

# CREATE A TEXT TO SPEECH SERVICE

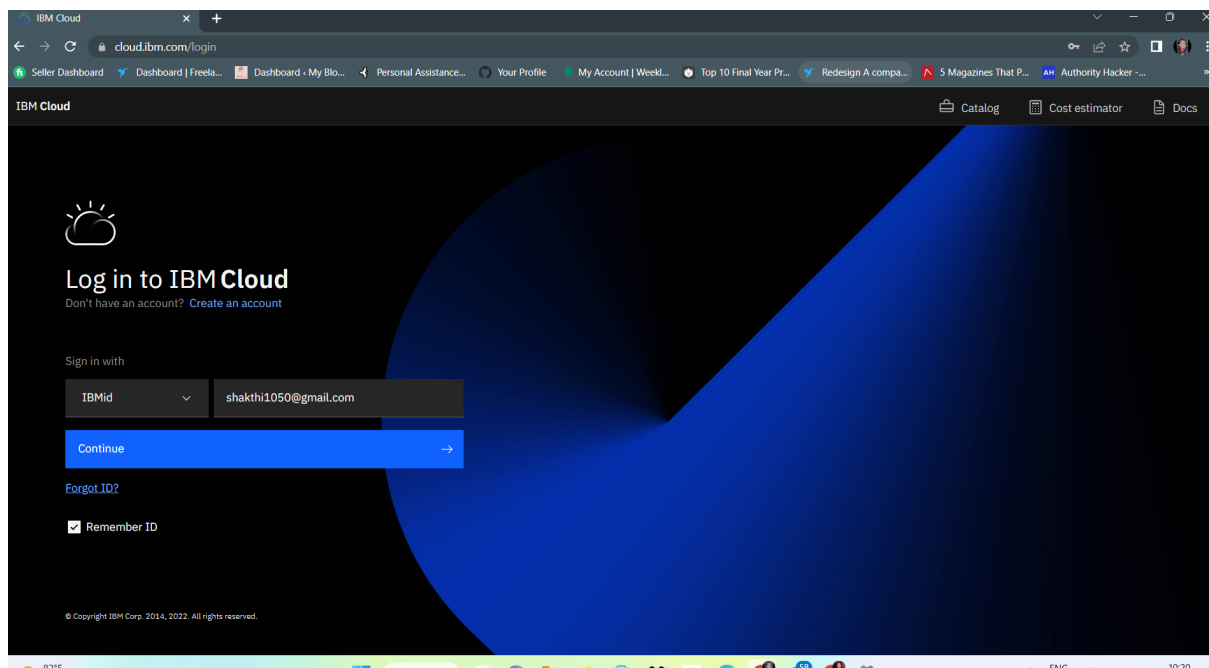
TEAM ID	PNT2022TMID26580
PROJECT NAME	Personal Assistance for Seniors Who Are Self-Reliant.

## AIM :

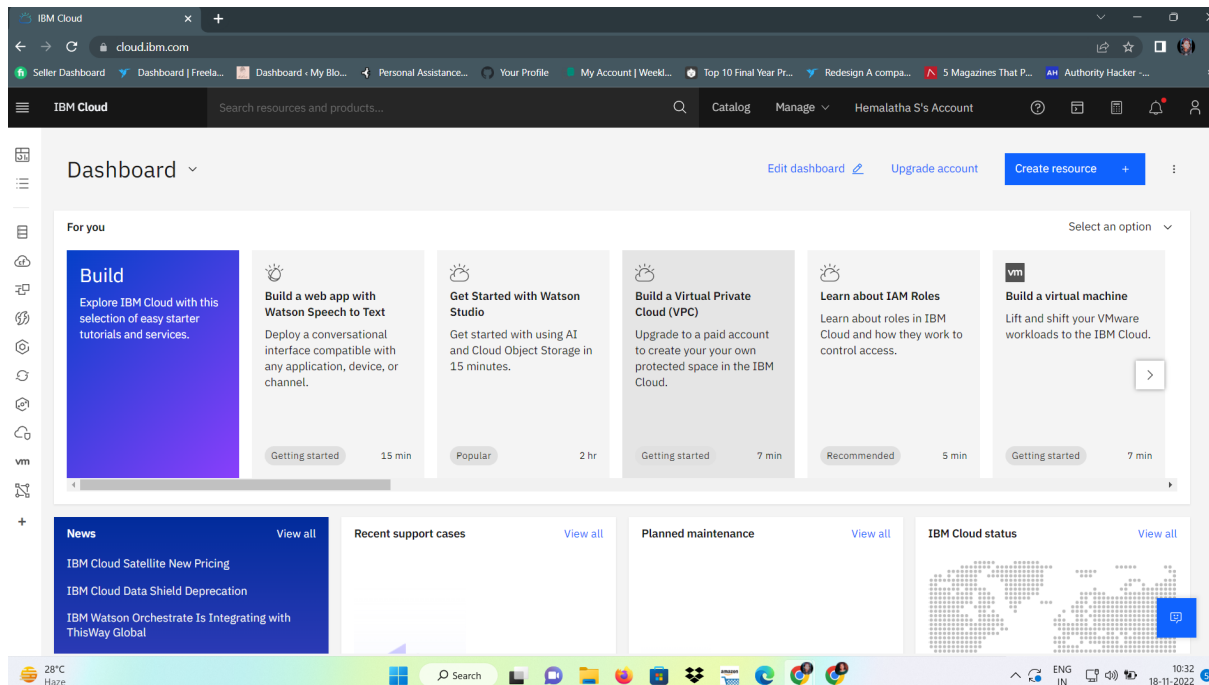
TO CREATE A TEXT TO SPEECH SERVICE IN IBM IOT WATSON PLATFORM

## STEPS TO BE FOLLOWED :

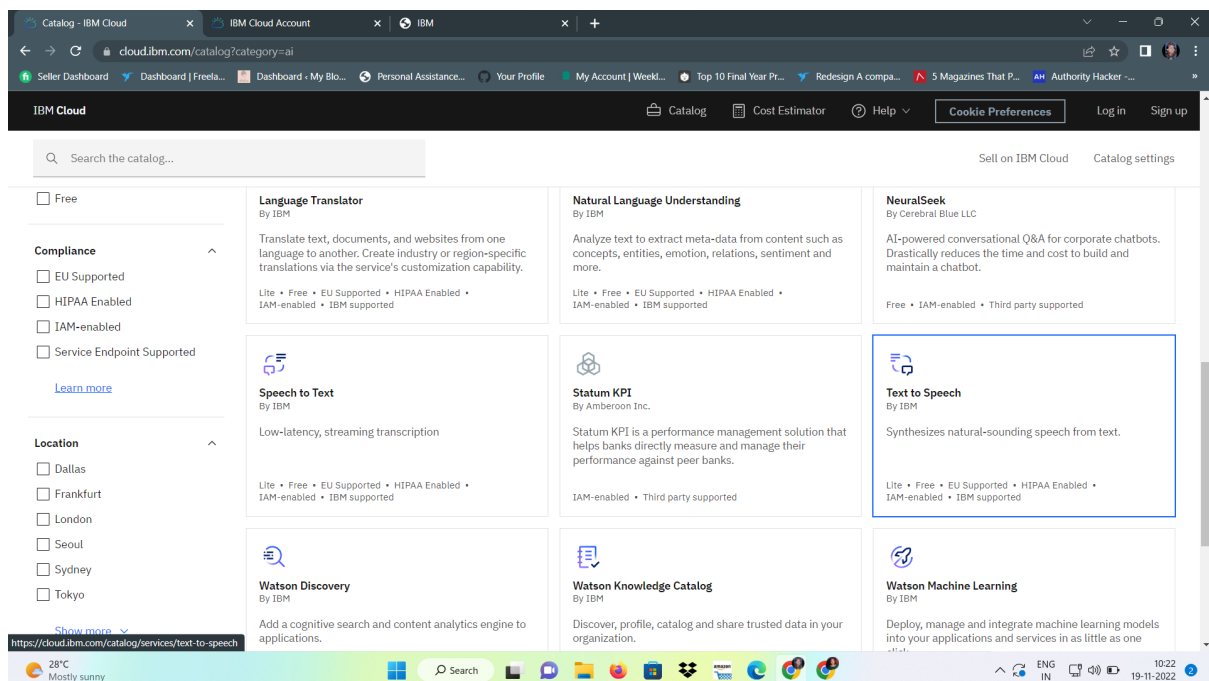
**STEP 1 :** FIRST SHOULD LOGIN IN YOUR IBM CLOUD ACCOUNT IN YOUR BROWSER



## STEP 2 : OPEN CATALOG MENU GIVEN ON THE RIGHT SIDE OF THE TOP



## STEP 3 : IN THE CATEGORY MENU GIVEN IN THE LEFT SIDE, CHOOSE AN ARTIFICIAL INTELLIGENCE OPTION.



## STEP 4 : JUST SCROLL DOWN, SEARCH AND SELECT THE “ TEXT TO SPEECH SERVICE “

The screenshot shows the IBM Cloud catalog page for the Text to Speech service. The page is titled "Text to Speech" and describes it as a service that synthesizes natural-sounding speech from text. It features a "Create" button and a "Pricing plans" section. The pricing table lists three plans: Lite, Standard, and Premium. The Lite plan is free and offers 10,000 characters per month. The Standard plan costs \$0.02 USD per thousand characters. The Premium plan includes everything in the Standard plan plus additional features like private storage and high availability. A sidebar on the right shows the service is free and includes a "Sign up to create" button. The bottom of the page shows a weather widget and system icons.

Plan	Features	Pricing
Lite	10,000 Characters per Month	Free
Standard	Standard Characters	\$0.02 USD/THOUSAND CHAR
Premium	Everything in Standard plus... Usage and Training Data is Private + Stored in an Isolated Single Tenant Environment High Availability and Service Level Uptime Guarantee IBM Cloud Service Endpoints HIPAA - Washington DC Only	

## STEP 5 : READ AND FILL CAREFULLY WITH REQUIRED DATA AND TO CREATE AN ACCOUNT

The screenshot shows the IBM Cloud "Getting started with Text to Speech" page. It provides a tutorial for using the service, including a video titled "Get started with Watson Text to Speech". The page includes a sidebar with navigation links for "Manage", "Getting started", "Service credentials", "Plan", and "Connections". A "Reading list" sidebar on the right shows a list of articles. The main content area includes a video player and a "Before you begin" section. The bottom of the page shows a weather widget and system icons.

**Getting started with Text to Speech**  
Last Updated: 2021-02-02

The IBM Watson™ Text to Speech service converts written text to natural-sounding speech to provide speech-synthesis capabilities for applications. This `curl`-based tutorial can help you get started quickly with the service. The examples show you how to call the service's `POST` and `GET /v1/synthesize` methods to request an audio stream.

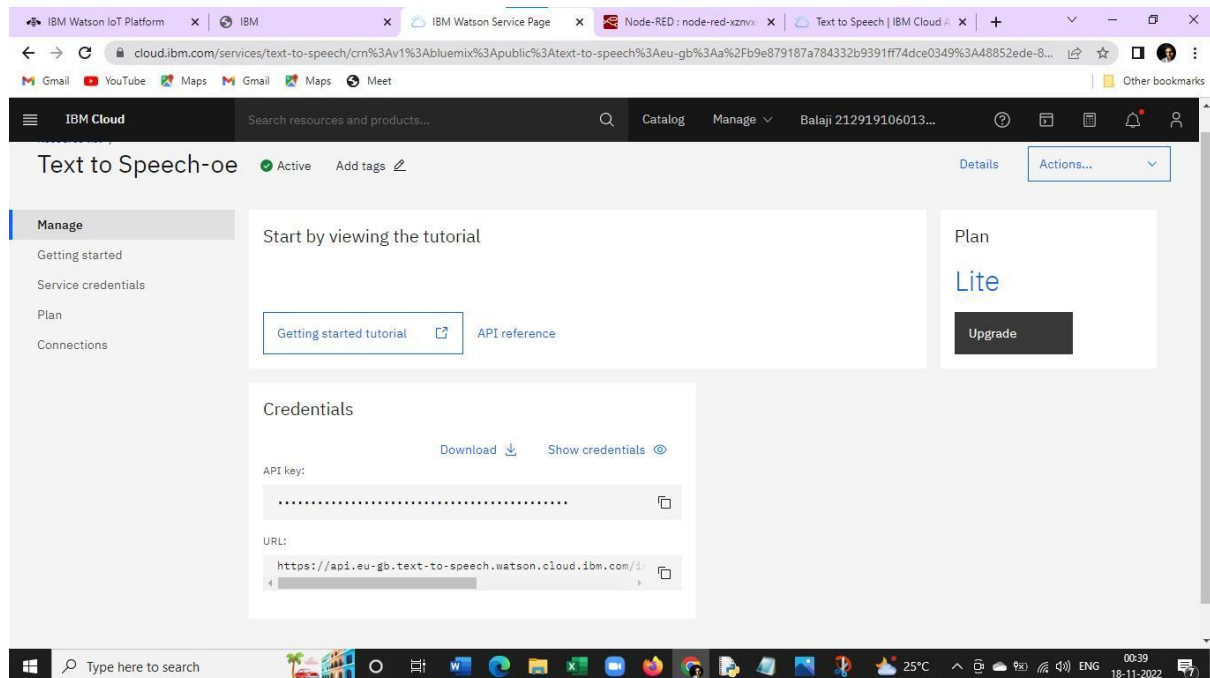
IBM Cloud only **IBM Cloud® only**. Watch the following video for a visual summary of getting started with the Text to Speech service.

**Get started with Watson Text to Speech**

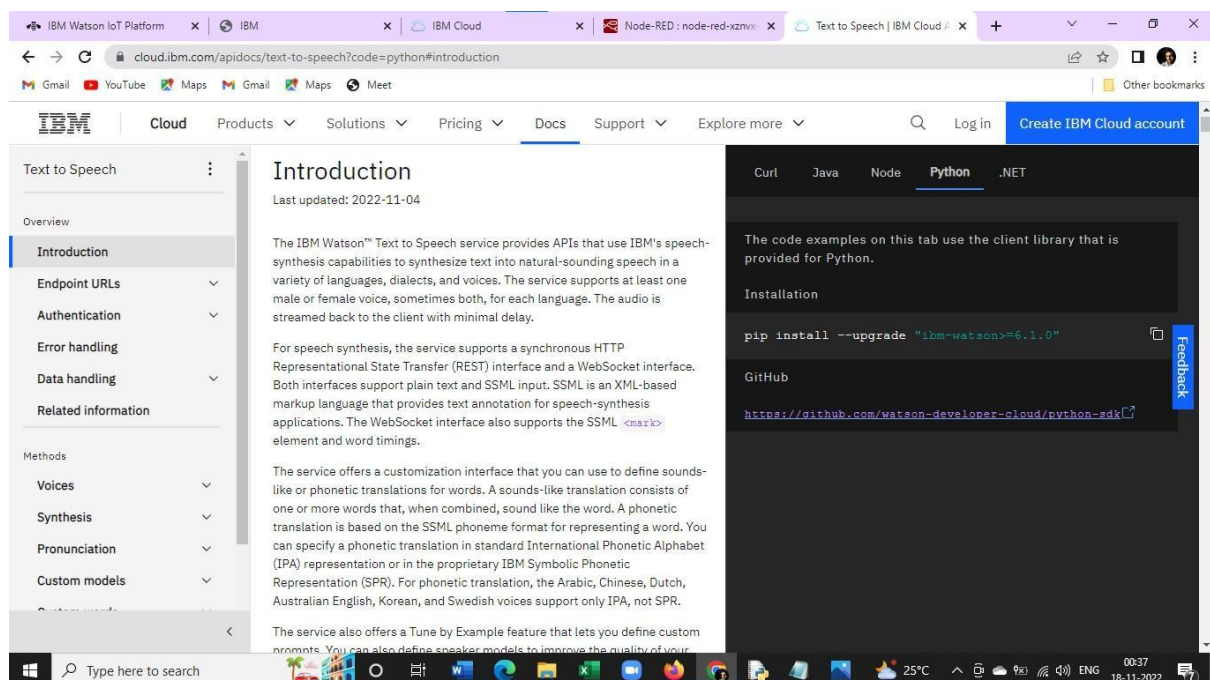
**Are you experiencing trouble?**  
Click here for help to resolve the problem.

**Before you begin**

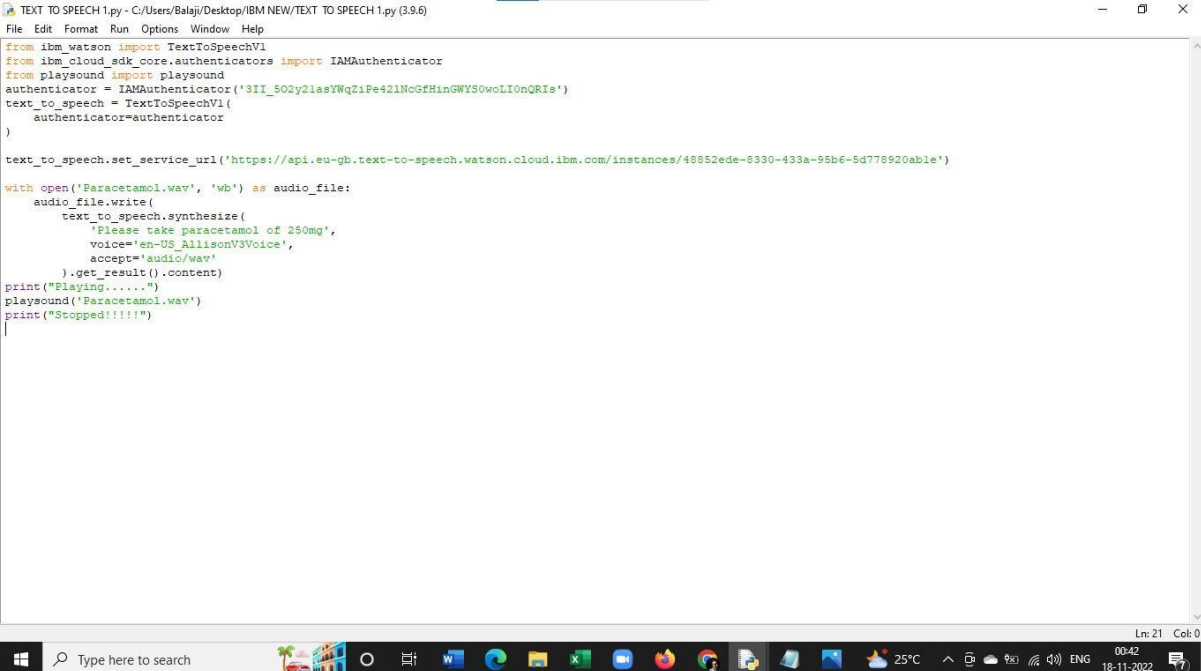
## STEP 6 : AFTER CREATING IT WILL GENERATES API KEY AND URL TAKE IT DOWN CAREFULLY



## STEP 7 : INSTALL USING THE GIVEN COMMAND IN YOUR PYTHON SOFTWARE



## STEP 8 : AND NOW START CODING IN YOUR PYTHON SOFTWARE AND GET RESULTS

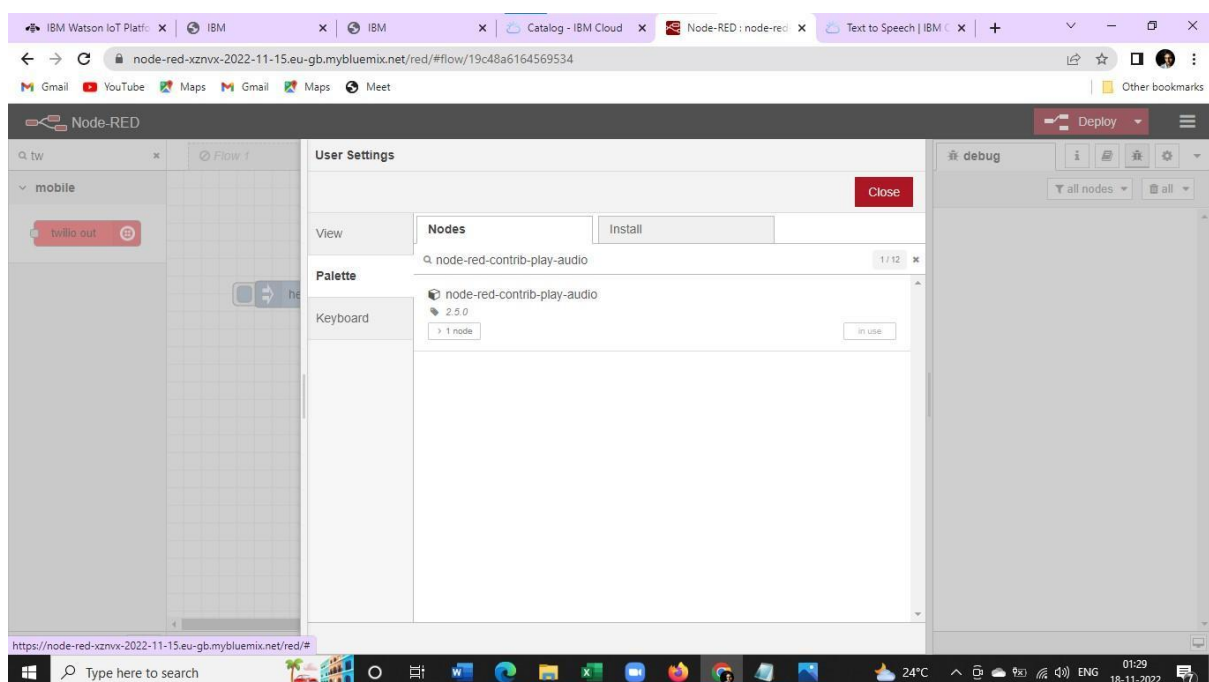


```
TEXT TO SPEECH 1.py - C:/Users/Balaji/Desktop/IBM NEW/TEXT TO SPEECH 1.py (3.9.6)
File Edit Format Run Options Window Help
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound
authenticator = IAMAuthenticator('3II_5O2y2lasYWqZ4Pe42lNoGfHnGWYS0woLIonQRIs')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

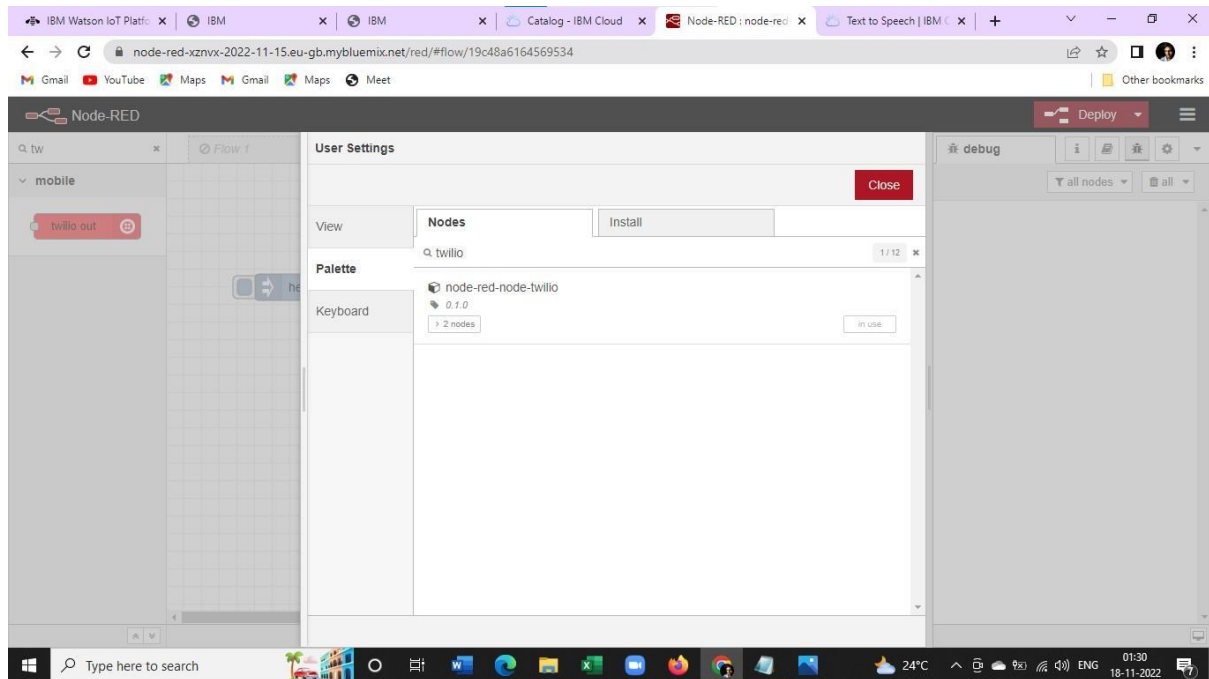
text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/48852ede-8330-433a-95b6-5d778920able')

with open('Paracetamol.wav', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            'Please take paracetamol of 250mg',
            voice='en-US_AllisonV3Voice',
            accept='audio/wav'
        ).get_result().content
    )
print("Playing.....")
playsound('Paracetamol.wav')
print("Stopped!!!!!!")
```

## STEP 9 : BY USING NODE RED SERVICES INSTALL A NODE-RED-PLAY-AUDIO TOOL



**STEP 10 :** ALSO INSTAL TWILIO FOR SENDING VOICE SMS AND CALLS (NOTE IN TWILIO WEB SITE NUMBERS SHOULD BE VERIFIED THEN ONLY WE WILL GET API KEYS AND AUTH TOKENS AND ALSO RECEIVE CALL OR VOICE NOTES )



**STEP 11 :** IN TWILIO NODE INJECT THE RECEIVER DETAILS AFTER DEPLOY AND INJECT THE MSG.PAYLOAD

IBM Watson IoT Platform | IBM | IBM | Catalog - IBM Cloud | Node-RED: node-red | Text to Speech | IBM

node-red-xznvx-2022-11-15.eu-gb.mybluemix.net/red/#flow/19c48a6164569534

Gmail | YouTube | Maps | Gmail | Maps | Meet | Other bookmarks

Node-RED

Successfully injected: hello

Deploy

mobile

twilio out

Flow 1

Flow 2

debug

all nodes

all

hello

text to speech

play audio

speaking

twilio

Type here to search

24°C

ENG

01:31

18-11-2022

The screenshot shows the Node-RED web interface in a browser. The top bar includes several open tabs: 'IBM Watson IoT Platform', 'IBM', 'IBM', 'Catalog - IBM Cloud', 'Node-RED: node-red', and 'Text to Speech | IBM'. The address bar shows the URL 'node-red-xznvx-2022-11-15.eu-gb.mybluemix.net/red/#flow/19c48a6164569534'. Below the browser tabs, there are quick links to 'Gmail', 'YouTube', 'Maps', 'Gmail', 'Maps', and 'Meet', along with 'Other bookmarks'. The main interface is the Node-RED editor. On the left, the 'mobile' palette contains a 'twilio out' node. The central workspace shows a flow named 'Flow 2' with a 'hello' input node connected to a 'text to speech' node, which then branches into 'play audio' and 'twilio' output nodes. A status bar at the top of the editor says 'Successfully injected: hello'. On the right, there is a 'debug' console and a 'Deploy' button. The bottom of the image shows a Windows taskbar with a search bar, various application icons, and system information including '24°C', 'ENG', and the time '01:31' on '18-11-2022'.

**RESULT :**

VOICE WILL BE GENERATED FOR THE GIVEN INPUT TO  
THE RECEIVER.