FINAL DELIVERABLES

Date	11 November 2022
Team ID	PNT2022TMID05356
Project Name	Project - Smart Waste Management system foe Metropolitan cities

PROGRAM CODE:

```
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
def distance():
# set Trigger to HIGH
GPIO.output(GPIO_TRIGGER, True)
# set Trigger after 0.01ms to LOW
time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER, False)
StartTime = time.time()
StopTime = time.time() # save
StartTime while
GPIO.input(GPIO_ECHO) == 0:
StartTime = time.time()
# save time of arrival
while GPIO.input(GPIO_ECHO) == 1:
StopTime = time.time()
# time difference between start and arrival
TimeElapsed = StopTime – StartTime
# multiply with the sonic speed (34300 cm/s)
# and divide by 2, because there and back
distance = (TimeElapsed * 34300) / 2
return distance
```

23

```
if __name__ == '__main__':
try:
while True:
dist = distance() print ("Measured Distance =
%.1f cm" % dist) percent = (100.0 - (dist *
100/40.0)) url =
"http://localhost:80/demoaddbin.php?bin_id=1&percent_filled="+str(percent)
x= urllib.urlopen(url)
print(x.read)
time.sleep(5)
# Reset by pressing CTRL + C
except KeyboardInterrupt:
print("Measurement stopped by User")
GPIO.cleanup()
```

PROGRAM CODE FOR ACCESS DATABASE:

```
package com.bin; import
android.app.NotificationManager; import
android.app.PendingIntent; import
android.app.Service; import
android.content.Context; import
android.content.Intent; import
android.content.SharedPreferences; import
android.media.RingtoneManager;

import android.net.Uri; import
android.os.AsyncTask; import android.os.Handler;
import android.os.IBinder; import
android.support.annotation.Nullable; import
android.support.v4.app.NotificationCompat; import
android.util.Log; import android.widget.Toast;
import org.json.JSONArray; import
org.json.JSONException; import
org.json.JSONObject; import
java.io.BufferedReader; import
java.io.IOException; import java.io.InputStream;
import java.io.InputStreamReader; import
java.net.HttpURLConnection; import
java.net.URL;
```

```
public class GetData extends Service {
String BASE_URL = "http://dustbin.000webhostapp.com/";
String POPMOVIES_BASE_URL = BASE_URL + "getresponsefrombin.php";
SharedPreferences preferences;
SharedPreferences.Editor editor;
@Override
public void onCreate() {
Toast.makeText(this, "Service Called", Toast.LENGTH_SHORT).show();
Log.d("Create:","called"); //addNotification(); super.onCreate();
```

```
}
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
Log.d("onStartCommand:","called"); preferences =
getSharedPreferences("DustBin",MODE_PRIVATE);
final Handler handler = new Handler();
Runnable runnable = new Runnable() {
@Override
public void run() {
Log.d("handler","run();"); new
DustbinTask().execute();
handler.postDelayed(this,
5000);
} };
//Start
handler.postDelayed(runnable, 1000);
return START_STICKY;
}
@Override
public void onDestroy() {
Log.d("Destroy:","called");
```

```
super.onDestroy();
}
@Nullable
@Override
```

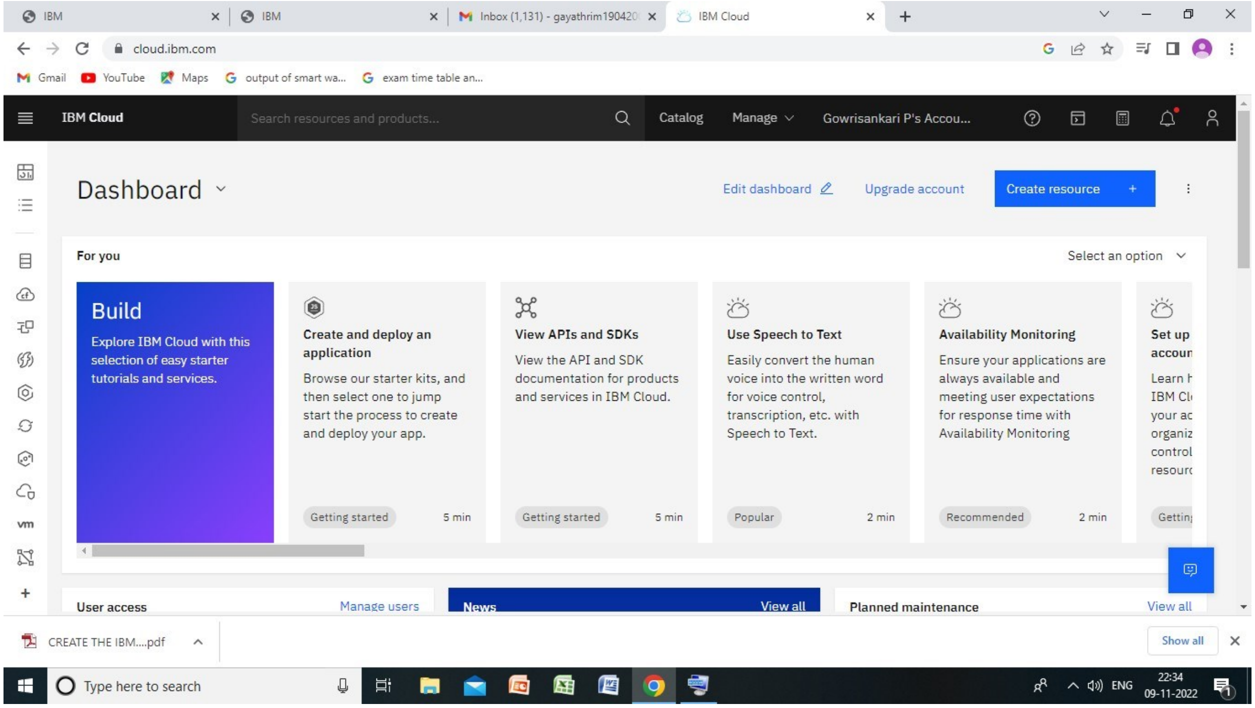
```
public IBinder onBind(Intent intent) {
Log.d("Bind:","called"); return null;
}
public class DustbinTask extends AsyncTask<Void, Void,Void>{
@Override
protected void onPreExecute() {
Log.d("onPreExecute","initiate");
try {
if (!new Network(GetData.this).isConnected()) {
Log.d("onPreExecute","No Internet Available!!");
cancel(true);
}}
catch (InterruptedException | IOException e) {
e.printStackTrace();
}
@Override
protected Void doInBackground(Void... params) {
HttpURLConnection urlConnection = null;
BufferedReader reader = null;
URL url;
String MoviesJsonStr;
try {
url = new URL(POPMOVIES_BASE_URL);
urlConnection = (HttpURLConnection) url.openConnection();
urlConnection.setRequestMethod("GET"); urlConnection.connect();
InputStream inputStream = urlConnection.getInputStream();
StringBuilder buffer = new StringBuilder() reader = new
BufferedReader(new InputStreamReader(inputStream)); String line;
while ((line = reader.readLine()) != null) {
buffer.append(line).append("\n");
}
MoviesJsonStr = buffer.toString();
getMovieNames(MoviesJsonStr); } catch
(IOException | JSONException e1) {
e1.printStackTrace();
} finally {
if (urlConnection != null) {
urlConnection.disconnect();
} if (
reader != null) {
try {
reader.close();
} catch (final IOException ignored) {}
}} return null;
}}
private void getMovieNames(String MovieJsonStr) throws JSONException {
JSONObject MovieJson = new JSONObject(MovieJsonStr);
JSONArray movieLists = MovieJson.getJSONArray("bin_info");
for (int i = 0; i < movieLists.length(); i++) {
```

```
JSONObject jMovieDetails = movieLists.getJSONObject(i);
String name = jMovieDetails.getString("bin_id"); int id =
jMovieDetails.getInt("percent_filled");
Log.d("DATA", name + " " + id);
MainActivity.percent = id;
if(id>=80){
30
addNotification(id);
}} //Log.v("Length: ", String.valueOf(movieLists.length()));
}
//Show a notification private void addNotification(int
id) { int min, max; int percentage =
preferences.getInt("last_percent",0); min =
percentage - 5; max = percentage + 5;
if (min > id || id > max) {
Intent intent = new Intent(this, MainActivity.class
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
editor = preferences.edit();
editor.putInt("last_percent",id);
editor.apply();
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0/*Request code*/,
intent, PendingIntent.FLAG_ONE_SHOT);
//Set sound of notification
Uri notificationSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
NotificationCompat.Builder notifiBuilder = new NotificationCompat.Builder(this)
.setSmallIcon(R.mipmap.ic_launcher
.setContentTitle(id + "% Dustbin Full")
.setContentText("Please clear your trash")
.setAutoCancel(true)
.setSound(notificationSound)
.setContentIntent(pendingIntent);
NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(999 /*ID of notification*/, notifiBuilder.build());
//stopSelf()
```

PROGRAM FOR CONNECTING APPLICATION TO INTERNET:

```
package com.bin; import
android.content.Context; import
android.net.ConnectivityManager; import
android.util.Log; import
java.io.IOException;
/**
* Created by Sylvester on 03-Mar-17.
*/
class Network { private
Context mContext;
Network(Context mContext) {
this.mContext = mContext;
}
private boolean isNetworkAvailable() { final ConnectivityManager
connectivityManager = ((ConnectivityManager)
mContext.getSystemService(Context.CONNECTIVITY_SERVICE));
```

```
return connectivityManager.getActiveNetworkInfo() != null &&
connectivityManager.getActiveNetworkInfo().isConnected();
}
boolean isConnected() throws InterruptedException, IOException
{ if (
isNetworkAvailable()) {
String command = "ping -c 1 google.com";
return (Runtime.getRuntime().exec (command).waitFor() == 0
```



Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Create and deploy an application

Browse our starter kits, and then select one to jump start the process to create and deploy your app.

Getting started 5 min

View APIs and SDKs

View the API and SDK documentation for products and services in IBM Cloud.

Getting started 5 min

Use Speech to Text

Easily convert the human voice into the written word for voice control, transcription, etc. with Speech to Text.

Popular 2 min

Availability Monitoring

Ensure your applications are always available and meeting user expectations for response time with Availability Monitoring

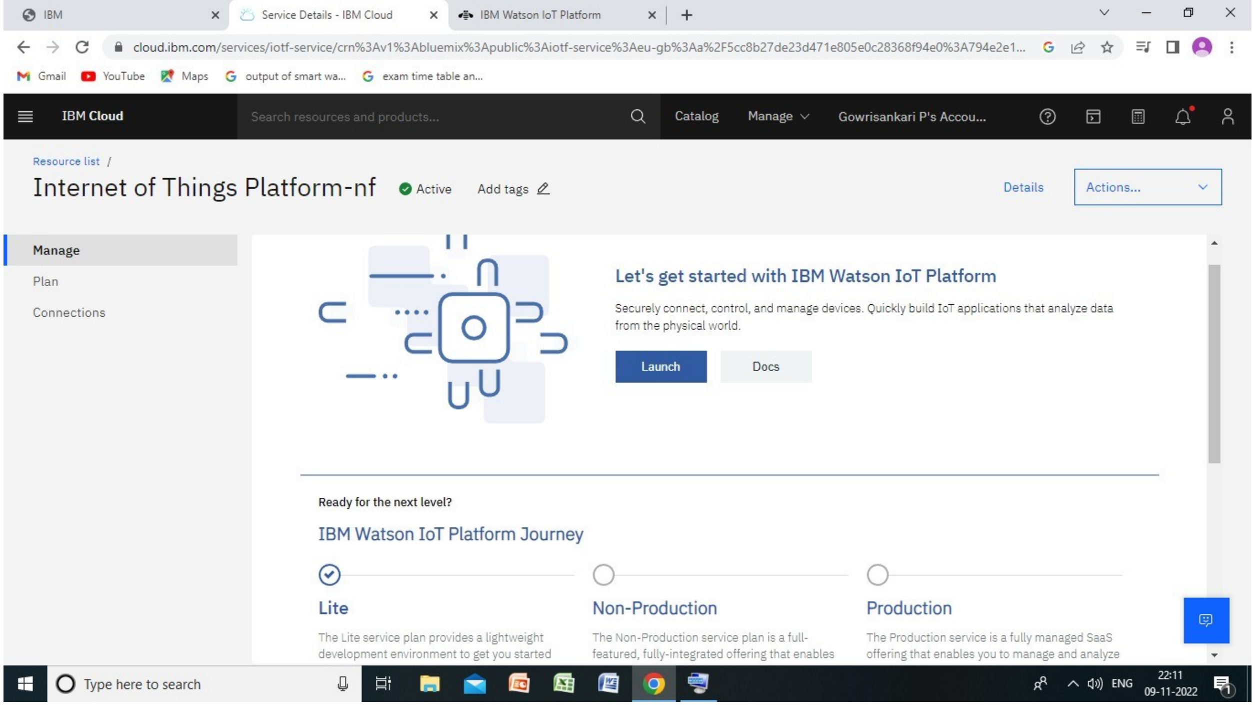
Recommended 2 min

Set up accoun

Learn h IBM Cl your ac organiz control resourc

Getting

CLOUD SERVICE


IBM
Service Details - IBM Cloud
IBM Watson IoT Platform
cloud.ibm.com/services/iotf-service/crn%3Av1%3Abluemix%3Apublic%3Aiotf-service%3Aeu-gb%3Aa%2F5cc8b27de23d471e805e0c28368f94e0%3A794e2e1...
Gmail
YouTube
Maps
output of smart wa...
exam time table an...
IBM Cloud
Search resources and products...
Catalog
Manage
Gowrisankari P's Accou...
Resource list /
Details
Actions...
Manage
Plan
Connections
Launch
Docs
Type here to search
ENG
22:11
09-11-2022

Internet of Things Platform-nf ● Active Add tags

Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Ready for the next level?

IBM Watson IoT Platform Journey

Lite

The Lite service plan provides a lightweight development environment to get you started

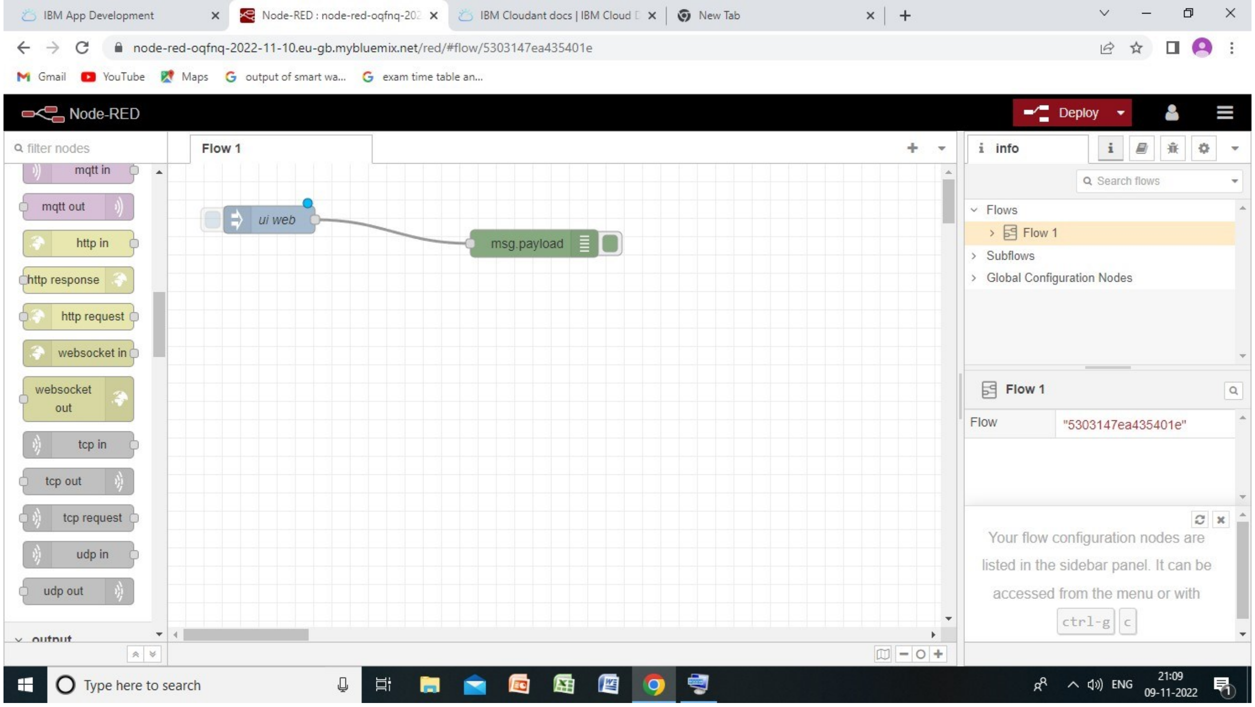
Non-Production

The Non-Production service plan is a full-featured, fully-integrated offering that enables

Production

The Production service is a fully managed SaaS offering that enables you to manage and analyze

IBM WATSON IOT PLATFORM

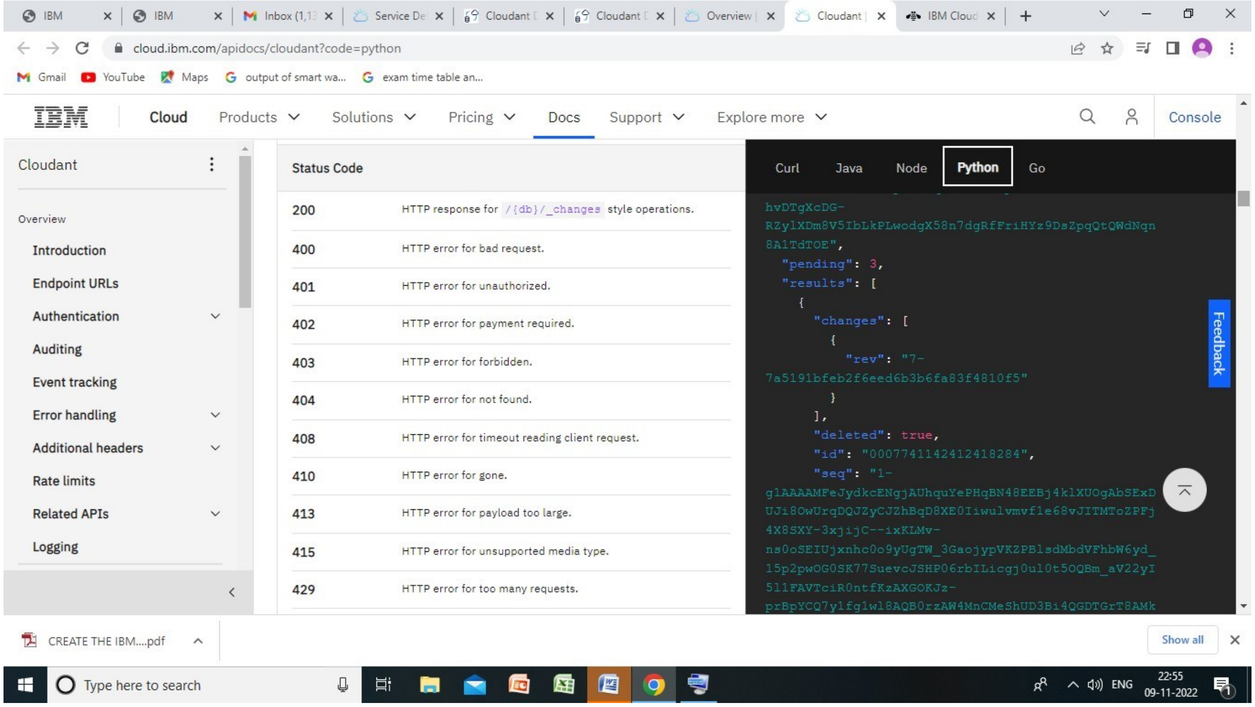


Flow 1

Flow	"5303147ea435401e"

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with ctrl-g c

UI Web using NODE RED

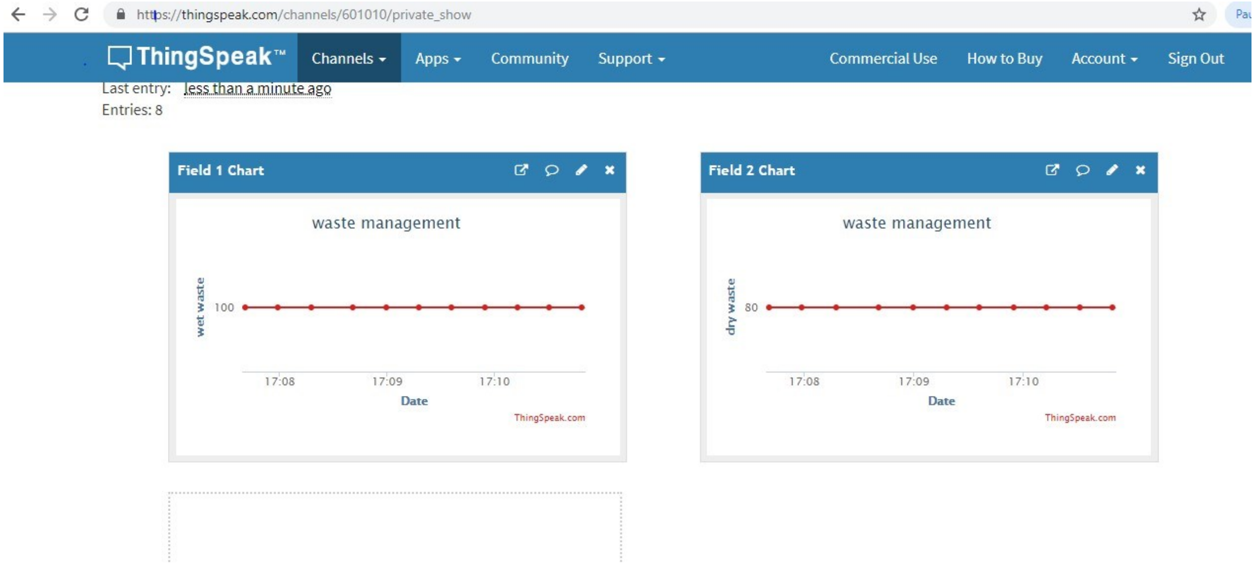


Status Code	
200	HTTP response for `/{db}/_changes` style operations.
400	HTTP error for bad request.
401	HTTP error for unauthorized.
402	HTTP error for payment required.
403	HTTP error for forbidden.
404	HTTP error for not found.
408	HTTP error for timeout reading client request.
410	HTTP error for gone.
413	HTTP error for payload too large.
415	HTTP error for unsupported media type.
429	HTTP error for too many requests.

```
hvDTgXcDG-
RZylXDm8V5IbLkPLwodgX58n7dgRfFriHYz9DsZpqQtQWdNqn
8A1TdTOE",
  "pending": 3,
  "results": [
    {
      "changes": [
        {
          "rev": "7-
7a5191bfeb2f6eed6b3b6fa83f4810f5"
        }
      ],
      "deleted": true,
      "id": "0007741142412418284",
      "seq": "1-
g1AAAAMFeJydkcENgjAUhquYePHqBN48EEBj4klXUOgAbSExD
UJi8OwUrqDQJZyCJZhBqD8XE0Iiwulvmvf1e68vJITMToZPFj
4X8SXY-3xjijC--ixKLMv-
ns0oSEIUjxnhc0o9yUgTW_3GaojypVKZPBlsdMbdVFhbW6yd_
15p2pwOG0SK77SuevcJSHP06rbILicgj0ul0t5OQBm_aV22yI
511FAVTciR0ntfKzAXGOKJz-
prBpYCQ7y1fg1wl8AQB0rzAW4MnCMeShUD3Bi4QGDTGrT8AMk
```

A DATABASE IN CLOUD

SOLUTIONING OF SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES



Last entry: less than a minute ago
Entries: 8