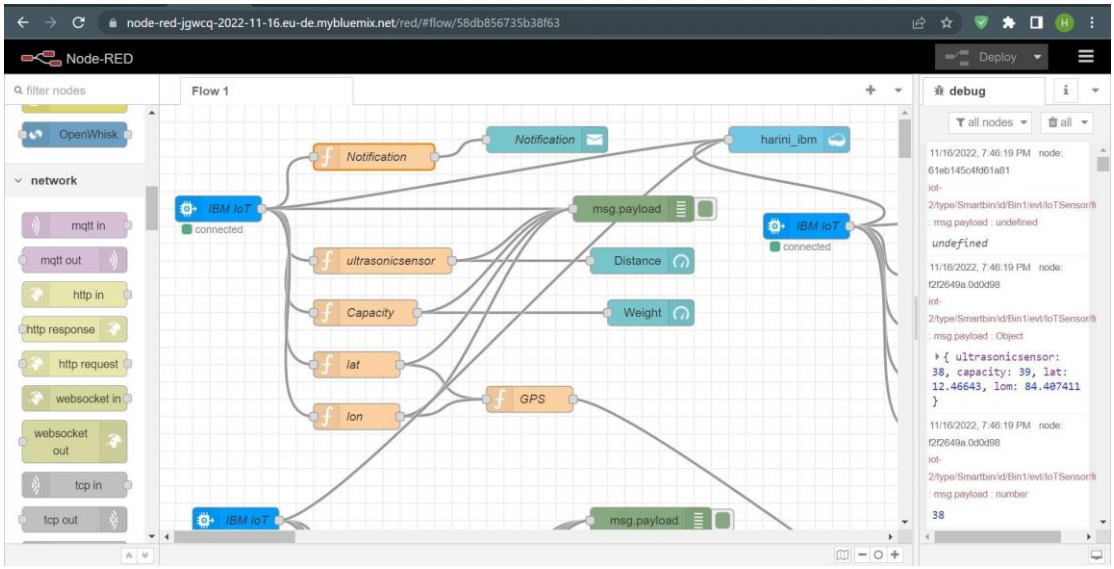
CLOUDANT METADATA:



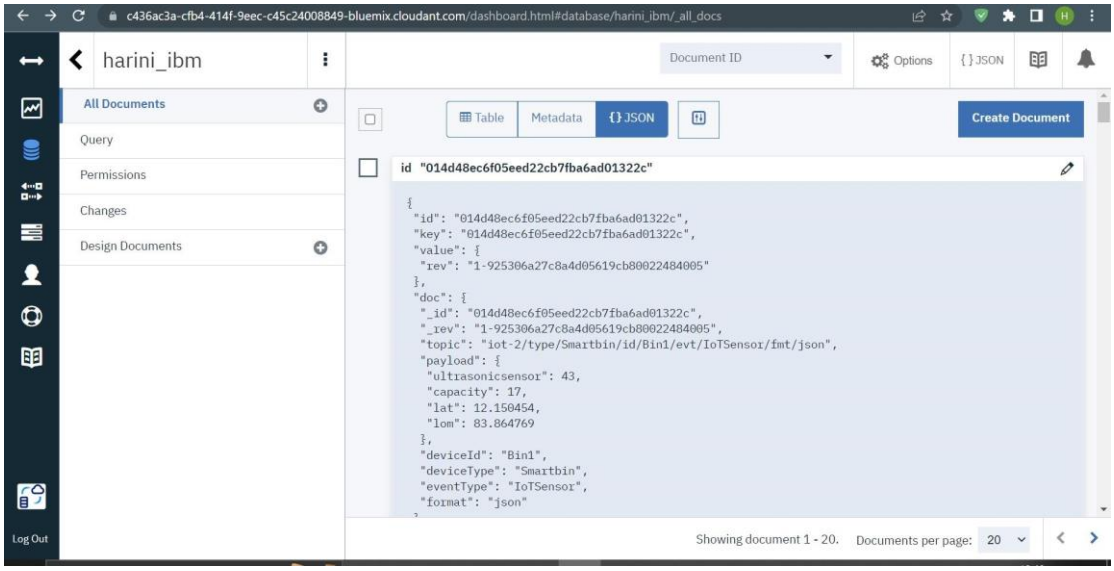
CLOUDANT METADATA INFO:



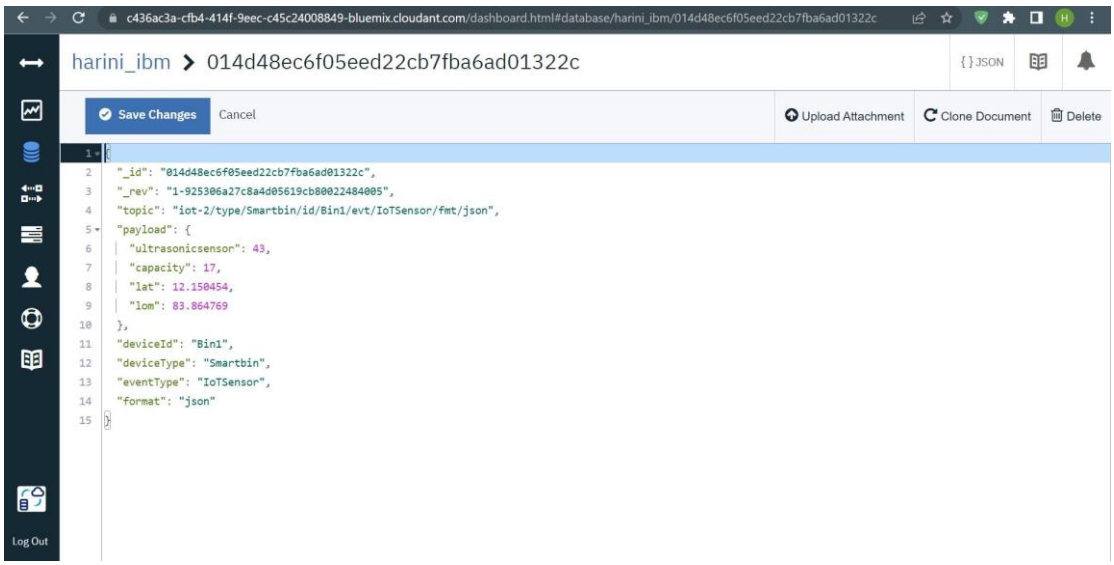
CLOUDANT DIAGRAM:



JSON CODE:



CLOUDANT DOCUMENT:



RESULT:

The node red web app ui was used to successfully create the cloudant database.The cloudant was able to correctly store the data.