CLOUD APPLICATION DEVELOPMENT (GROUP 1)

PHASE 1 : ASSIGNMENT NOTEBOOK SUBMISSION

NAME : ABITHA SUNDAR.S

EMAIL : sundarabitha8@gmail.com

GitHub Repository URL : https://github.com/Abitha63abi/Machine-learning.git

Title of the project :

MACHINE LEARNING MODEL DEPLOYMENT WITH IBM CIOUD WATSON STUDIO

Aim: To deploy a machine learning algorithm in Watson studio and predict the output by sending the sensor data from the IOT Device using IOT Platform and node-red Service.

Requirements:

- IOT Device integrated with sensors
- Data Set for Project.
- IBM cloud account

Outcome:

In this tutorial, you will explore the following services.

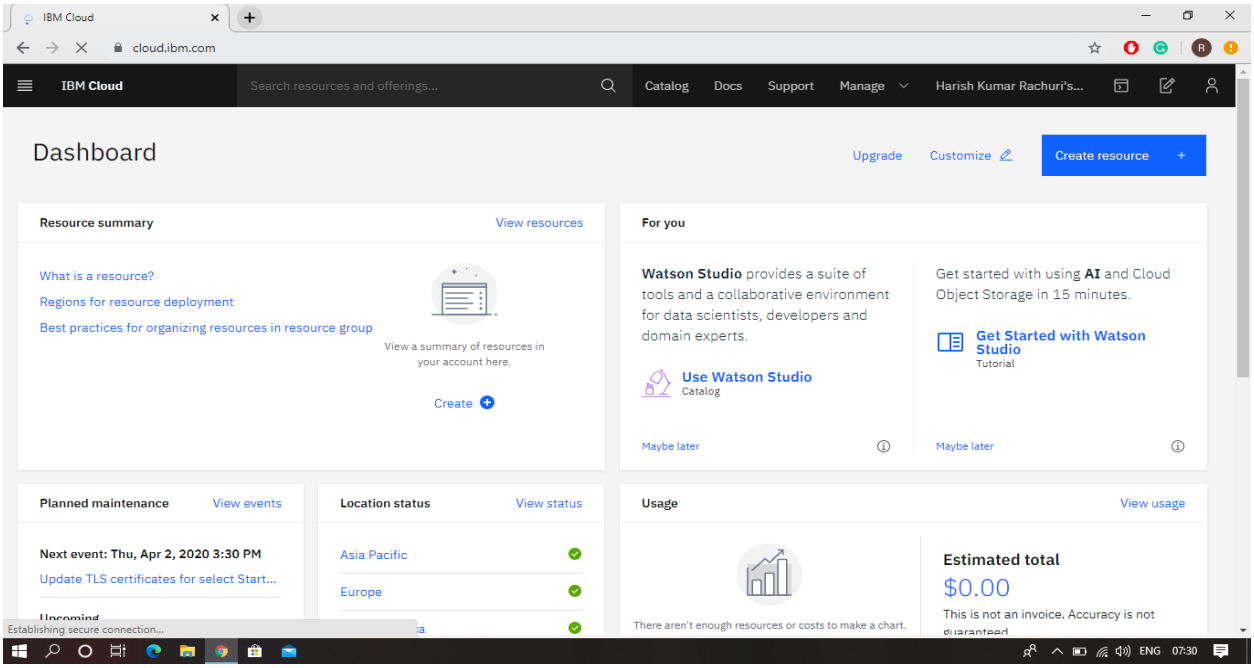
- Watson Studio service
- Cloud object Storage
- Machine learning instance
- Training the machine learning model with data set in Auto AI experiment
- Node-red application
- IBM IOT WATSON PLATFORM

Activities:

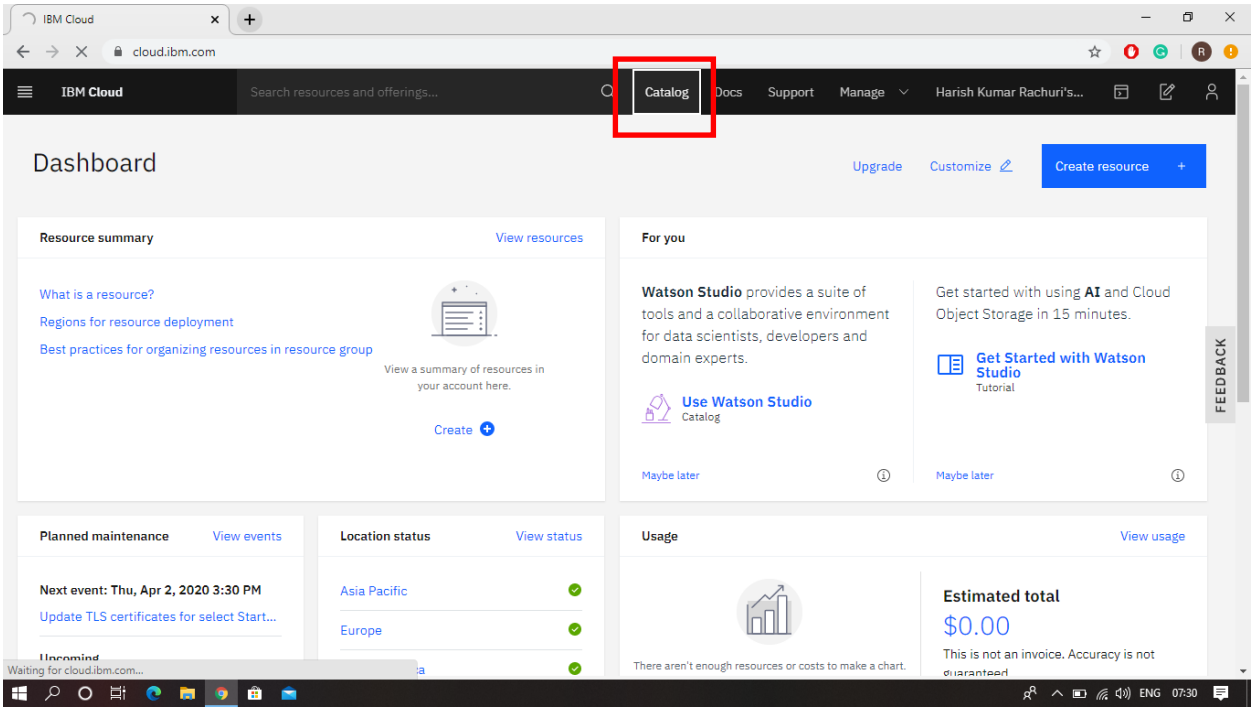
- Create the Watson Studio.
- Adding the cloud object storage.
- Create Auto AI experiment.
- Training and Deploying the Auto AI Experiment.
- Configure the Node Red Application to get the sensor data from IOT platform
- Predicting the output by providing sensor data as an input to machine learning model.
- Creating the Web UI to visualize the data

Task-1: Create the Watson Studio Instance

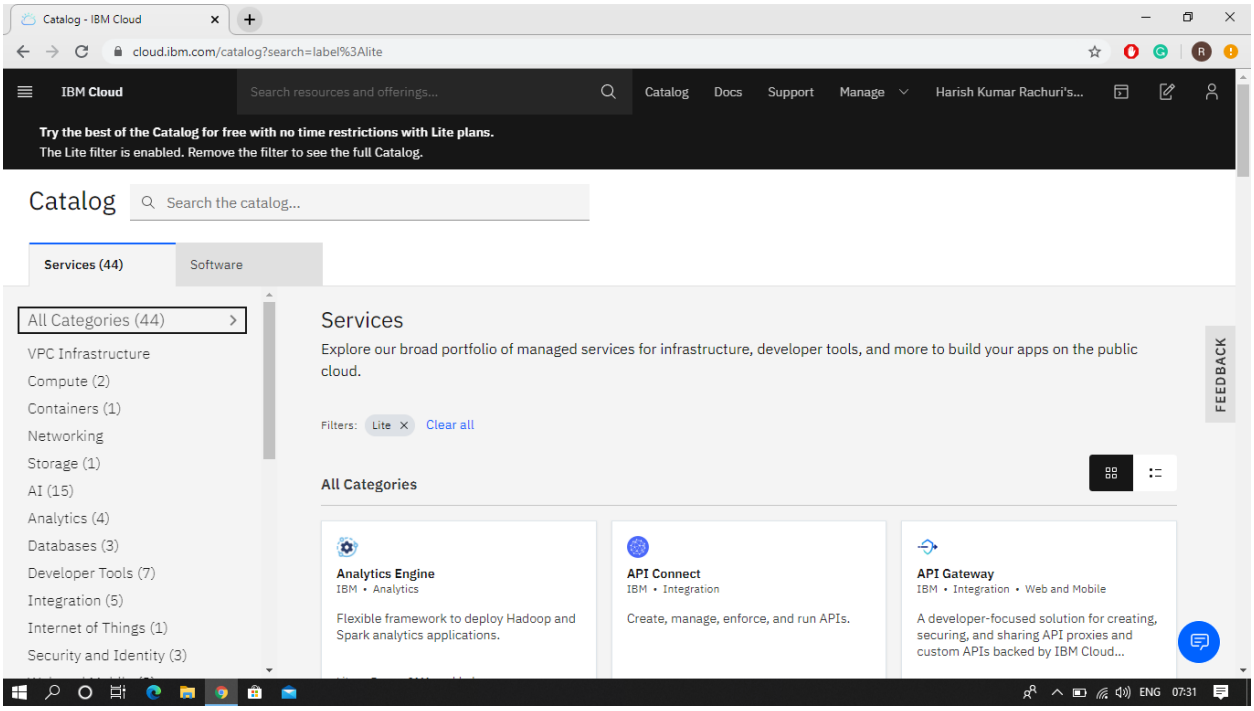
- After logging into the IBM Account, navigate to Dashboard



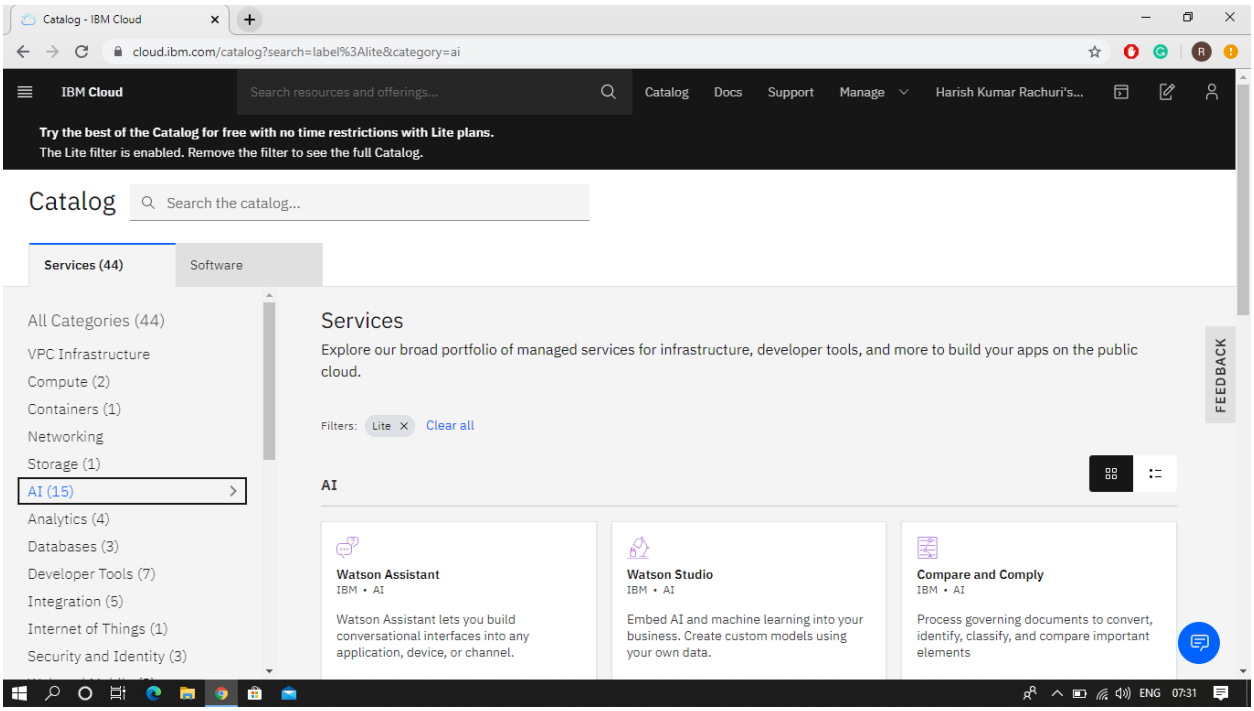
- Click on the Catalog to view the various services present in IBM cloud



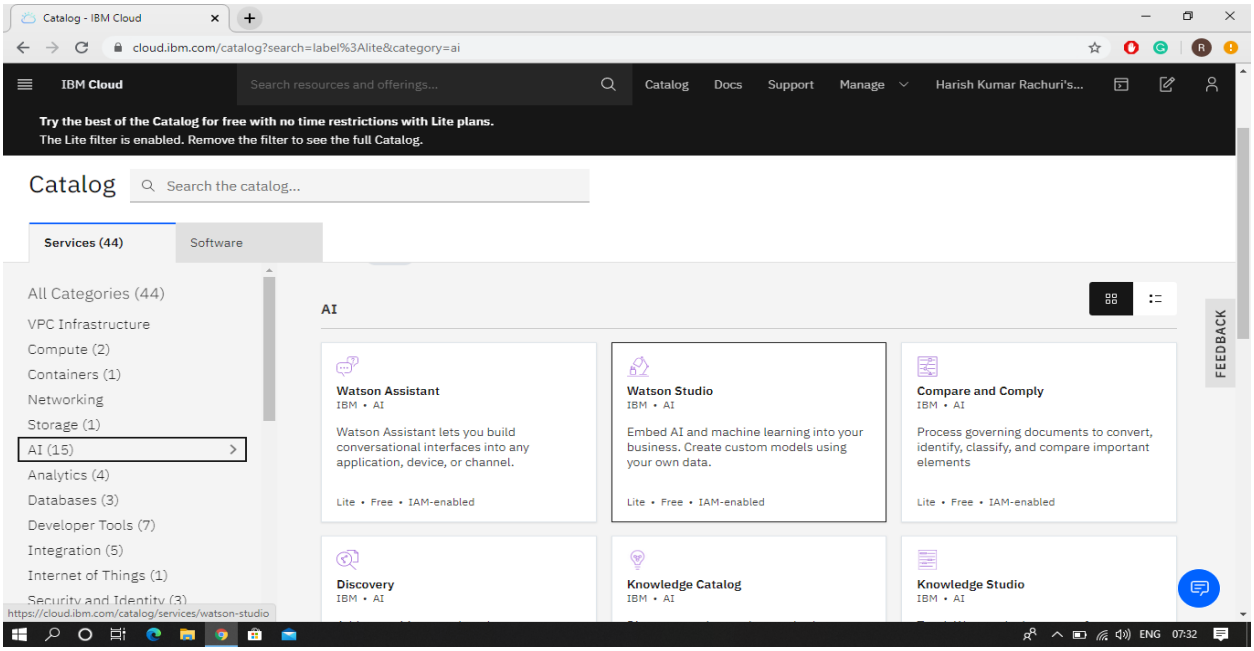
- Click on all Categories



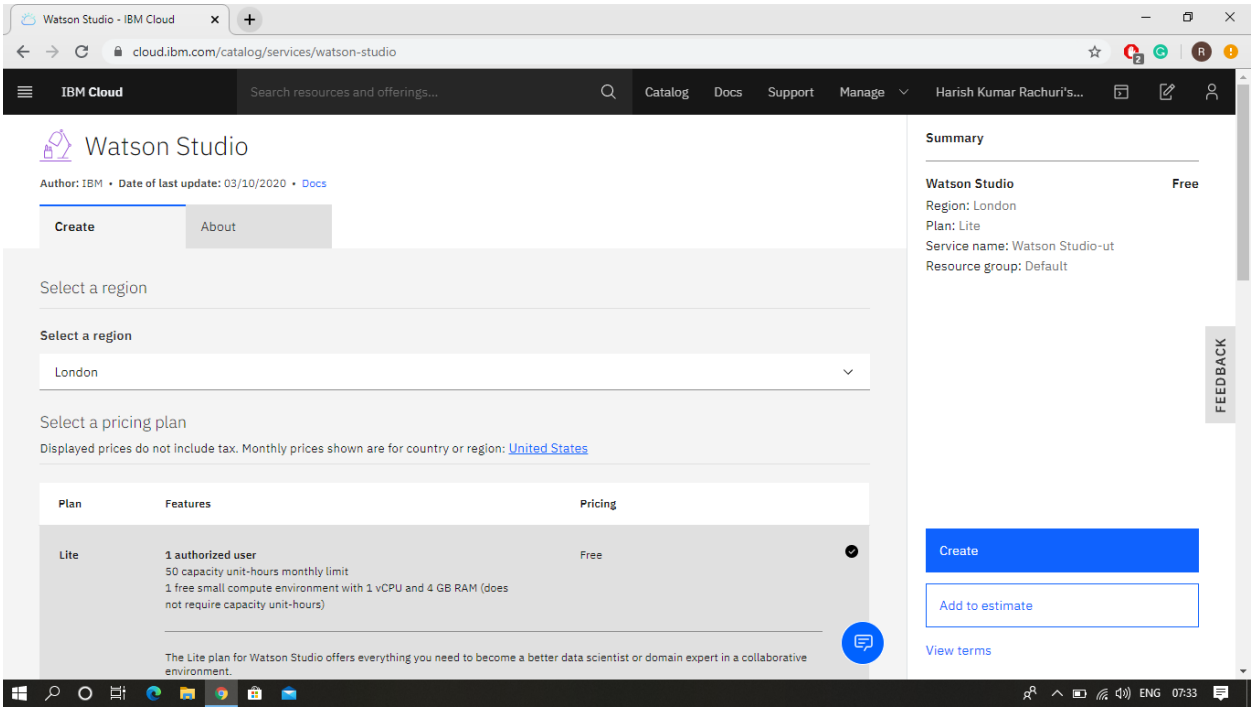
- Select the category AI.



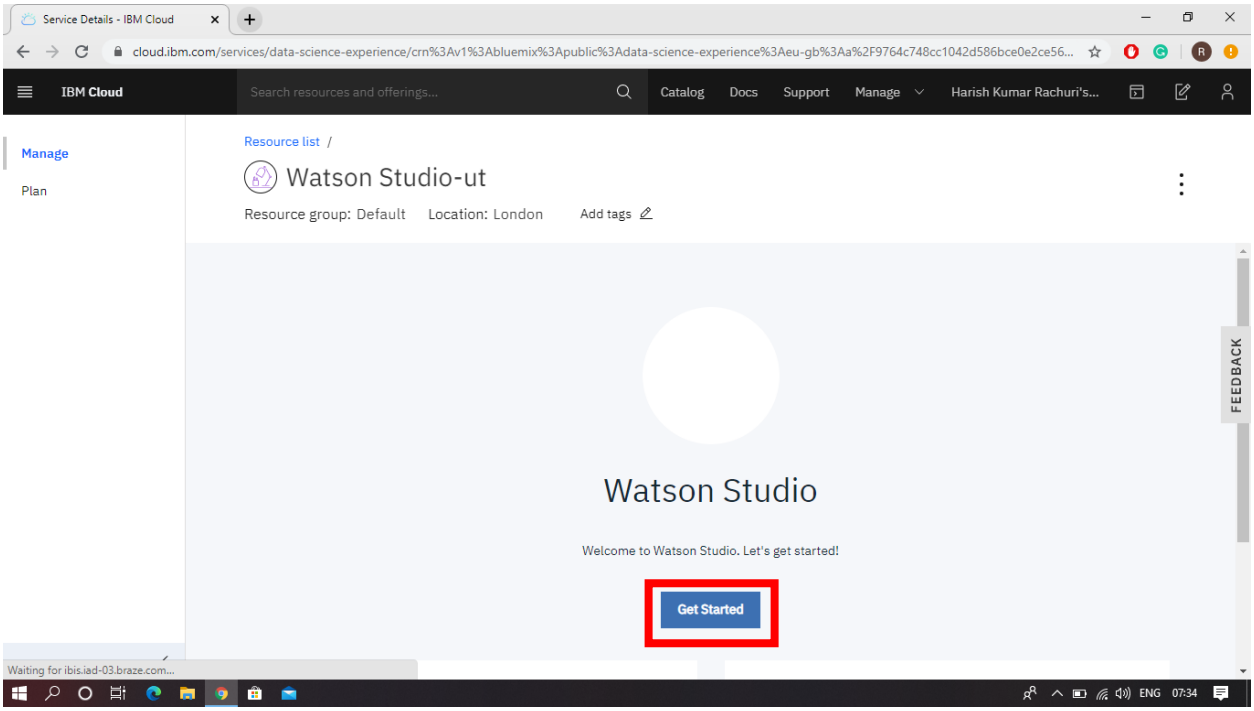
- Click on the Watson Studio



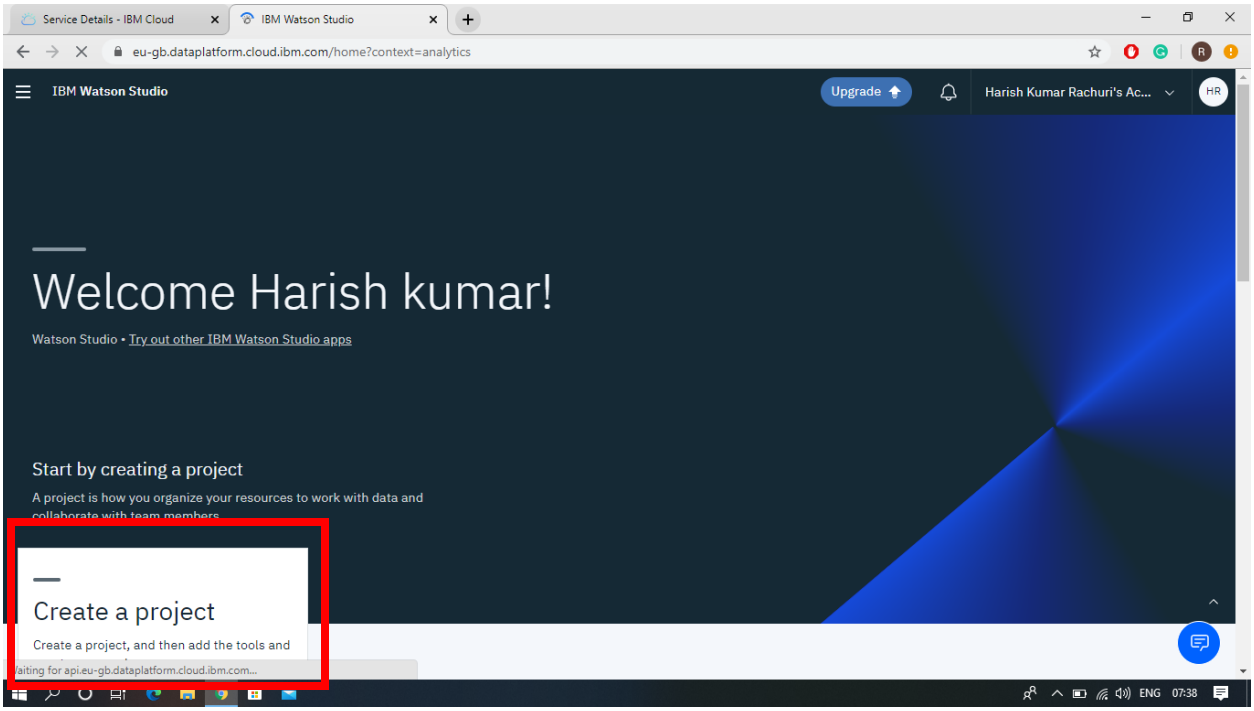
- You will be redirected to Watson studio page where you can create your Watson studio service instance. There you click on create button.
-



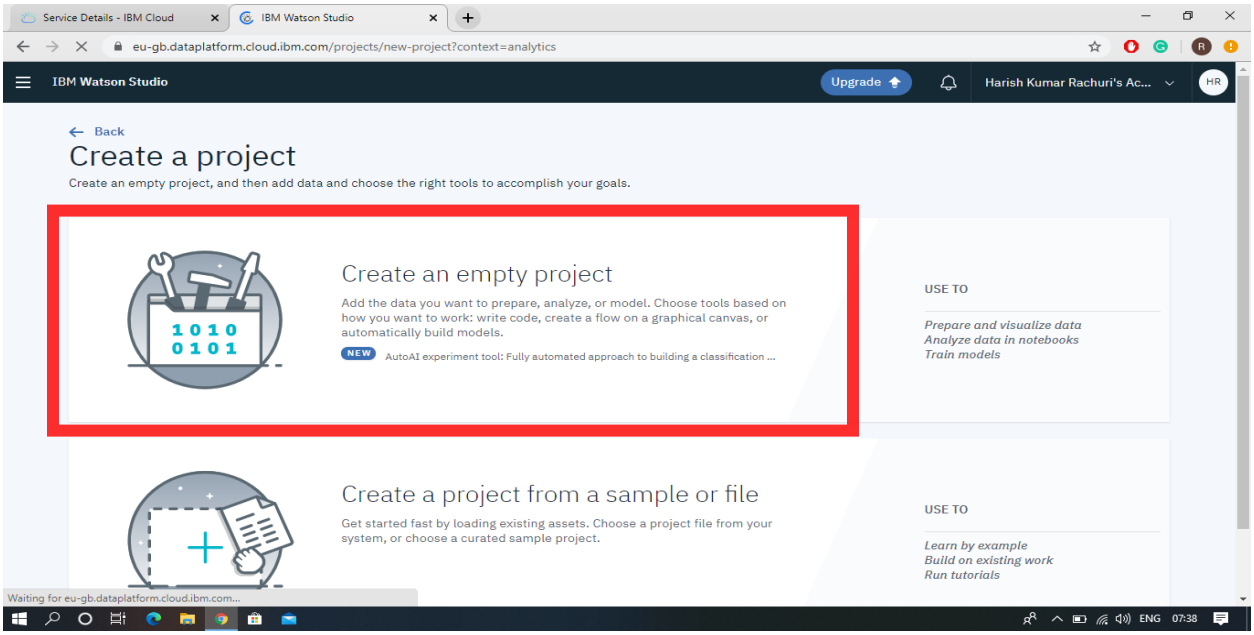
- Now you will be redirected to the below page there click on Get Started



- It will redirect to new page where you can create project, click on it.

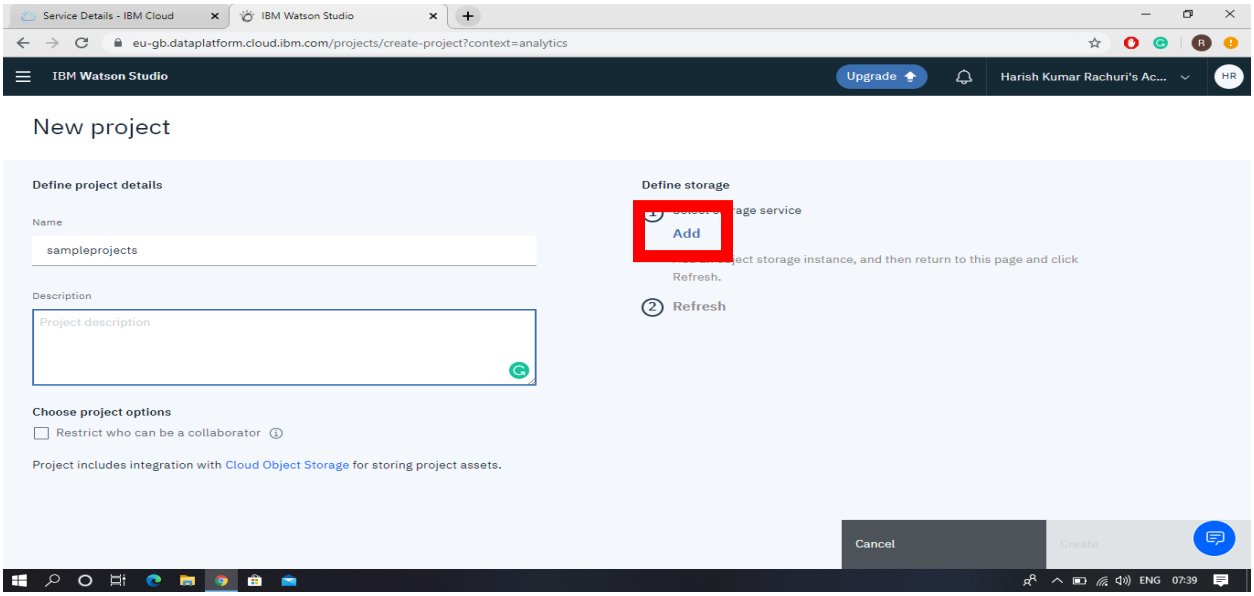


- Click on Create an Empty Project

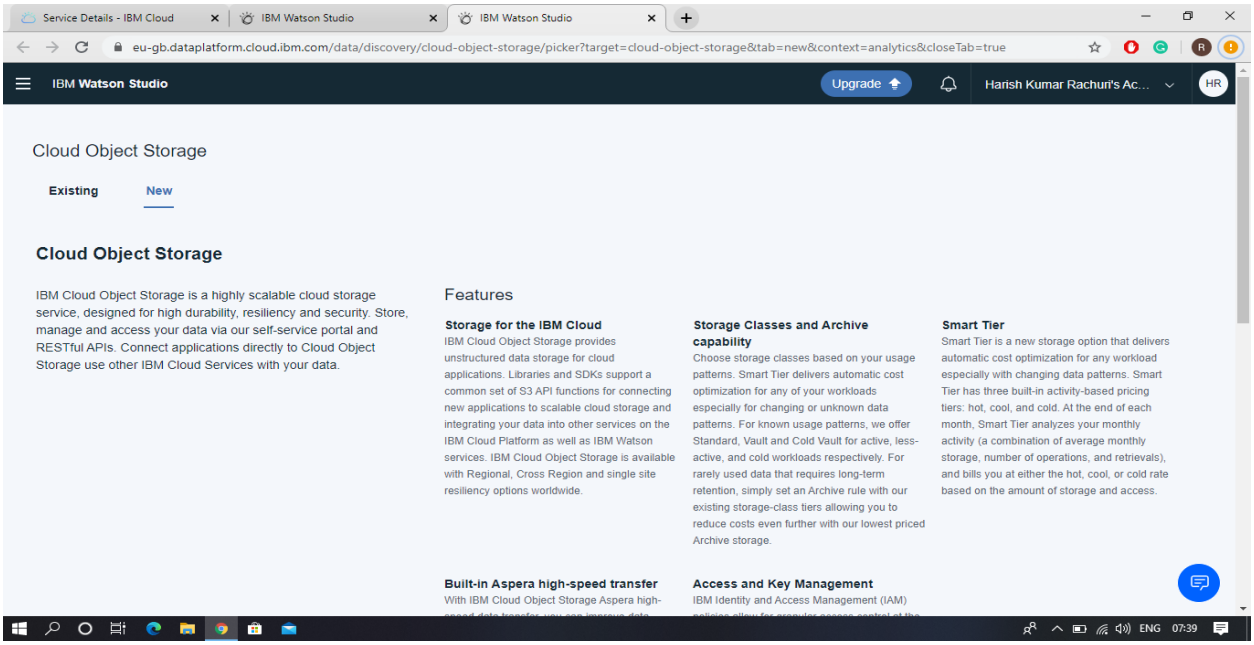


Task 2: Adding the Object Storage to project

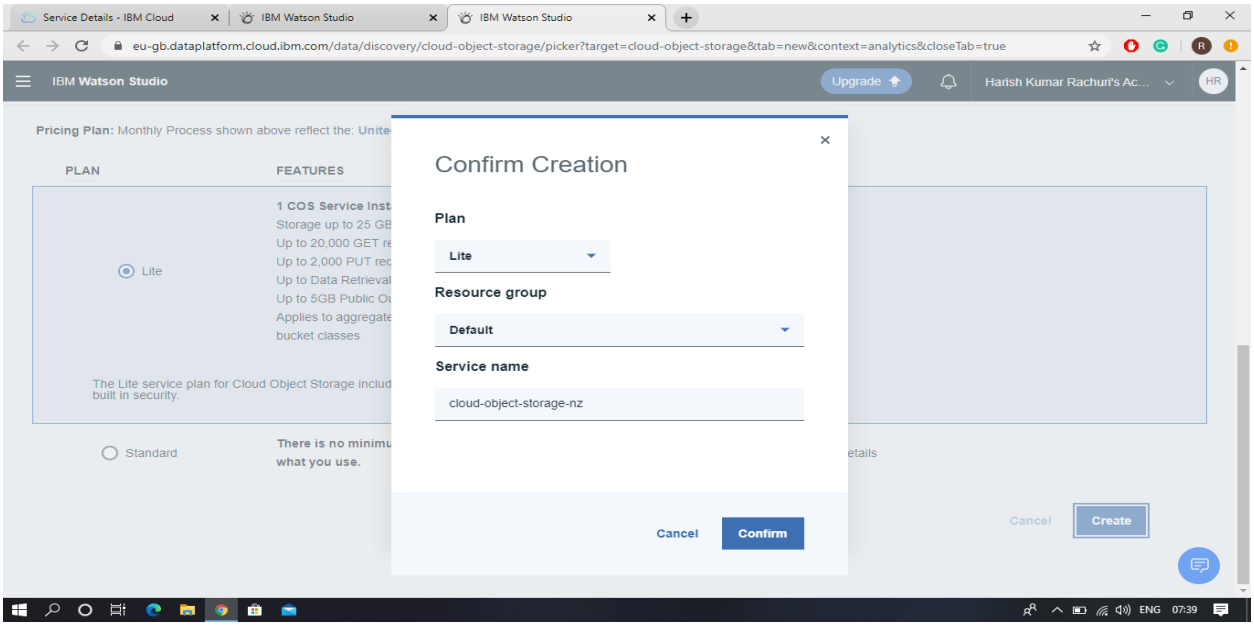
- While creating the project you need to provide name and you need to select the object storage service, on the right side under the object storage option you have an ADD option click on that.



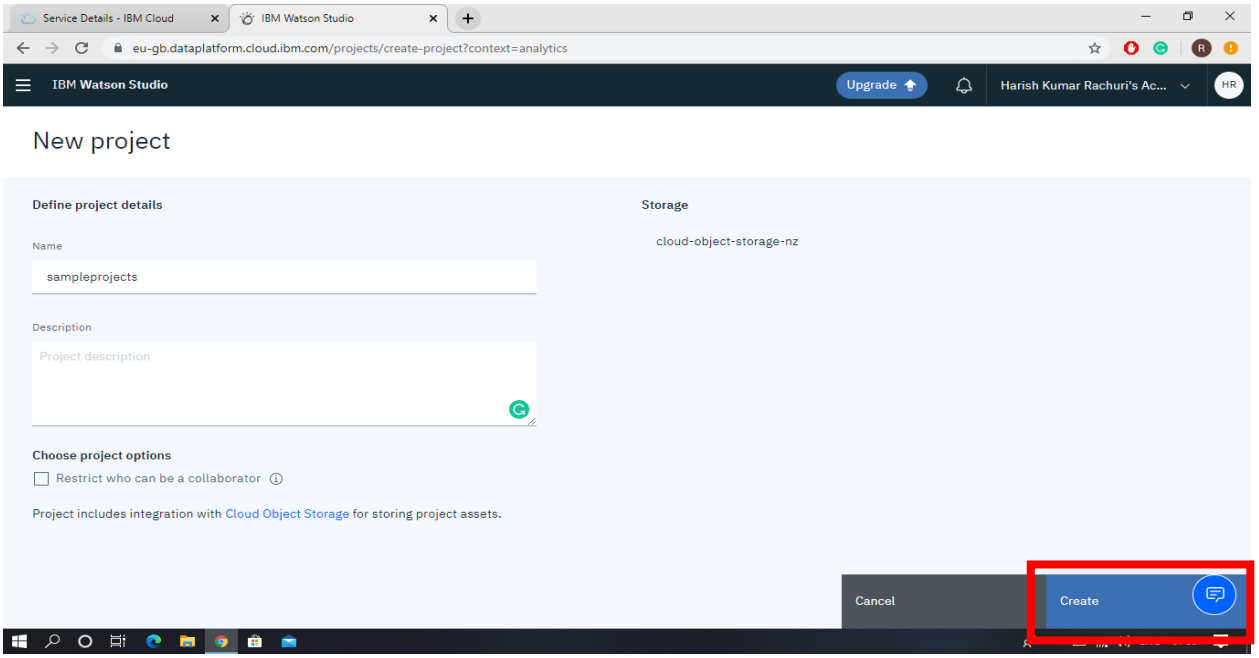
- It will redirect to the new tab where we need to add the Cloud Object storage service. If you have already created a cloud object storage service, you can select the existing instance. Or you need to create the new service instance.



- Scroll Down in the page and click on the Create button, now you confirm it by pressing confirm button.

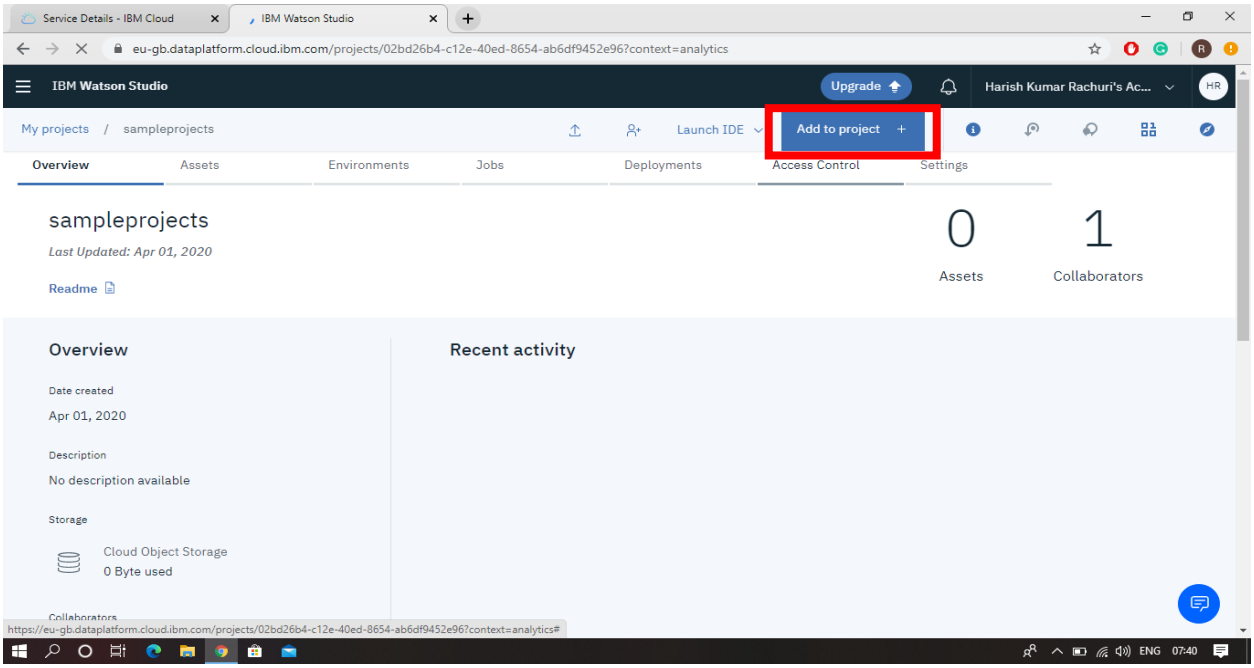


- After Successful creation of object Storage Service, it will redirect to pervious page there you need to select the object storage service which you have created in your previous step. If you dint get your service name, simply click on the refresh button there.
- After selecting the Cloud Object Storage service click on create. Then your project will be created.

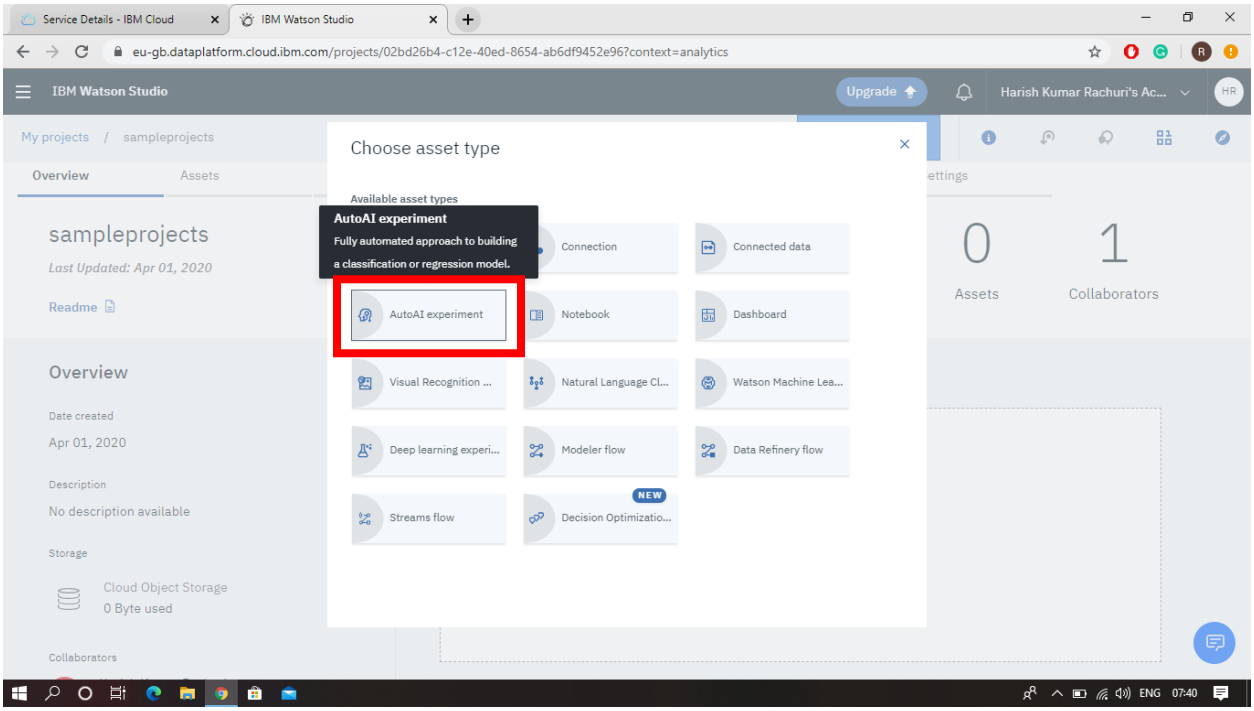


Task-3: Creating an Auto AI Experiment

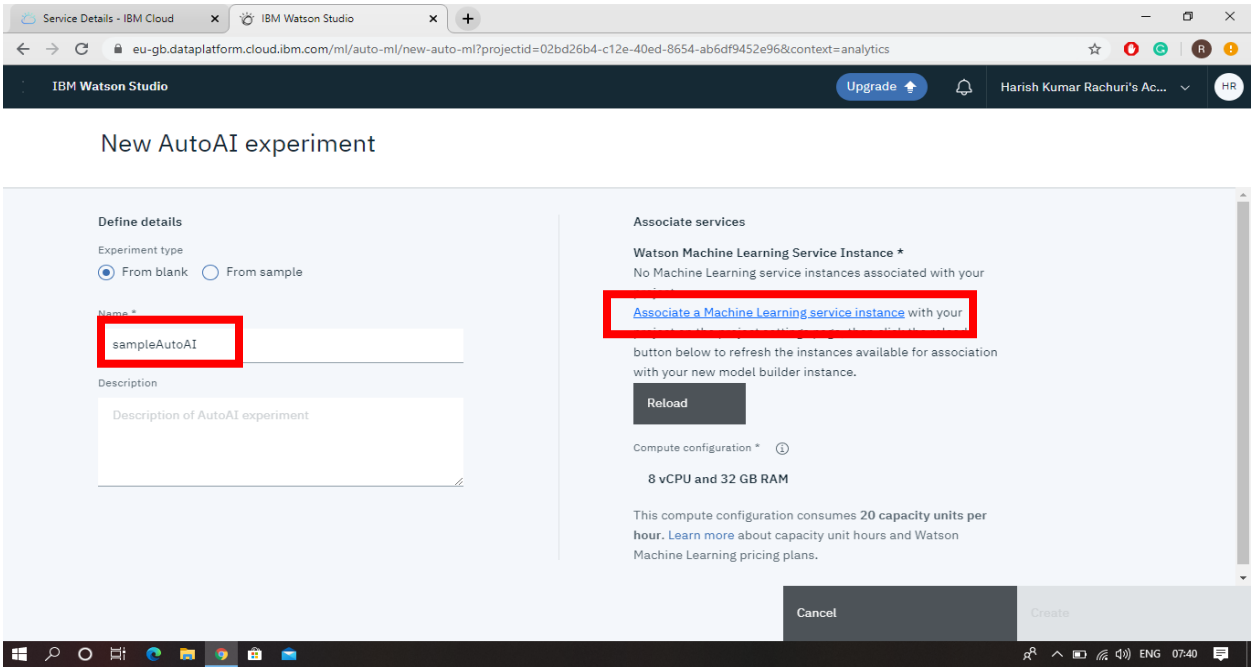
- After creating the project, you will get the below page.
- On the top of screen you will have “add to project” option, Click on Add to project.



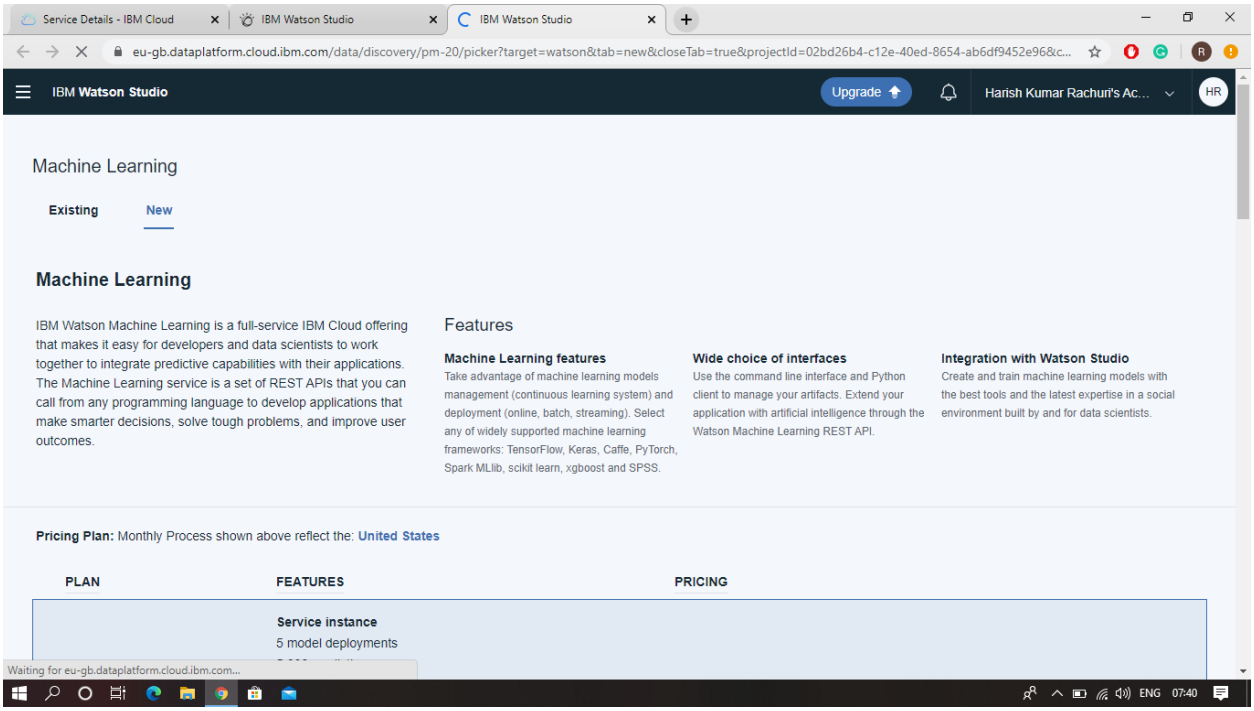
- Here you need to Select AUTO AI experiment as the asset type



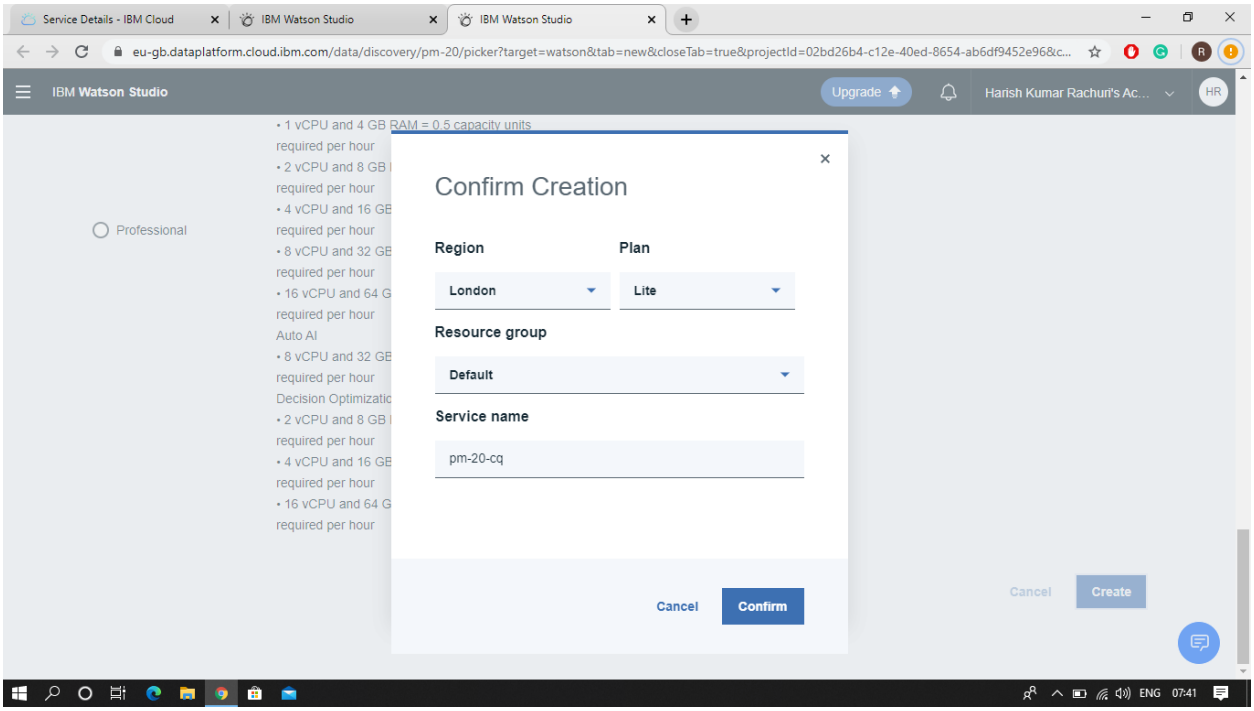
- In this page we need to create Auto AI experiment and we need to select the Watson machine learning service instance.
- Give some name to your service and click on Associate a Machine Learning Service instance.



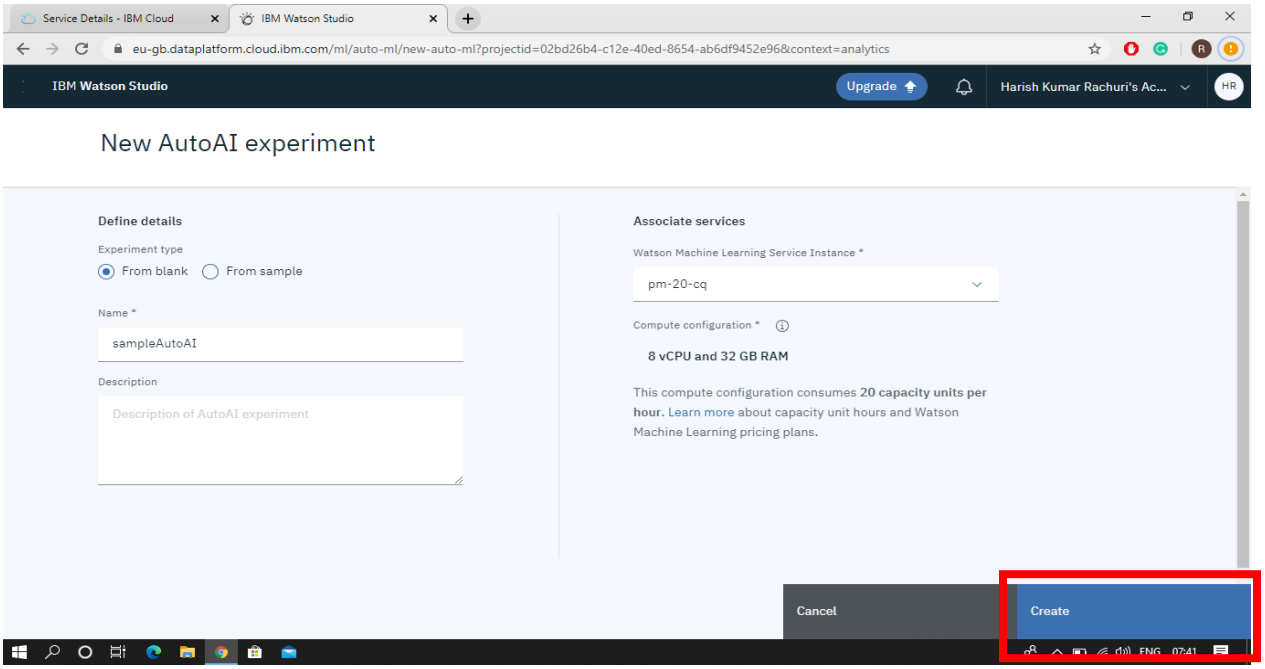
- It will open the new tab where we need to create the machine learning instance. Scroll down in that page and Click on create.



- Yu will get a pop up showing some info regarding your service, click on confirm.



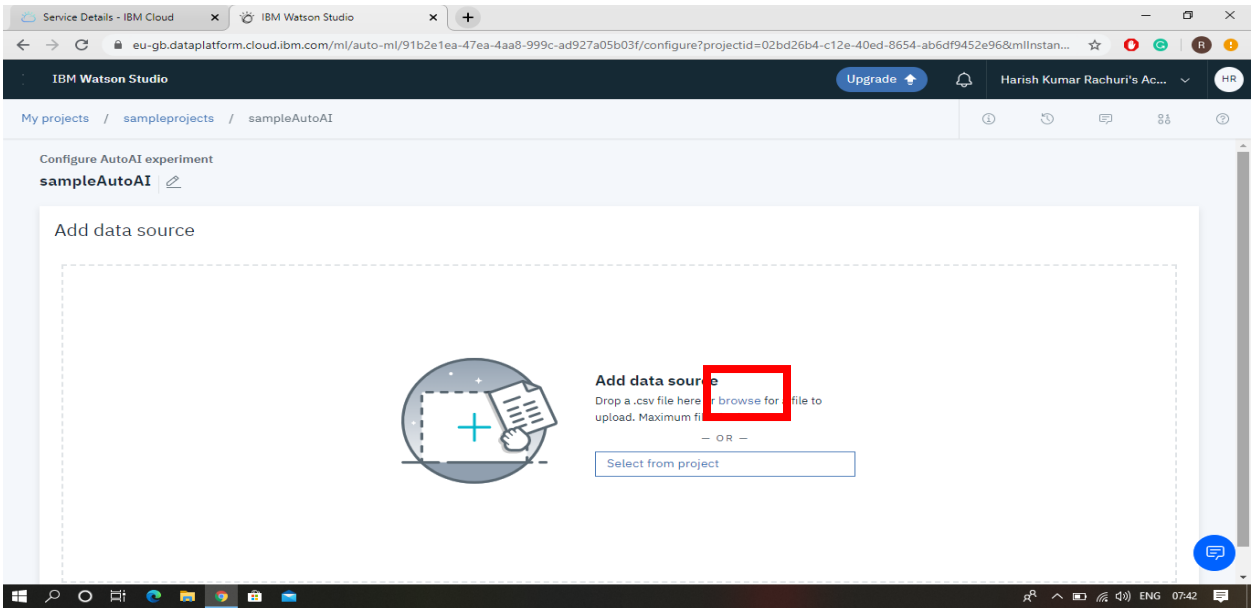
- Once the Machine Learning Instance is created successfully it will redirect to previous page there you reload you page and select your machine learning instance which you have created and click on create.



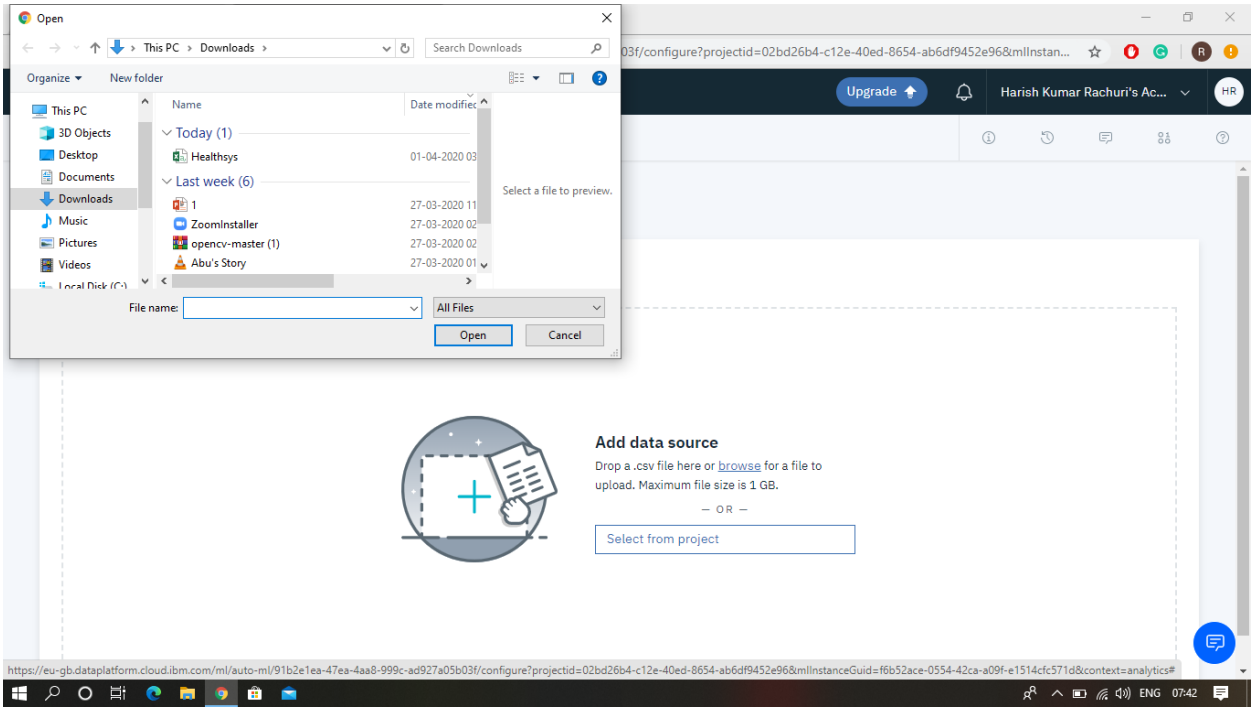
- Now with the above steps you have created a project in Watson studio and added a auto AI experiment as an asset to that project and you have linked a machine learning instance to your project.

Task 4: Training and Deploying the Auto AI Experiment.

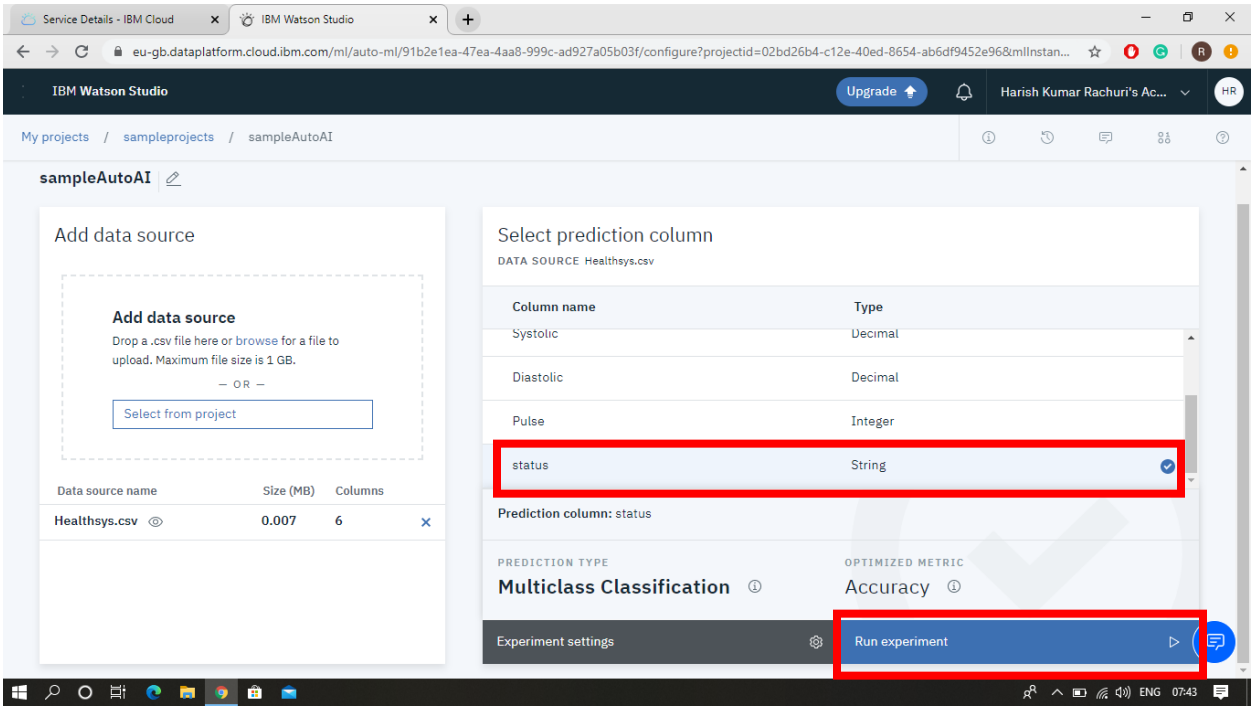
- After successfully creating a AutoAI Experiment, we need to upload the Data Set.
- Just click on Browse option and select your dataset.



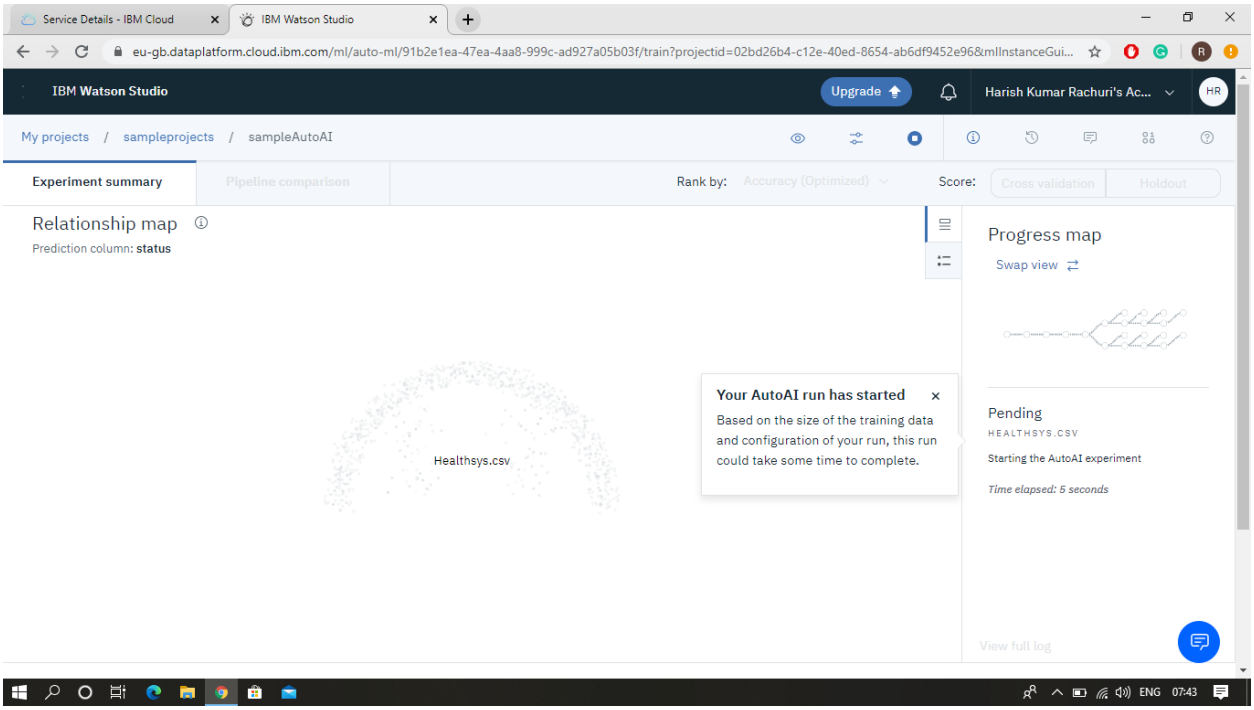
- It will open the Window we need to navigate to folder where Data Set is present. Select the Data Set and click on Open.
- The dataset should be in .csv format.



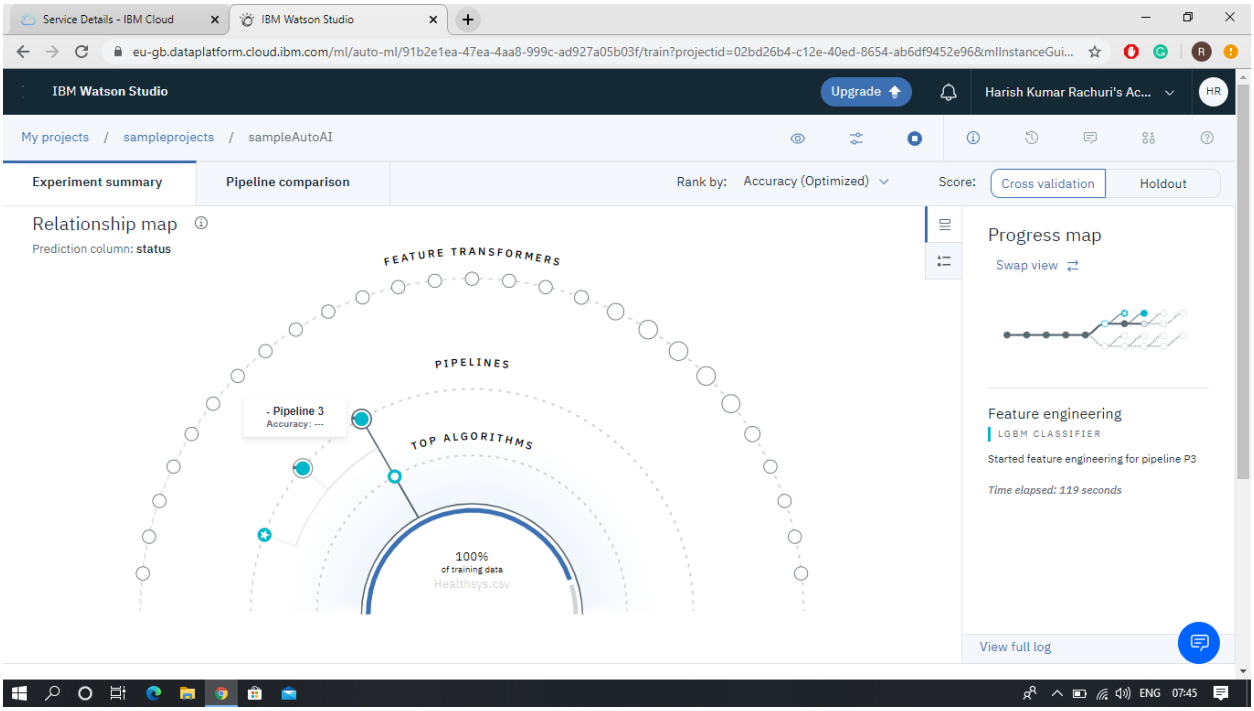
- After selecting the file ,Select the prediction column from the data set and click on Run Experiment.



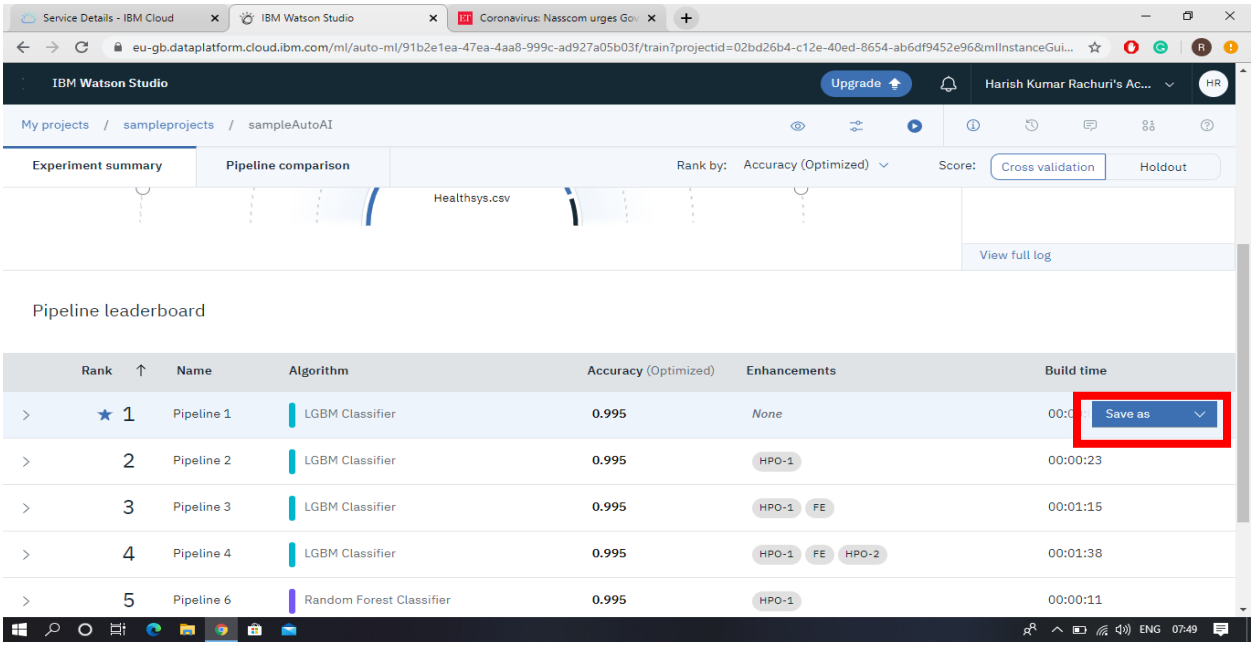
- After clicking on Run Experiment it will load the page with Progress map and it will start training your model.



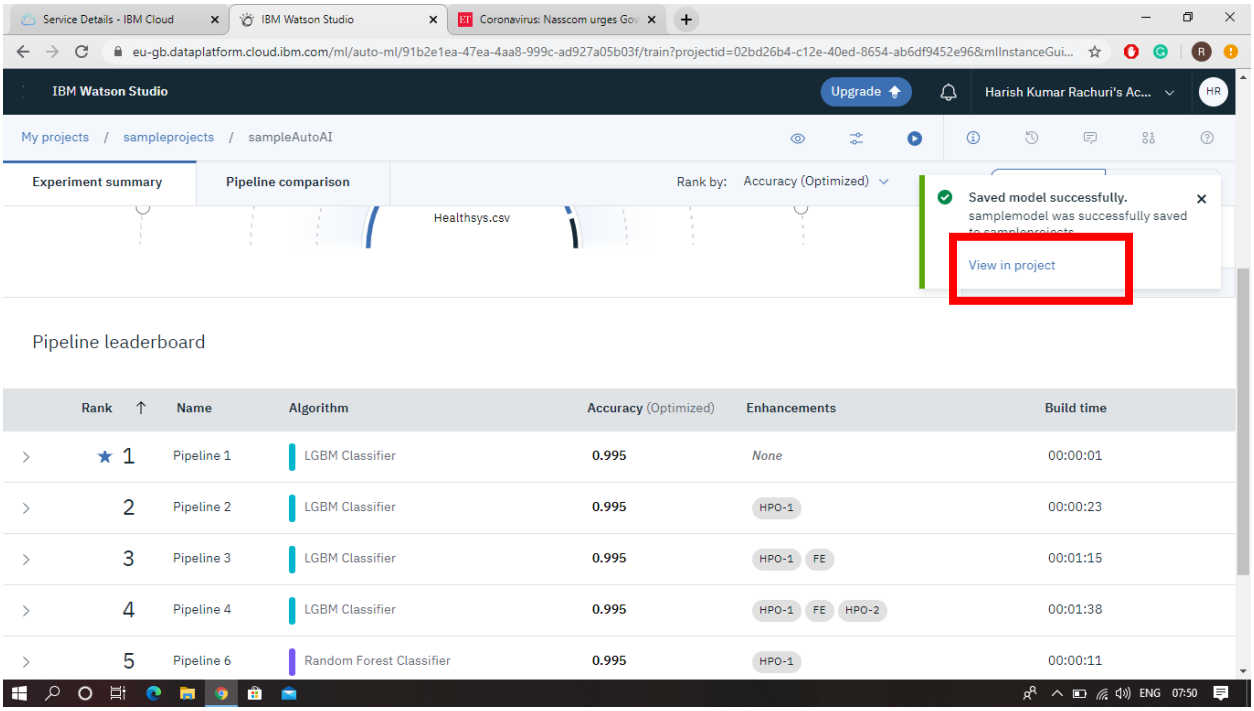
- Wait till the progress map complete its operations with different alogrithms to show the best suited alogrithm for Our Data set.



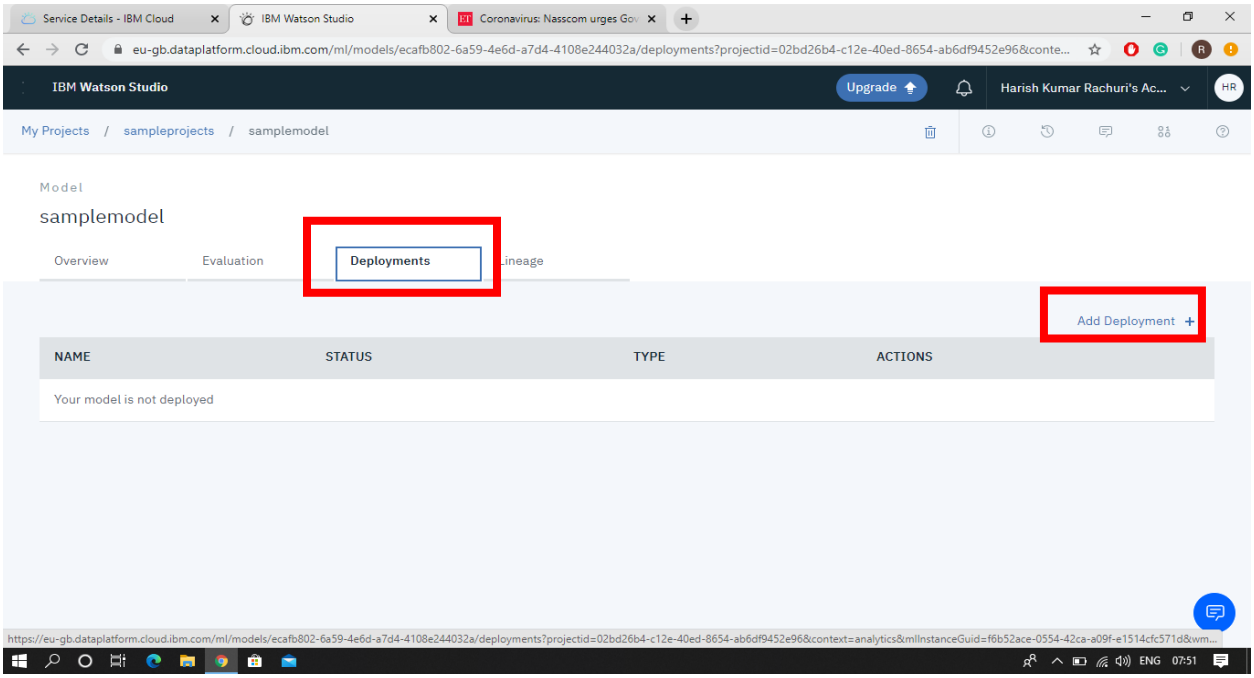
- We can see different Ranks for Different Algorithm's. We need to select the Best algorithm and save the model by clicking the save as option.
- You need to save it as model.



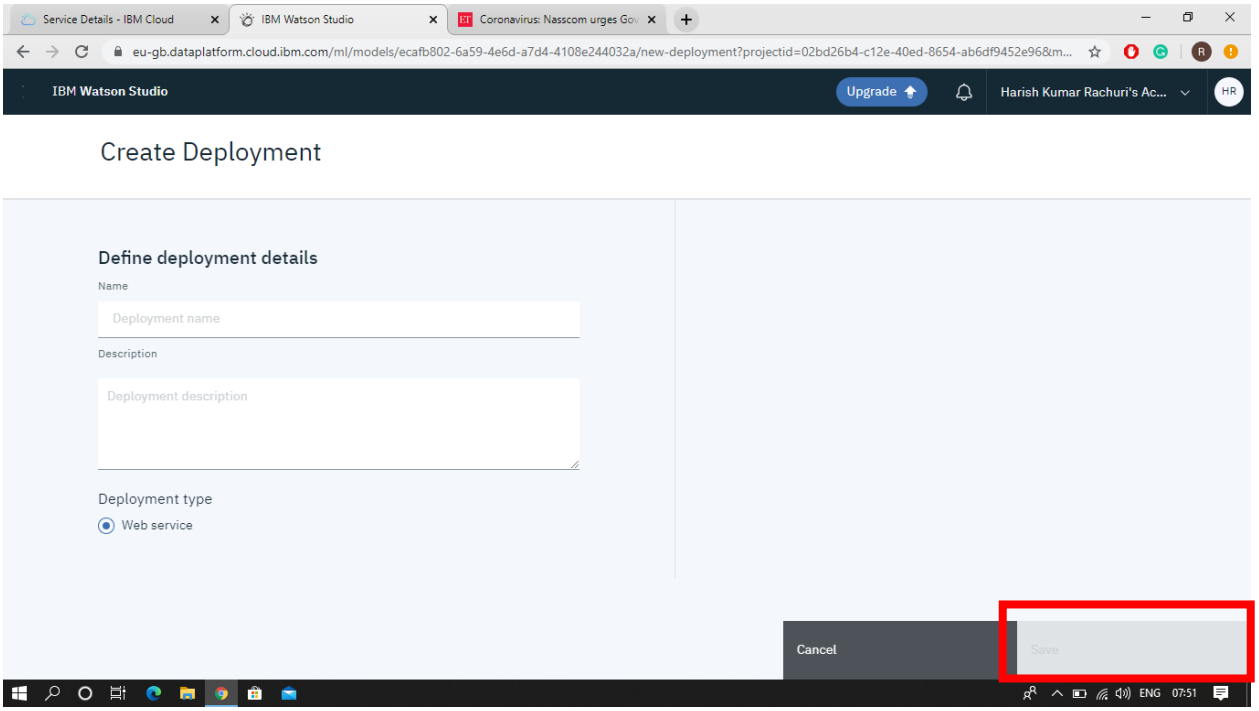
- Wait till model is successfully saved. You will get the notification once it is saved. Now you click on view in projects



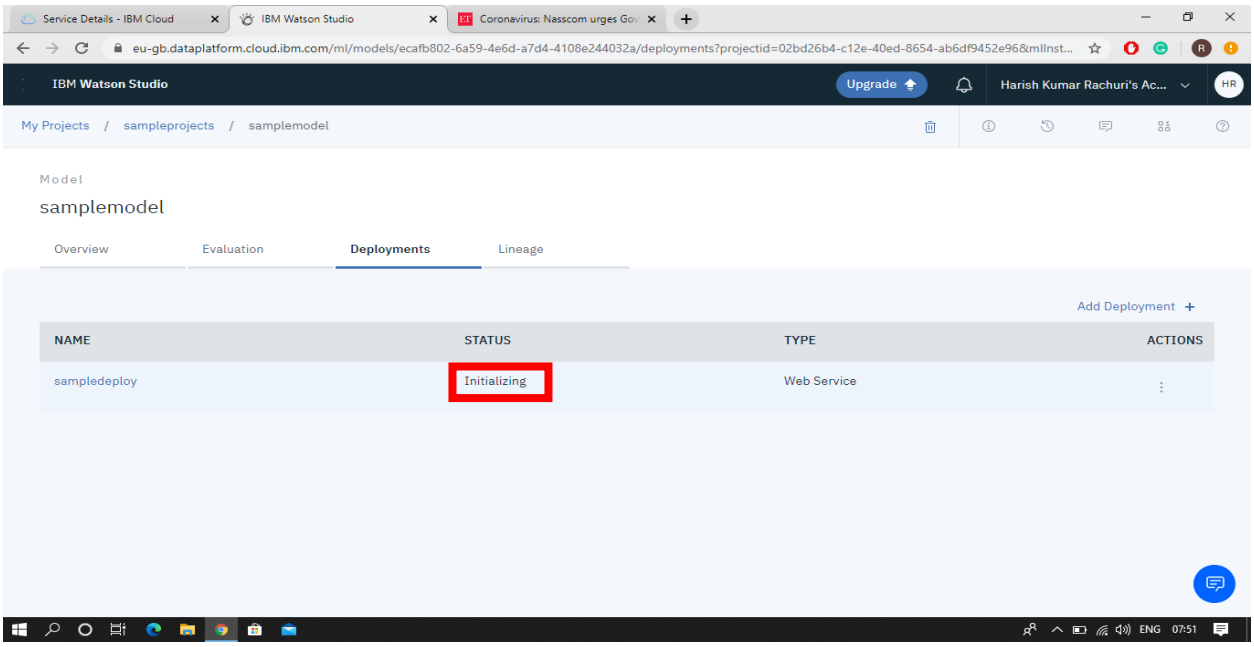
- It will redirect to page where we can find different options like Overview, Evaluation, Deployment and lineage.
- Now we will deploy our model. Select the deployments option. And then click on Add Deployment.



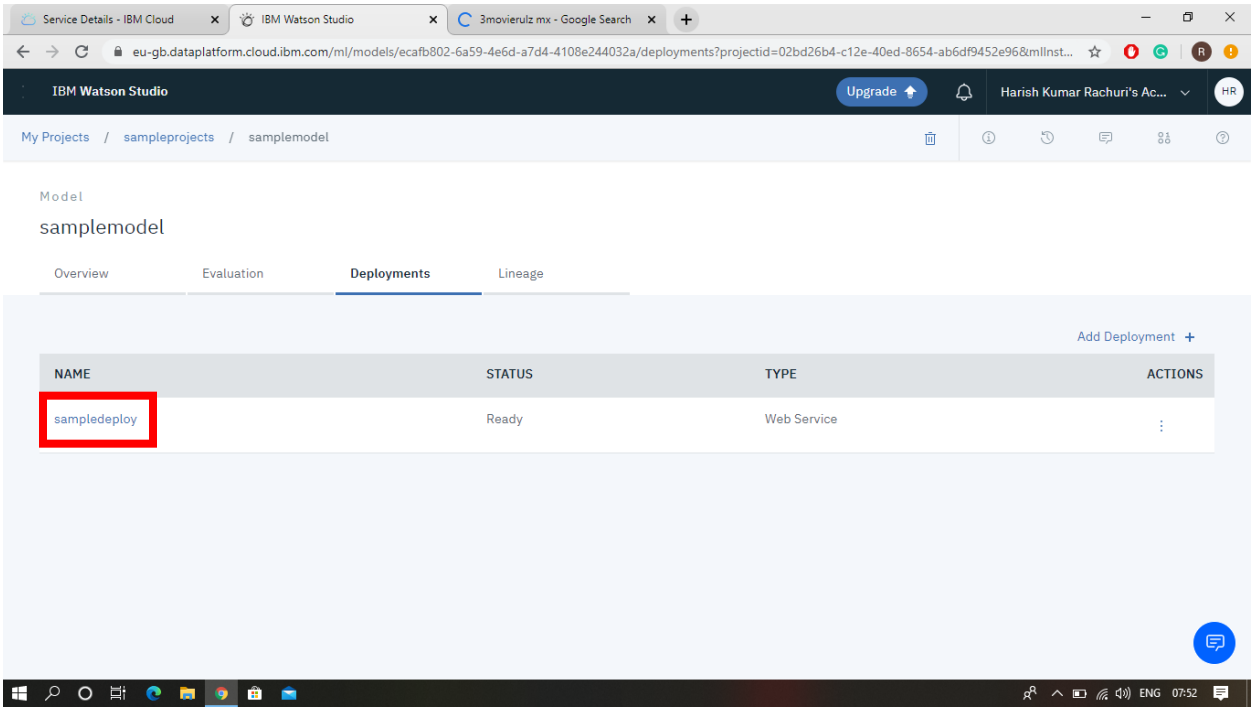
- After clicking on Add Deployment it will load page to fill the deployment Details.
- Give some name to your deployment and select type as web service and click on Save.



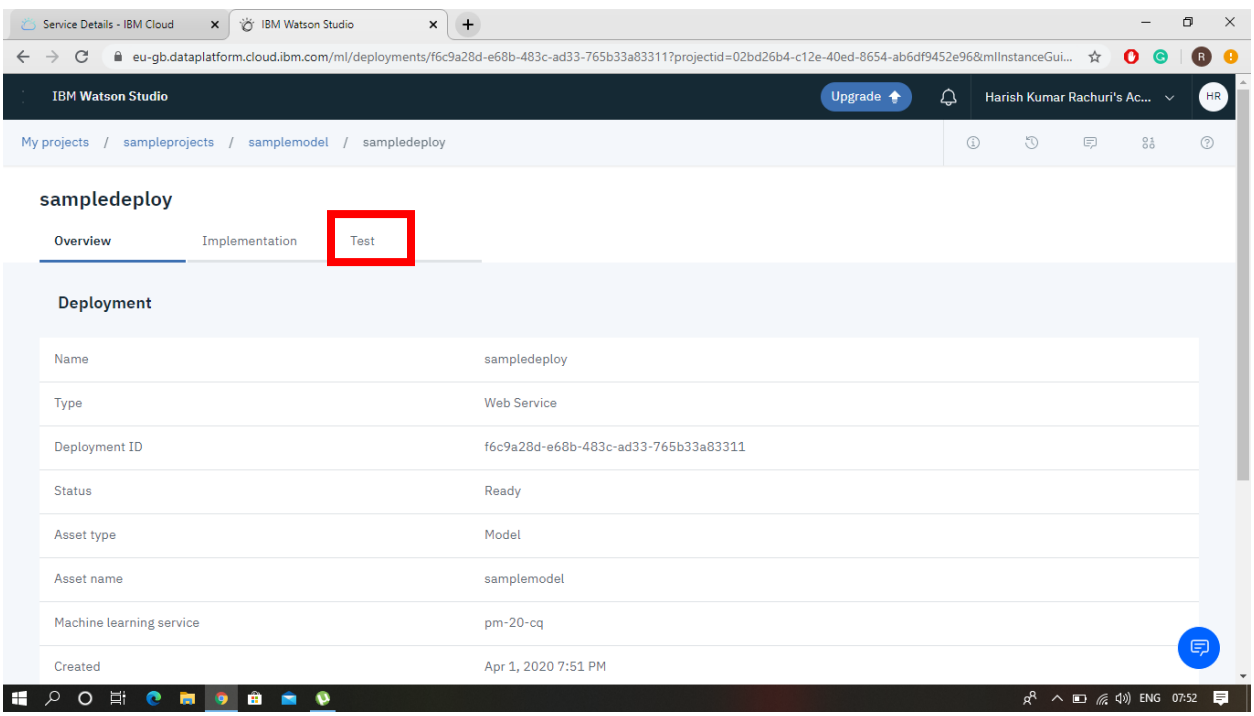
- Wait till Status changes from Initializing to Ready



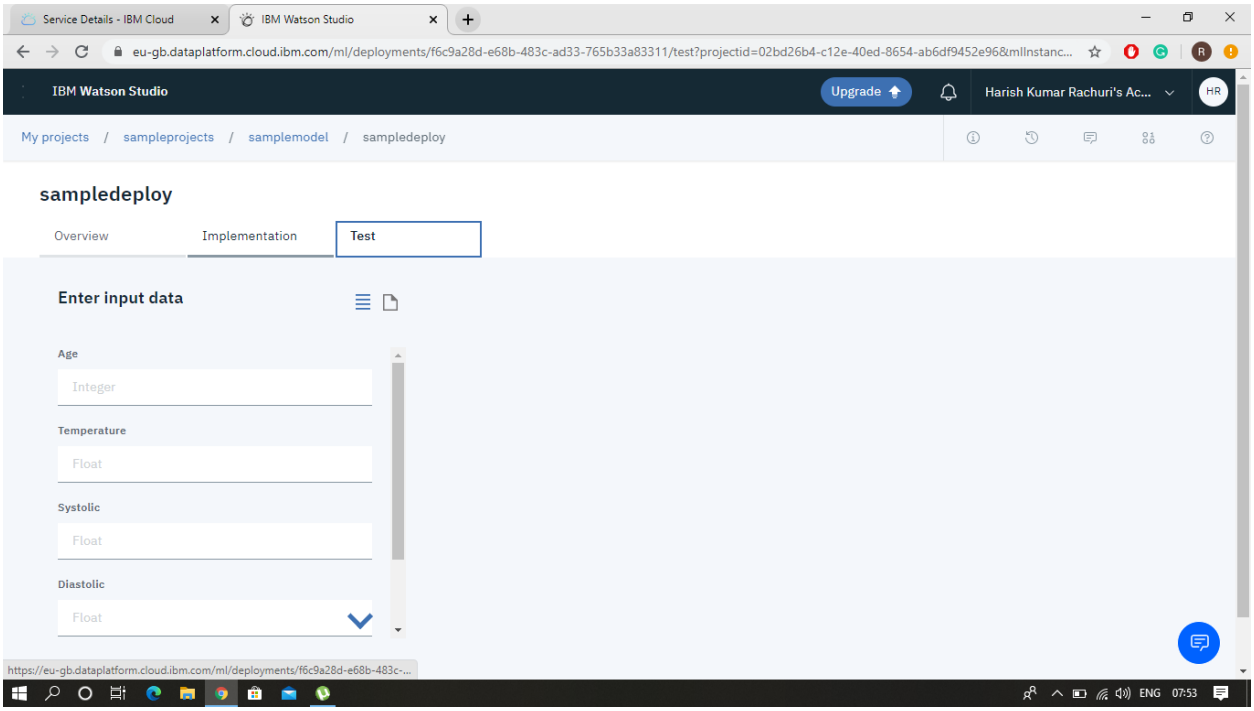
- Once you get the status as ready it means that your deployment is done successfully.
- Now you click on the deployment name.



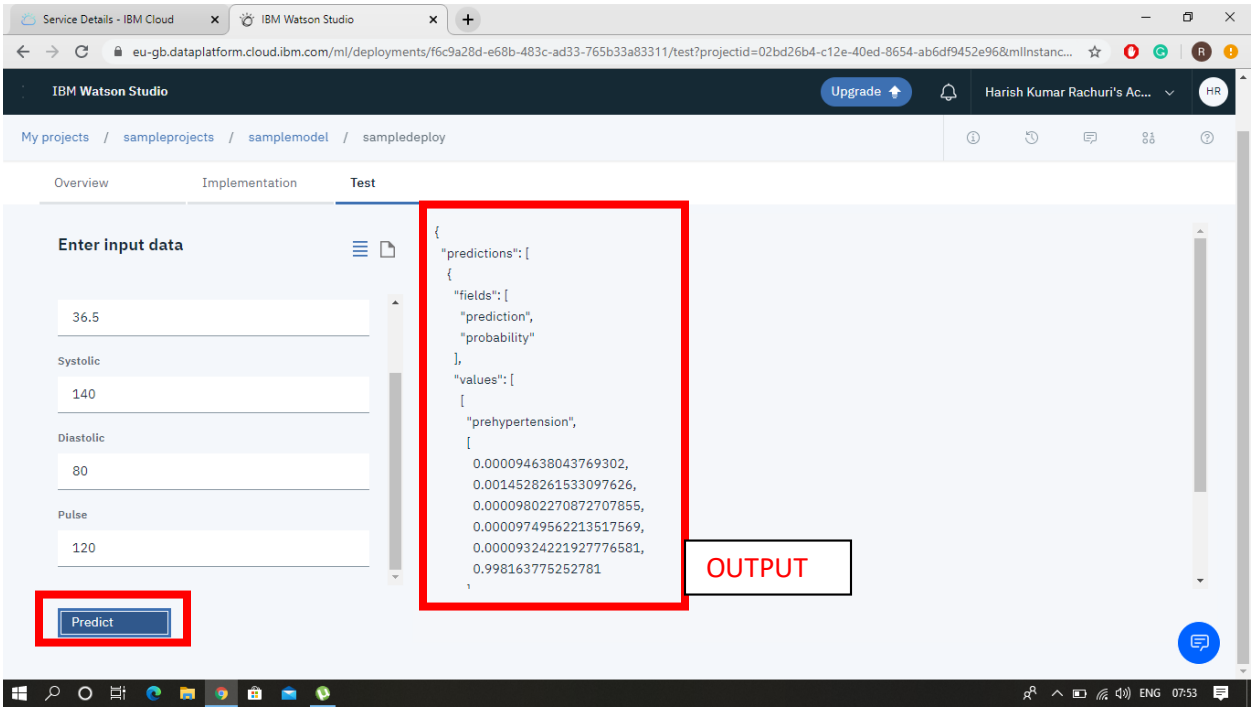
- After Clicking the Model Name, you will get Options like Overview, Implementation, and Test.
- First we will test our model so click on test.



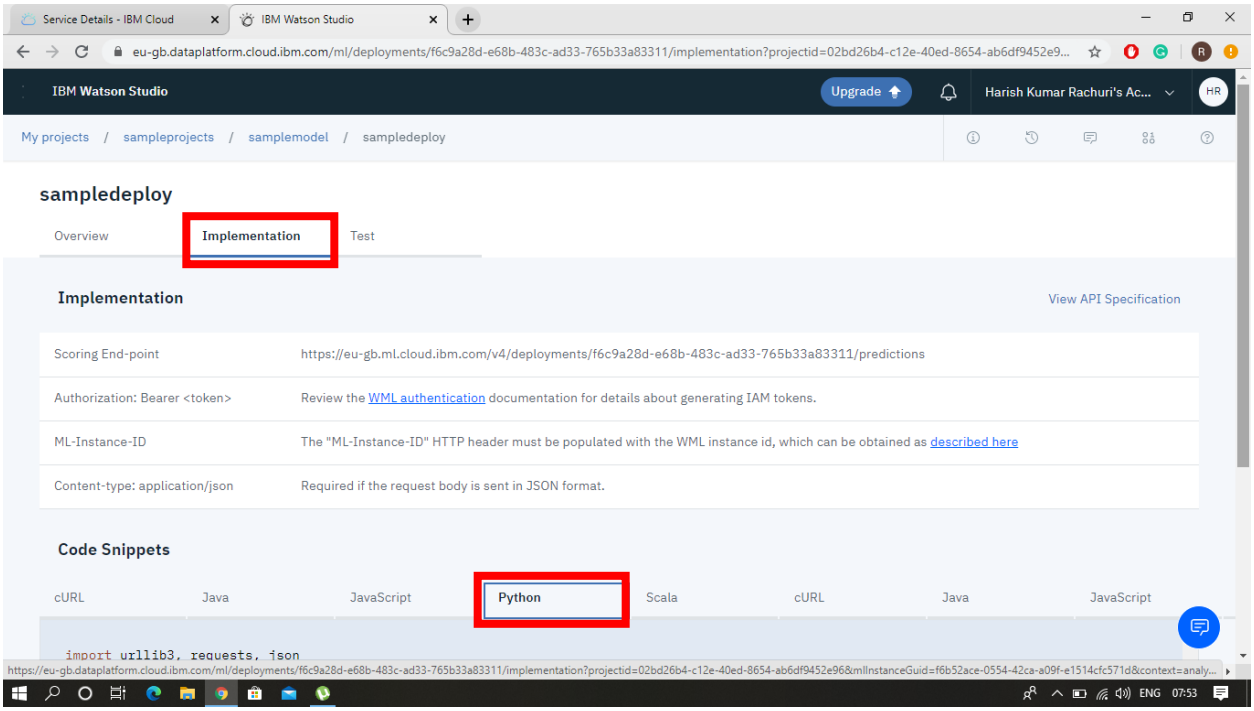
- In this window we need to give the input parameters to predict the output.



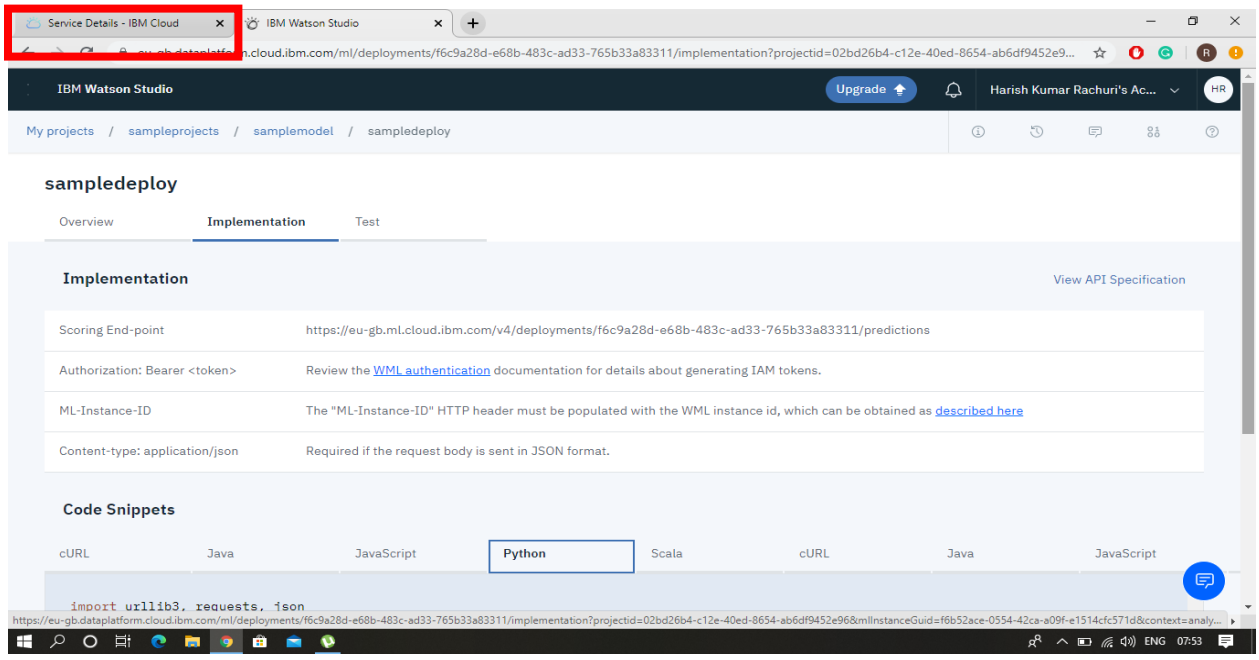
- After providing the input parameters Click on the predict button to check the output.



- If you got your prediction correctly. We can go for implementation. Click on implementation. And select python.

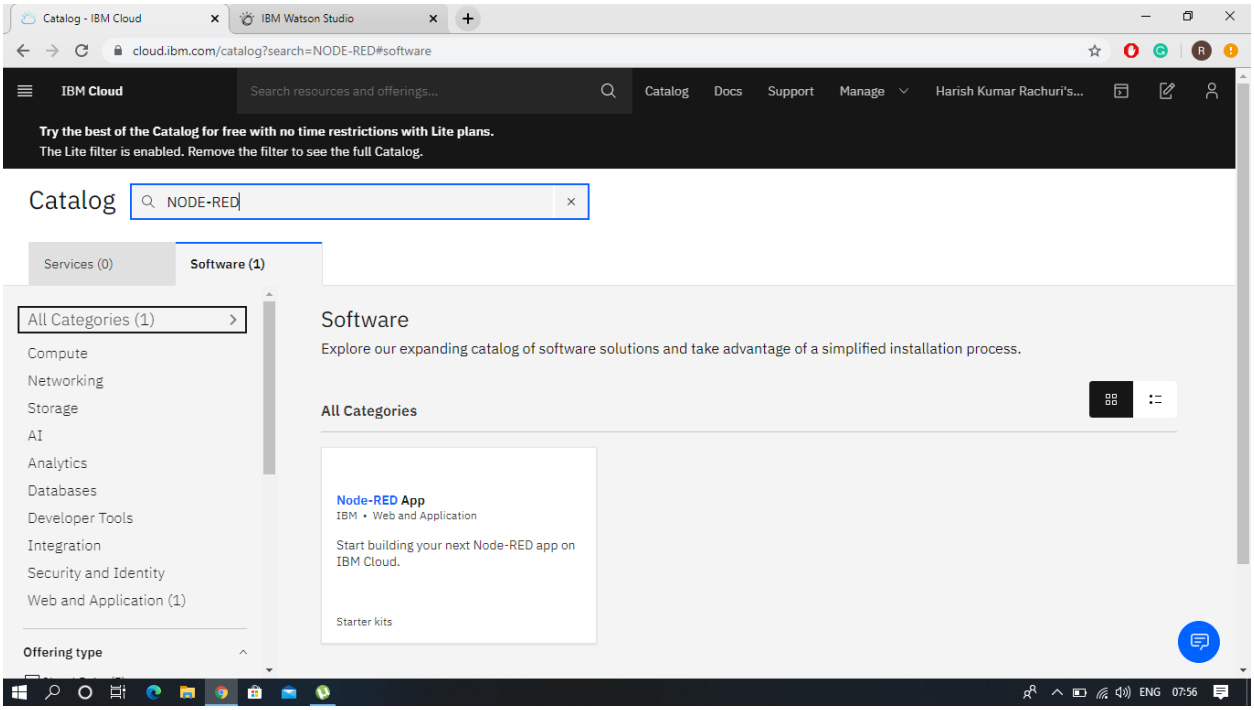


- Till now we have trained machine learning Model and tested it. Now, it's time to integrate the model with our IOT Platform through which you are getting your sensor data and design of web User Interface.
- Switch to "service Detail" tab to create the Node RED Application. Or simply you can open a new tab and open your IBM Dashboard.

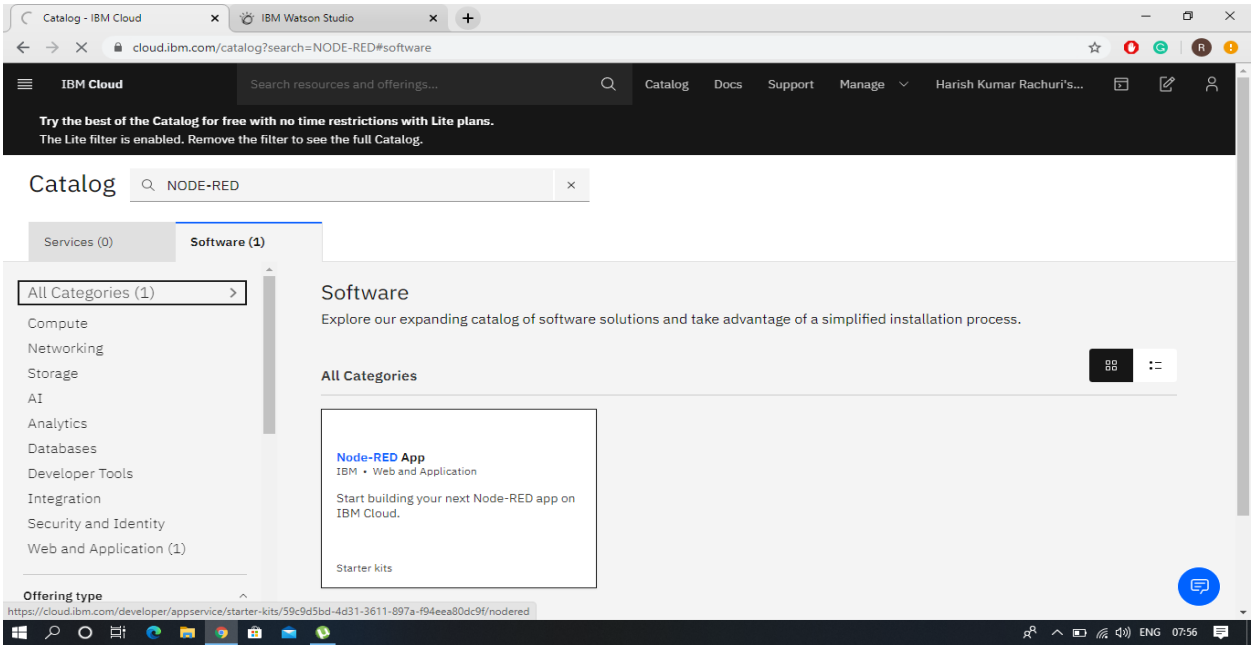


Task 5: Configure the Node Red Application to get the sensor data from IOT platform

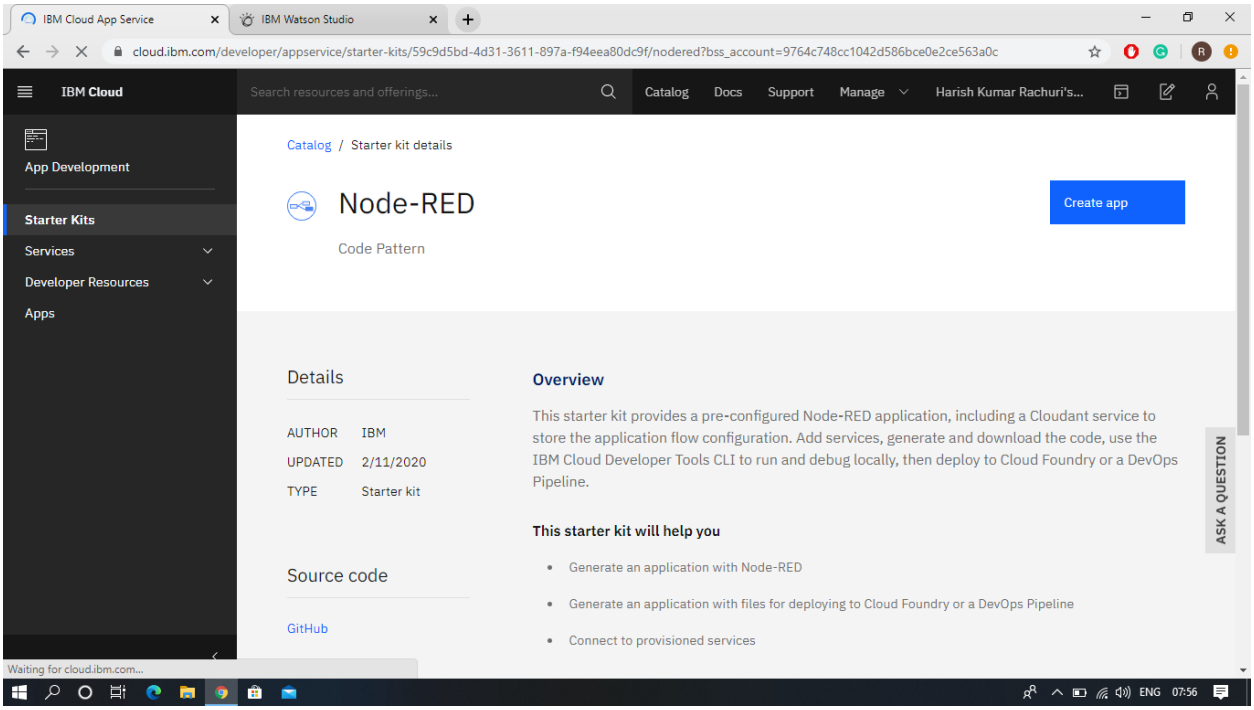
- in the dashboard, Click on the catalog.



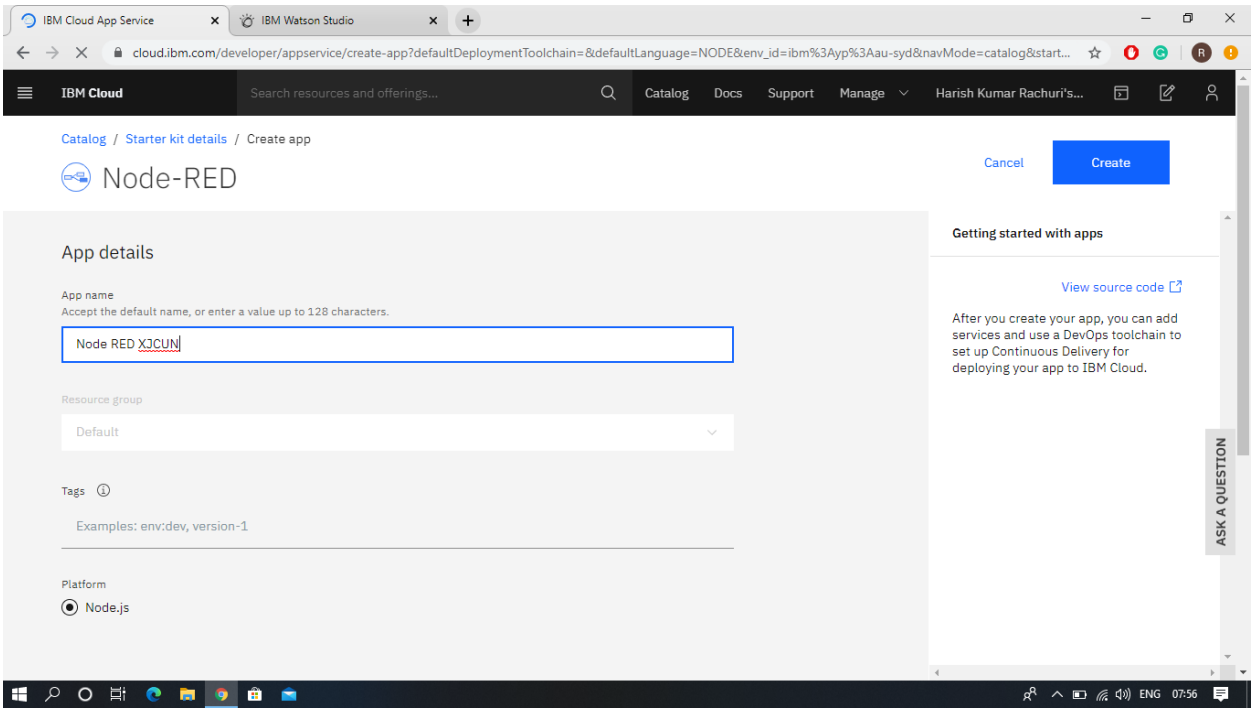
- Search for node-red in the search bar provided. You will get Node-Red App, click on that.



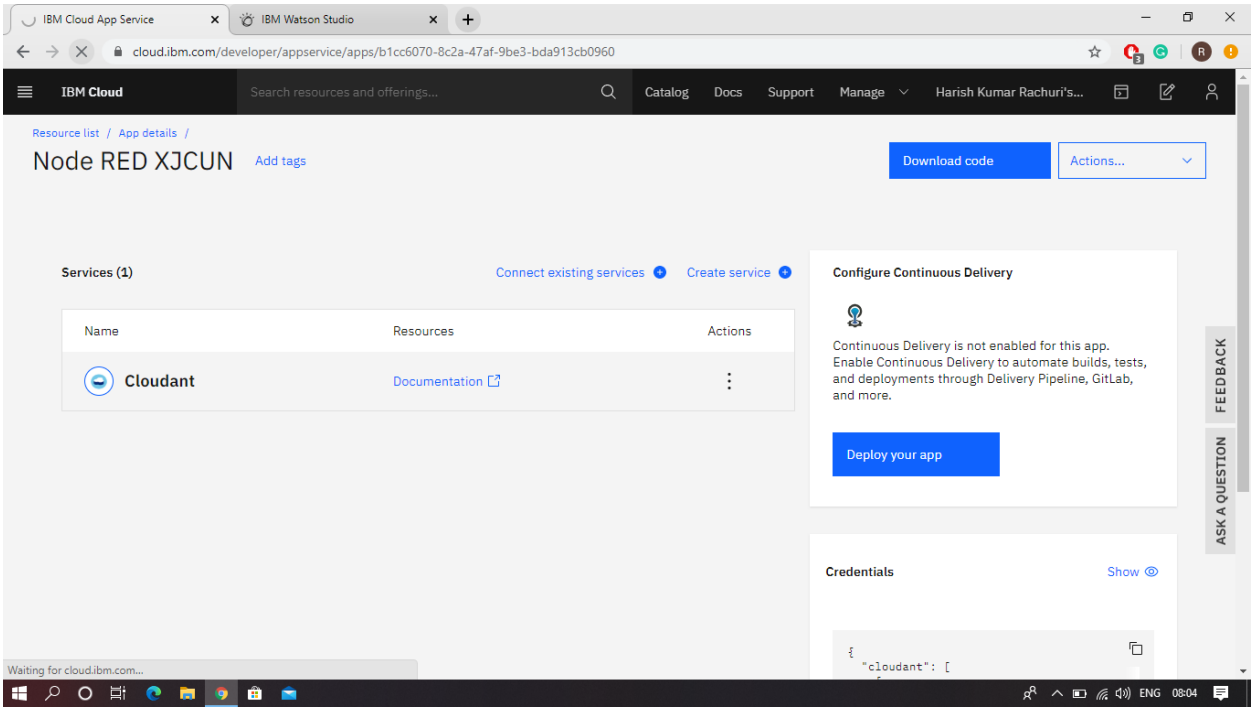
- After clicking on the node-red app you will be redirected to the node-red application starter kit page. Click on create app.



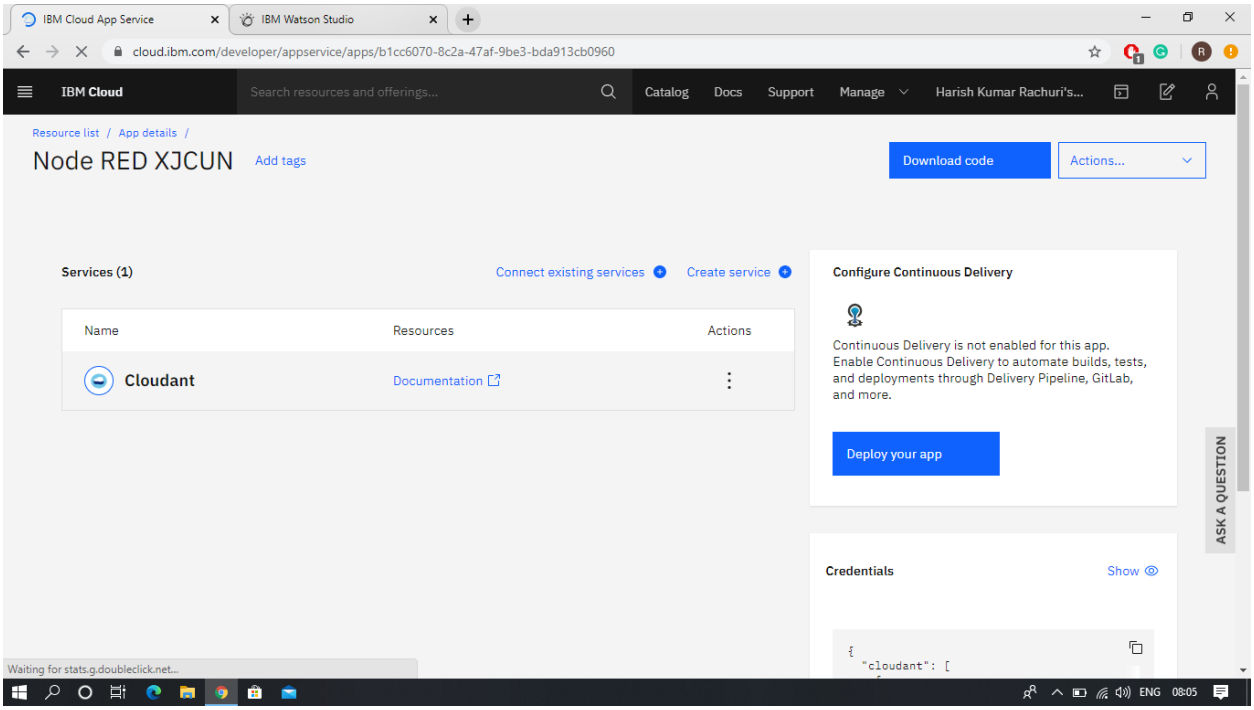
- To create the node-red application provide the App name. The name should be unique. One may use the default name provided or change it. Then click on create.



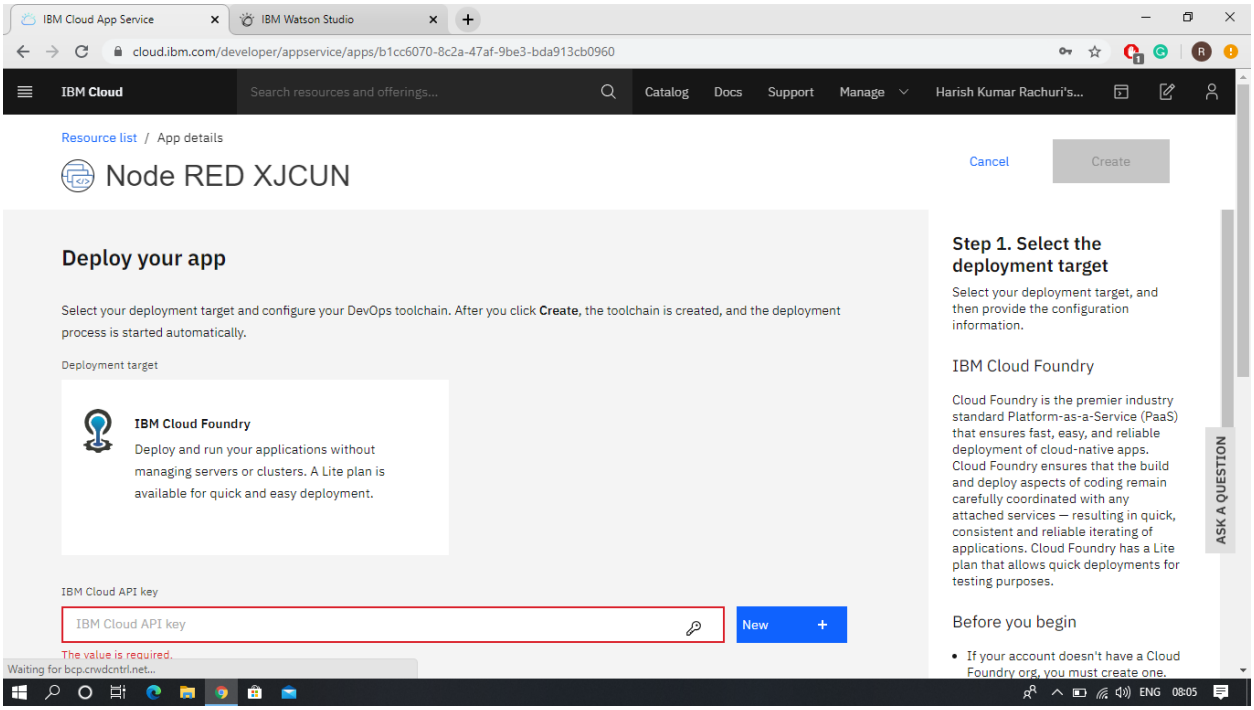
- To create the app, it will take some time. meanwhile the cloudant service will be created.



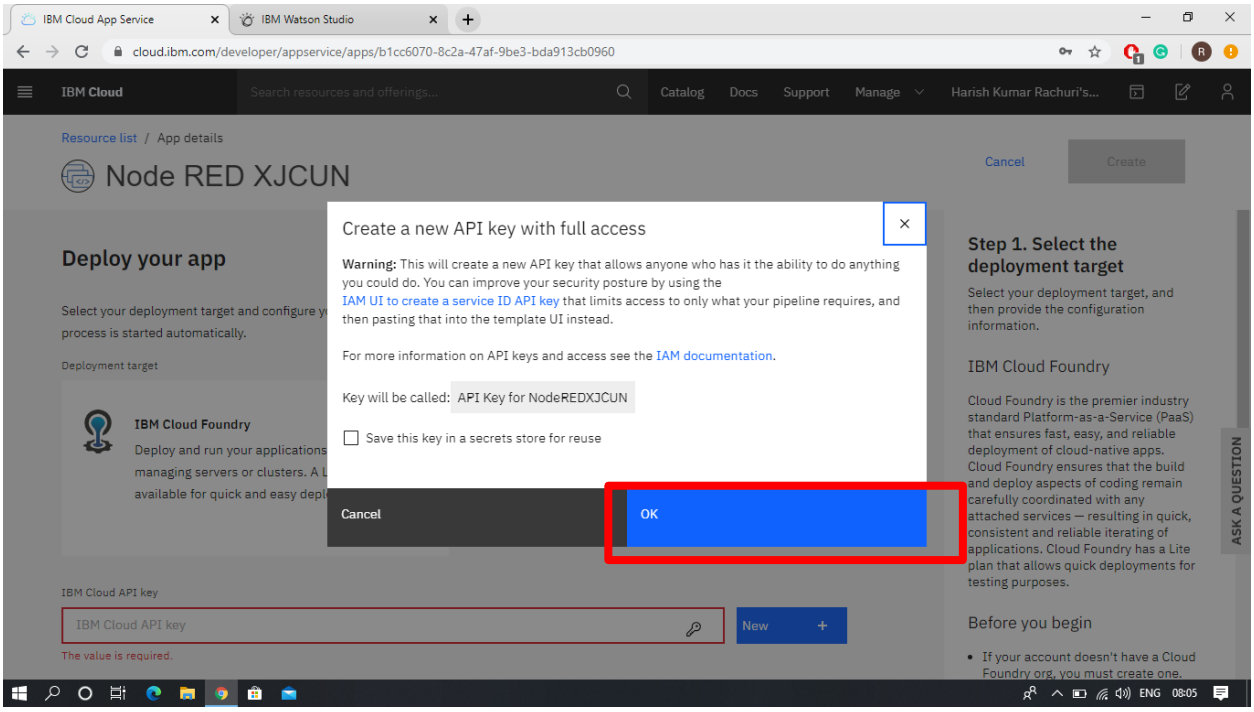
- After the successful creation of cloudant service, click on deploy your app button



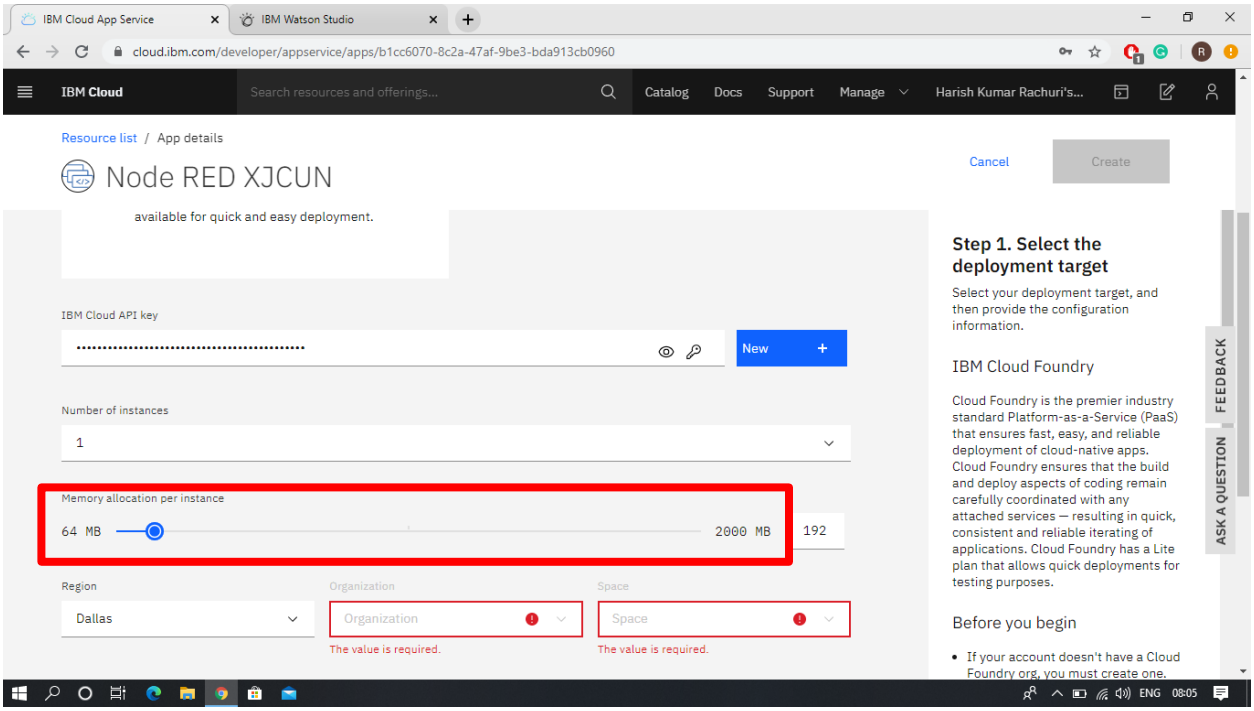
- In the process of deploying you will be prompted for the IBM cloud API key. Click on new to generate the one.



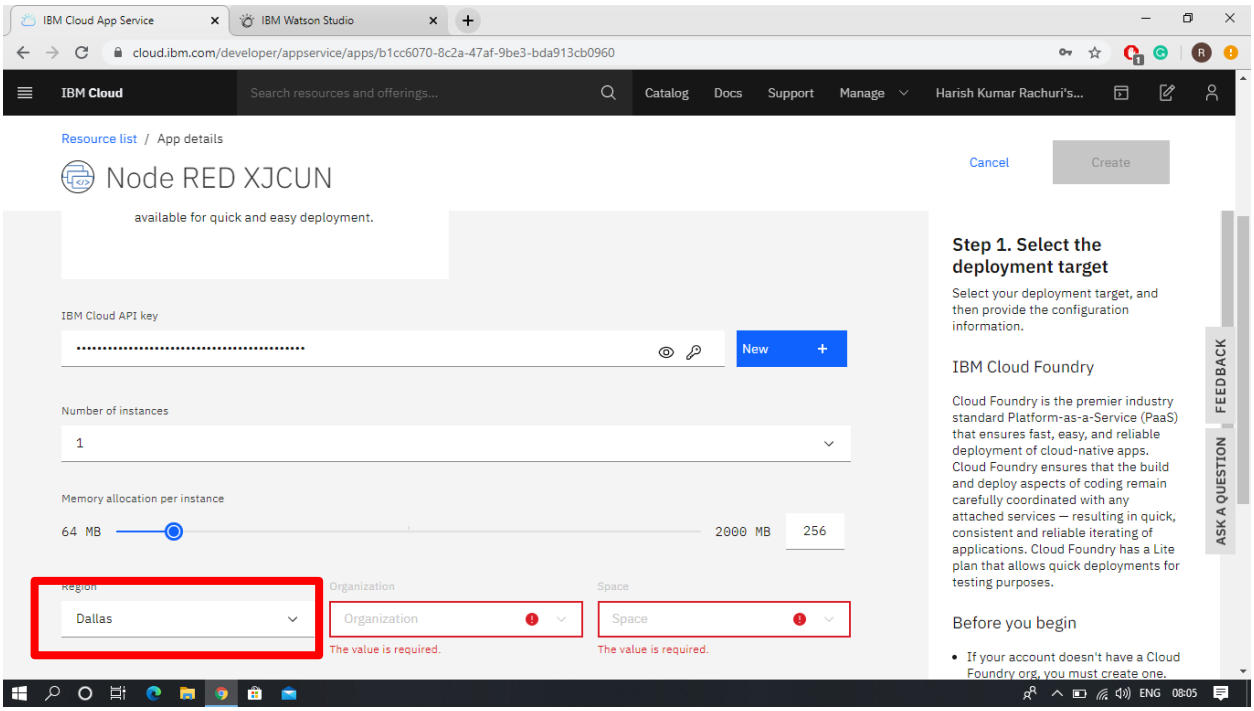
- At the time of API key generation, you will get a popup there you click on ok



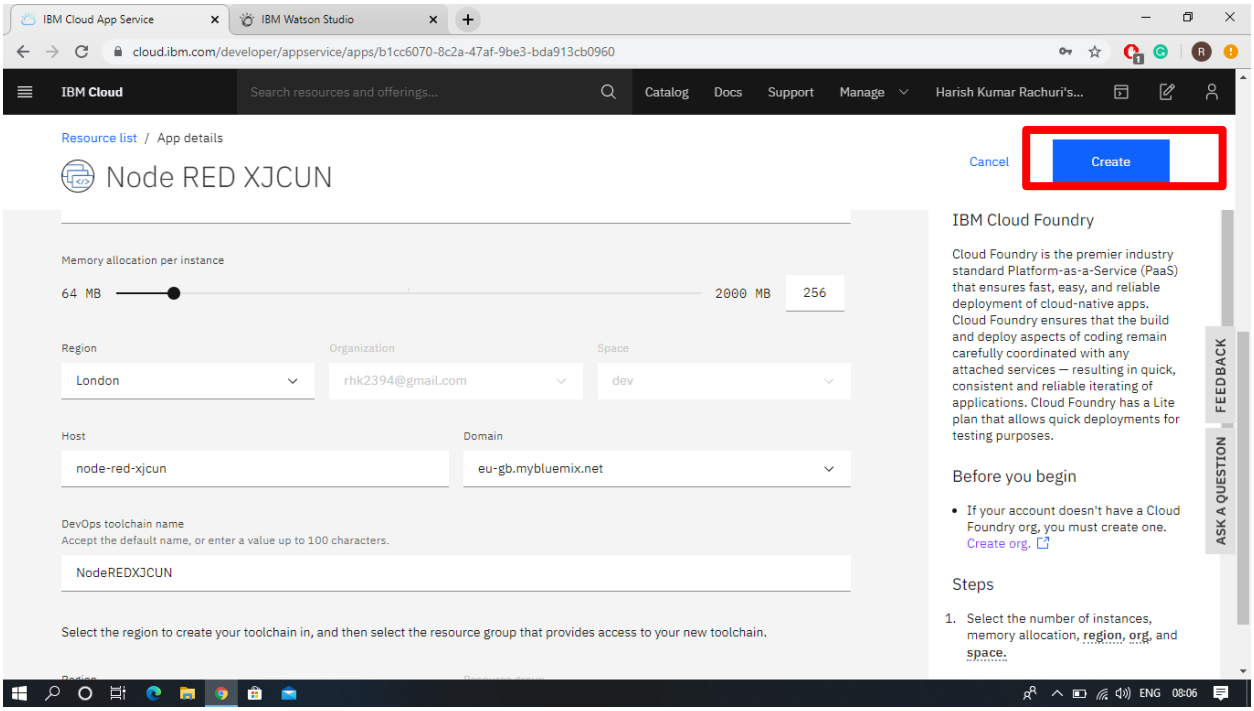
- Increase your memory allocation per instance to 256.



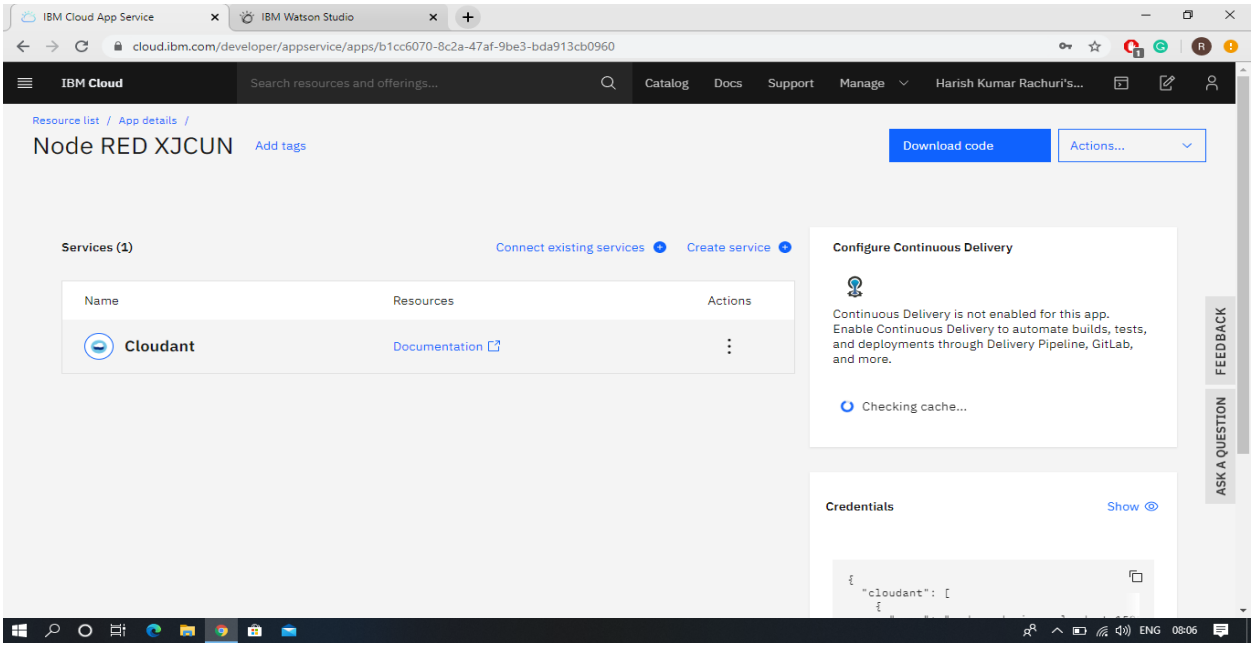
- Select the Region for getting the organization and space



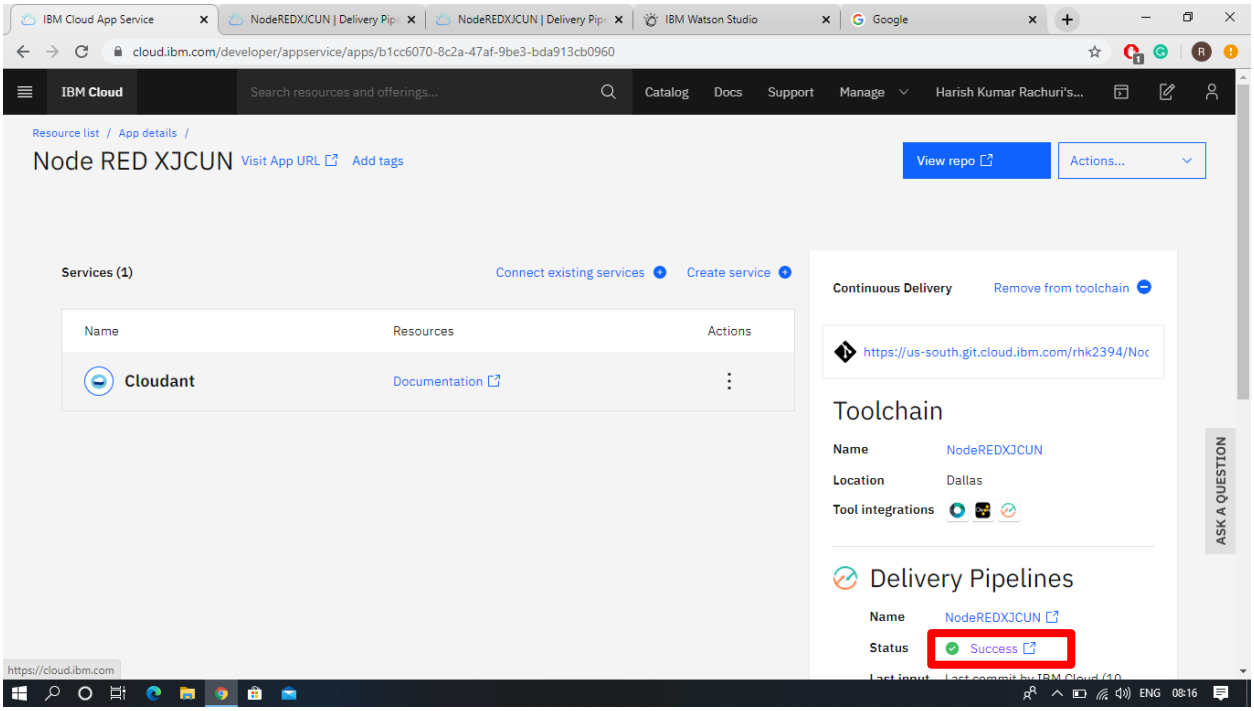
- Once we get the Organization Space and Host, click on create.



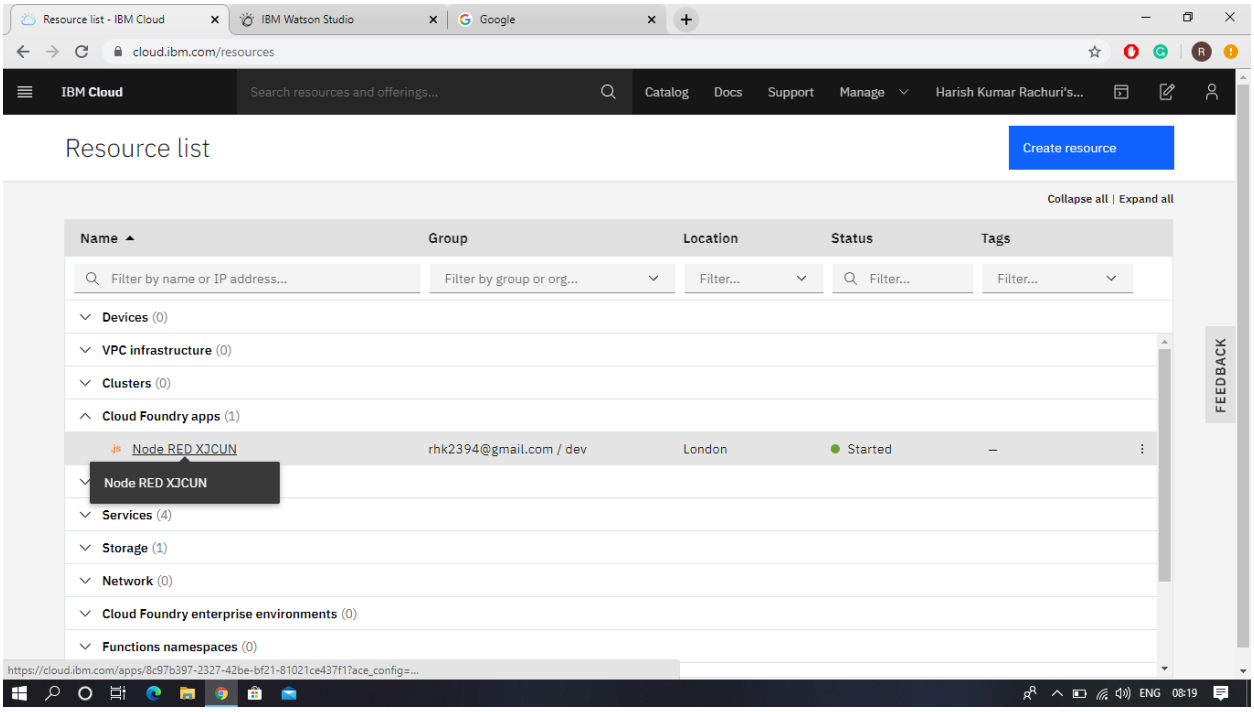
- Upon successful creation of the node-red app, we can see Status in Configure Continuous Delivery.



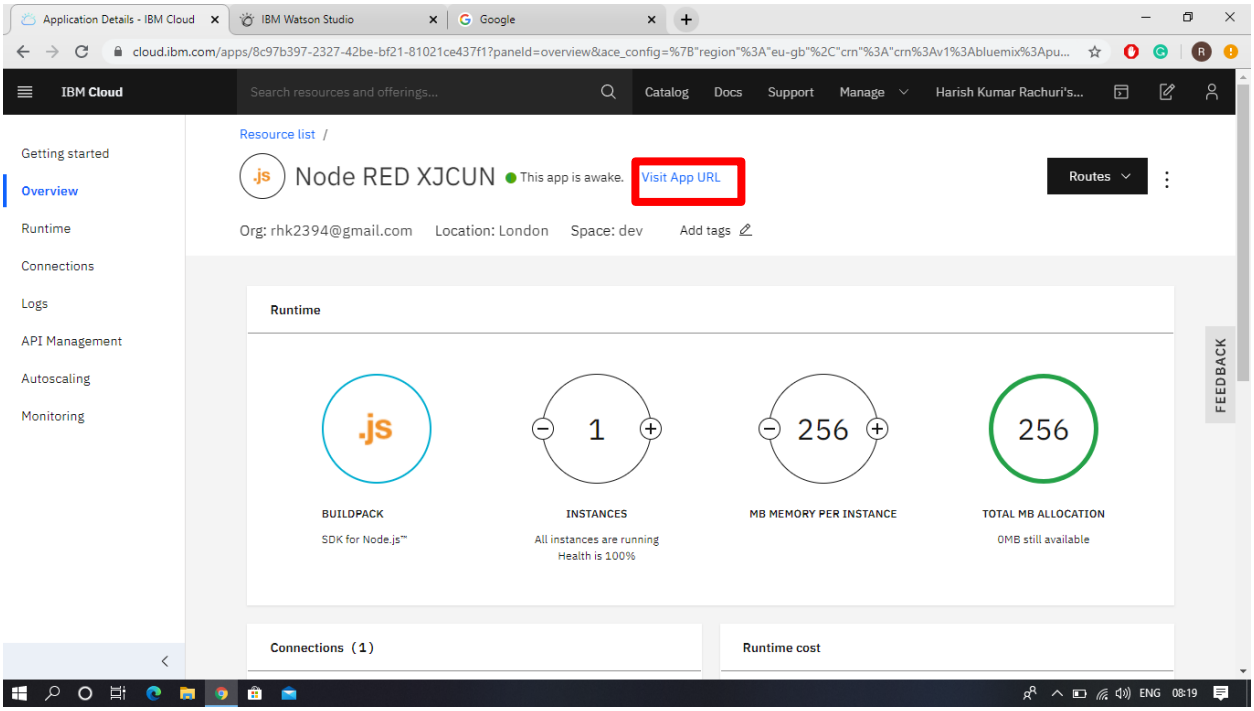
- After successful creation of tool chain we can check the status, wait till it changes to Success.



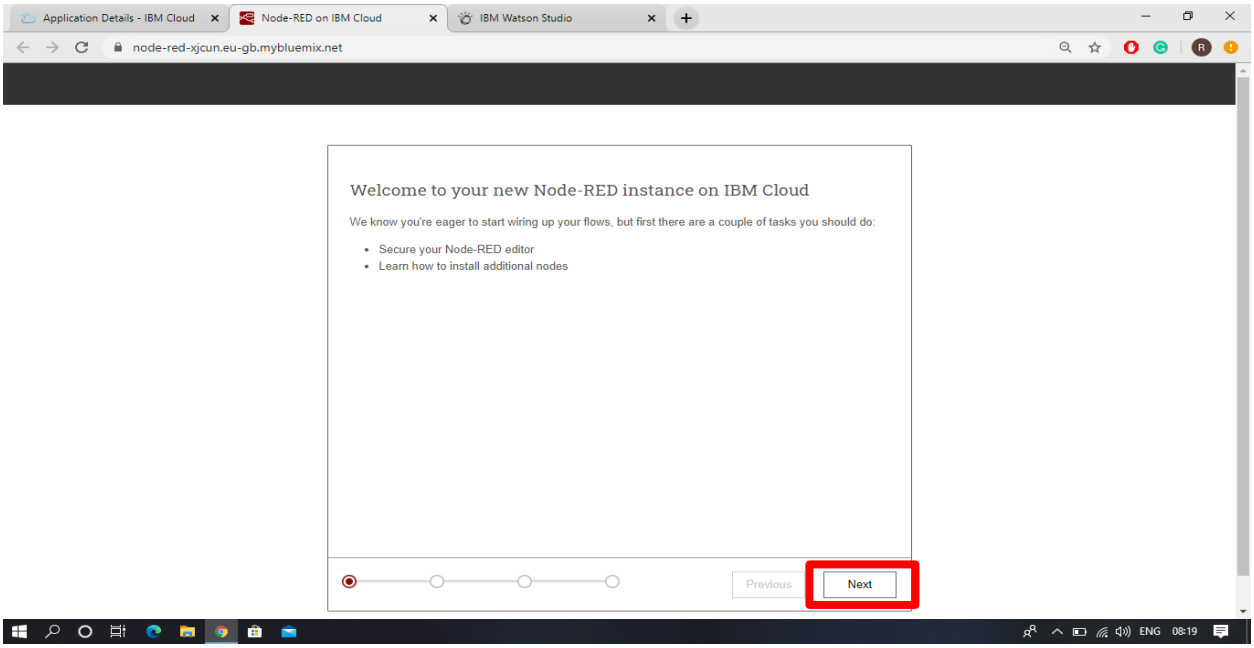
- To launch the node-red application click on cloud foundry apps in the dashboard. You will find your service name which is already created. Click on the service to launch



- Click on visit app URL to launch the app

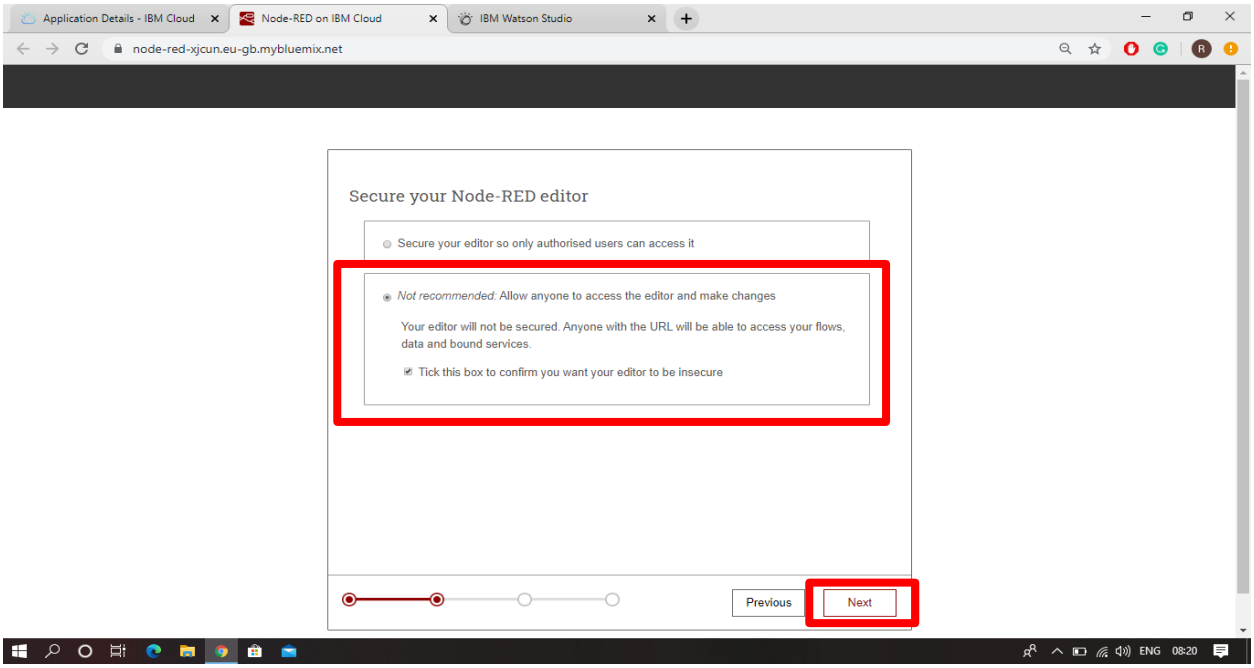


- Upon launching the app, you are redirected to the following page. Click on next

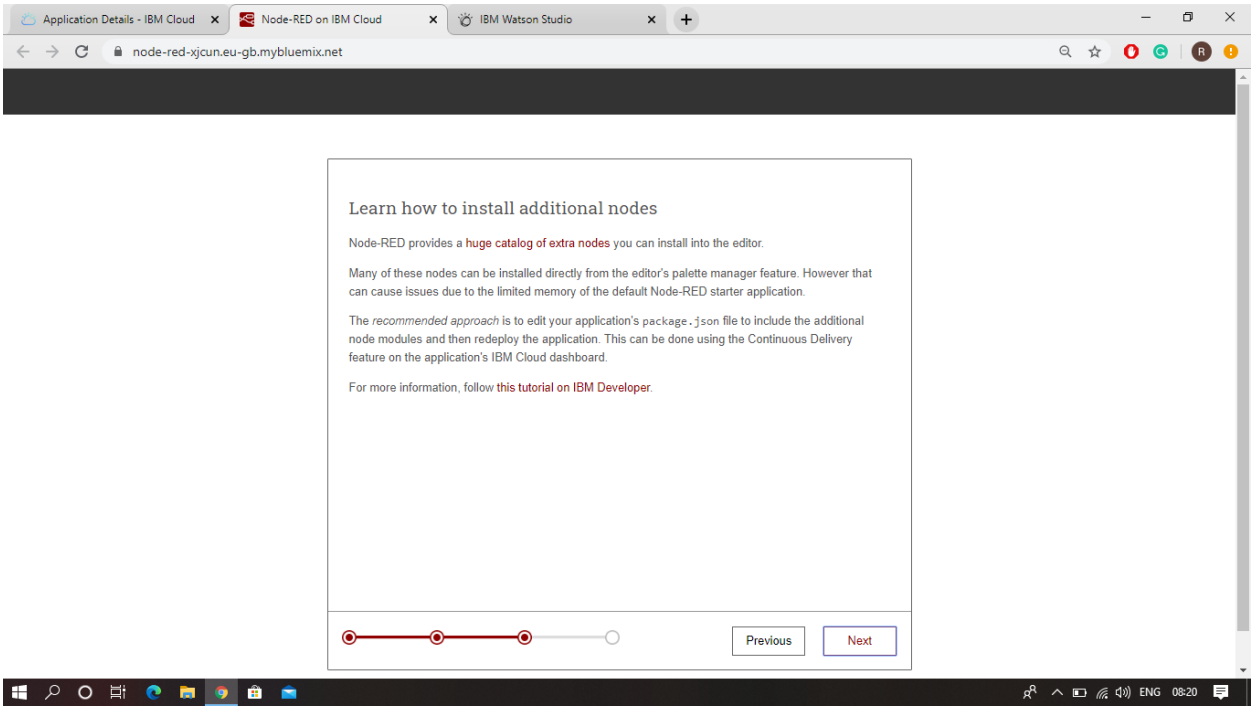


- In the next step, you will be asked for securing the node-red app, if you want you can provide the credentials as your wish but remember those credentials. We can't reset or retrieve them back. So better avoid giving credentials.

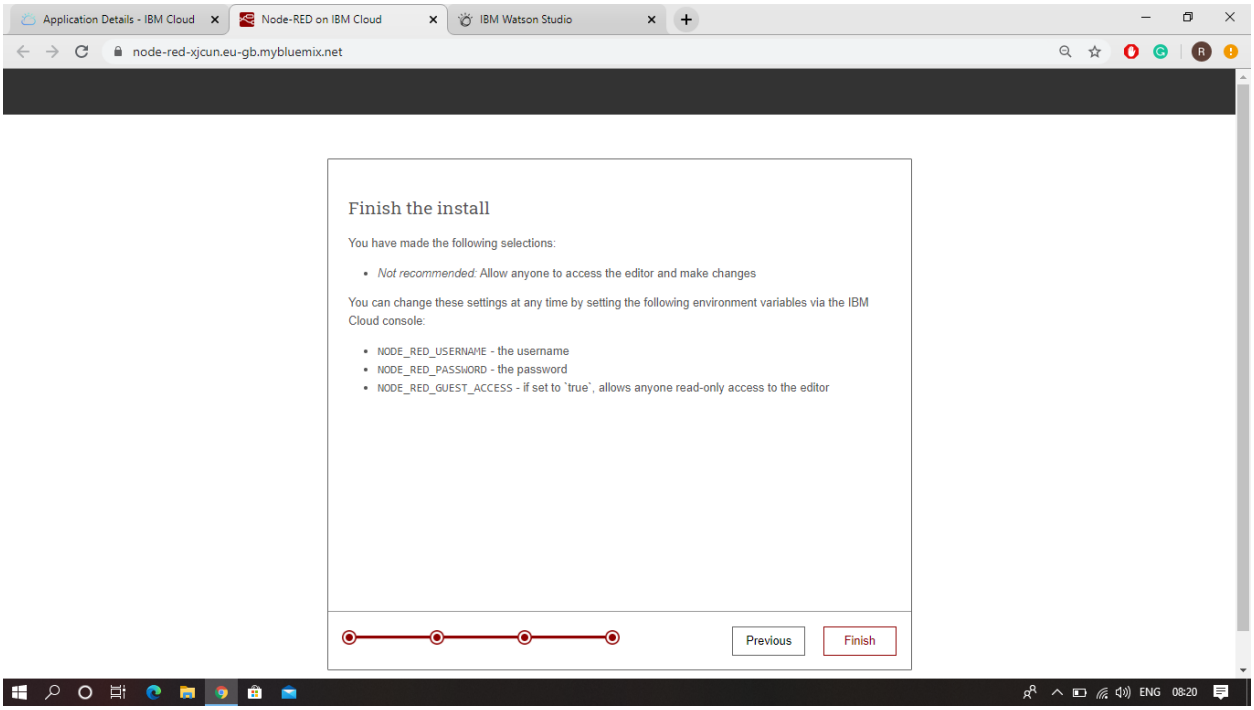
- click the two checkboxes as shown in the below image. This step is for not securing the node-red editor.



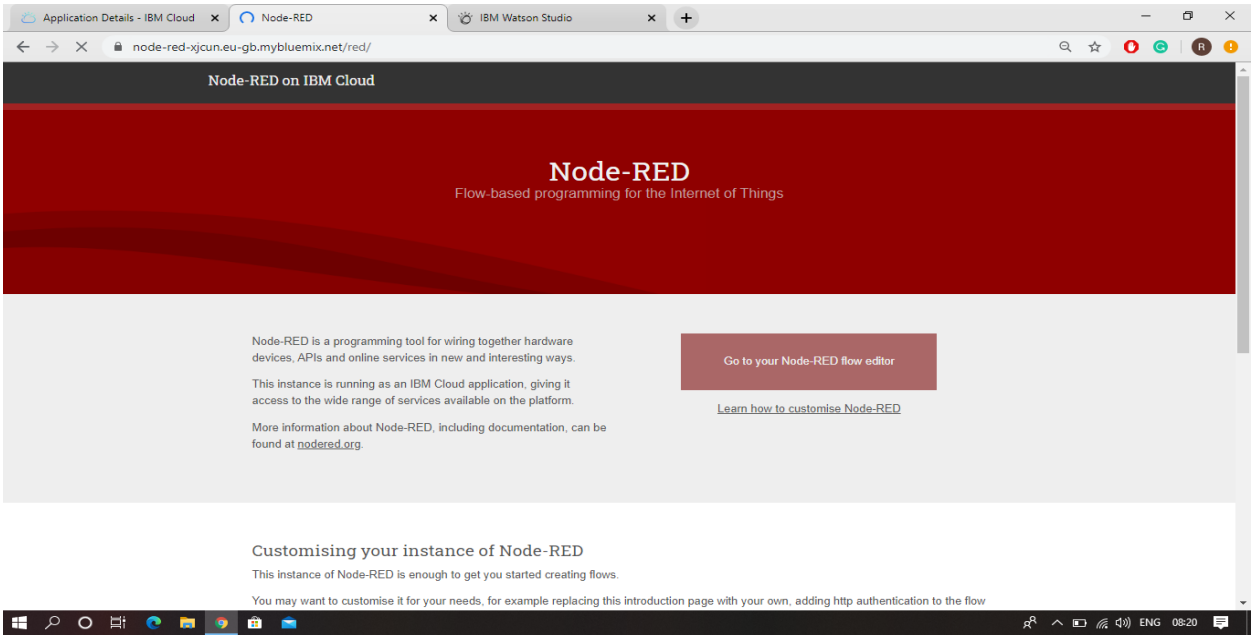
- Click Next.



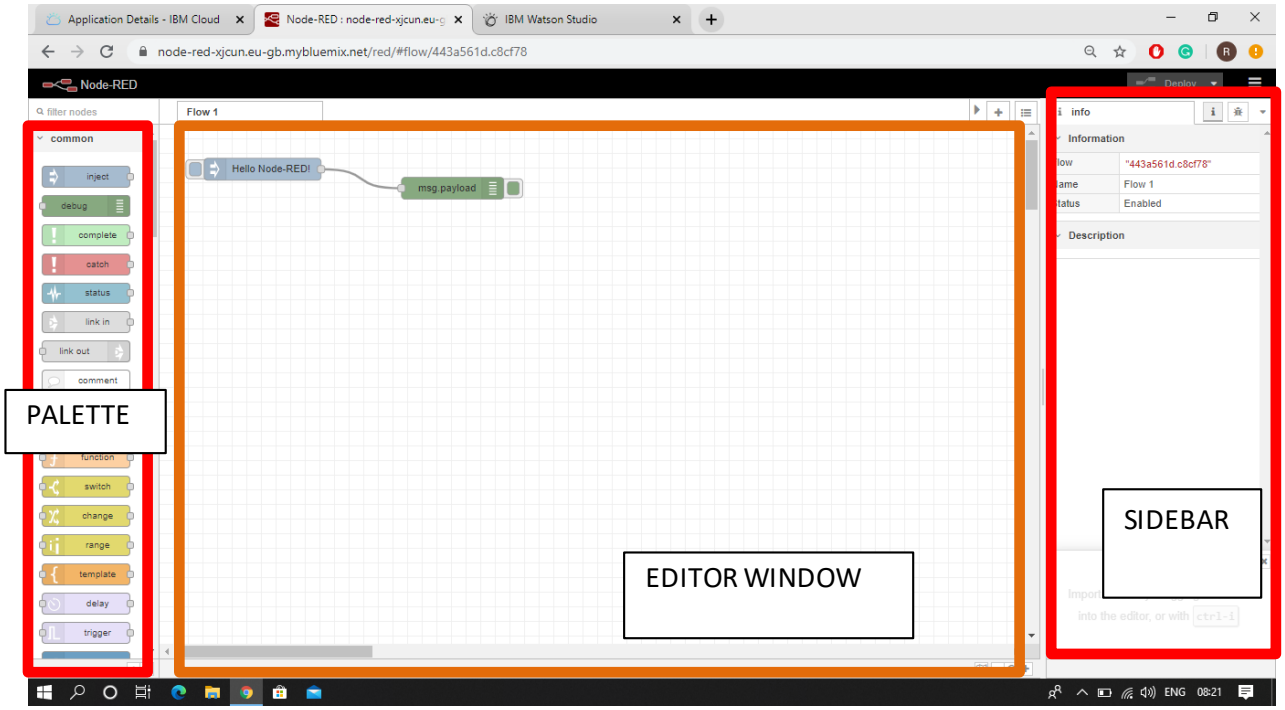
- This procedure is done only for launching the service for the first time. For a returning user, you can directly browse from the cloud foundry apps in the dashboard. And click on the visit app URL after getting the app is awake/running.



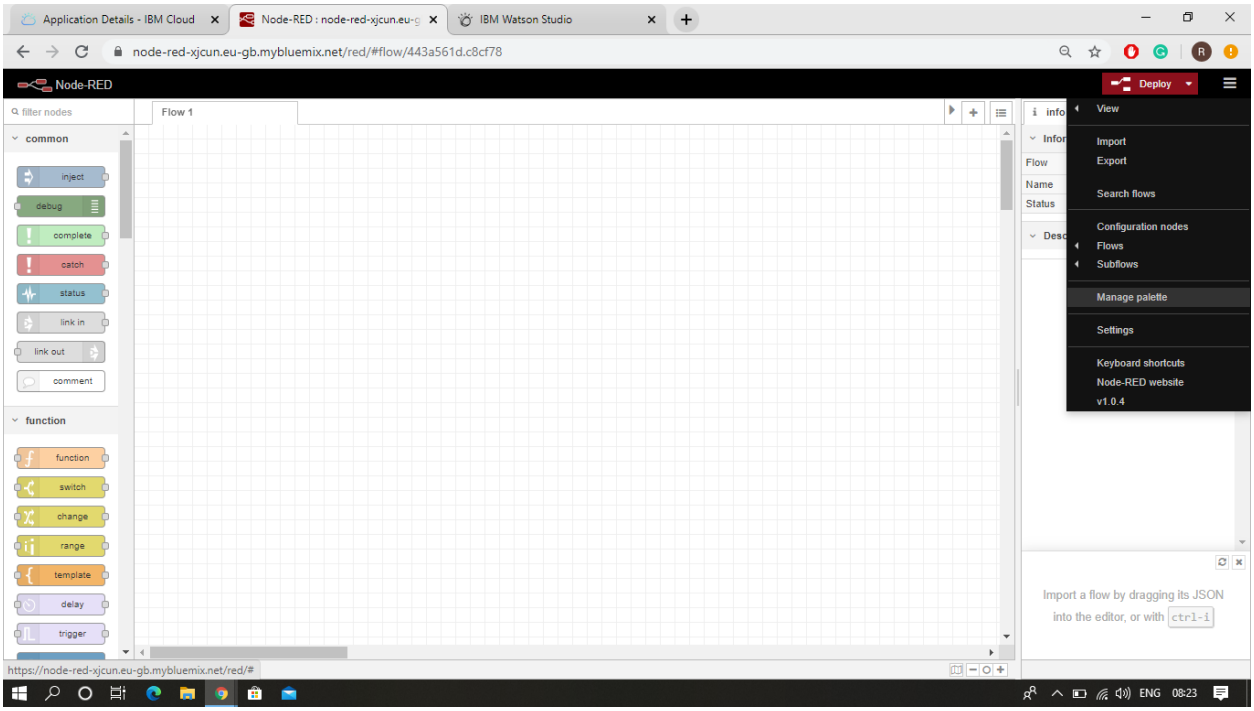
- After completing the procedure of launching node-red you are asked to go to the editor. Click on go to the editor button to launch the application



- On launching your node-red application you will be on the editor pane where one can drag and drop the nodes.

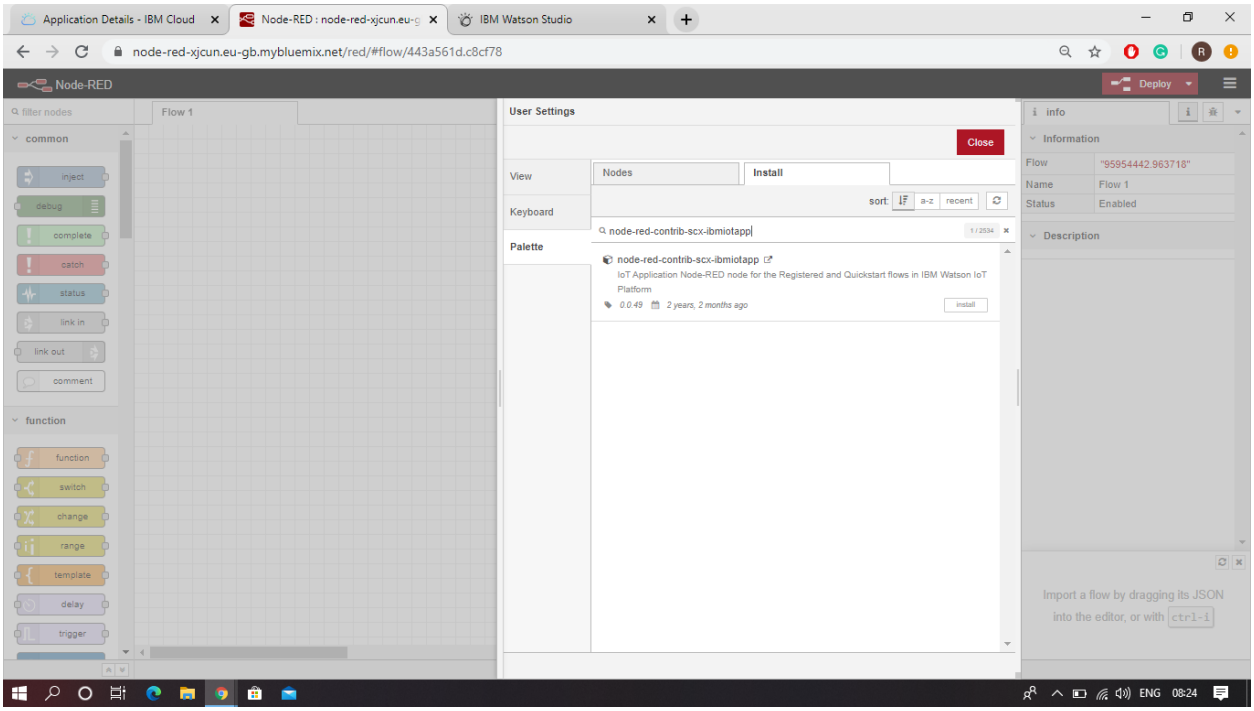


- Navigate to manage palette for installing the nodes required.

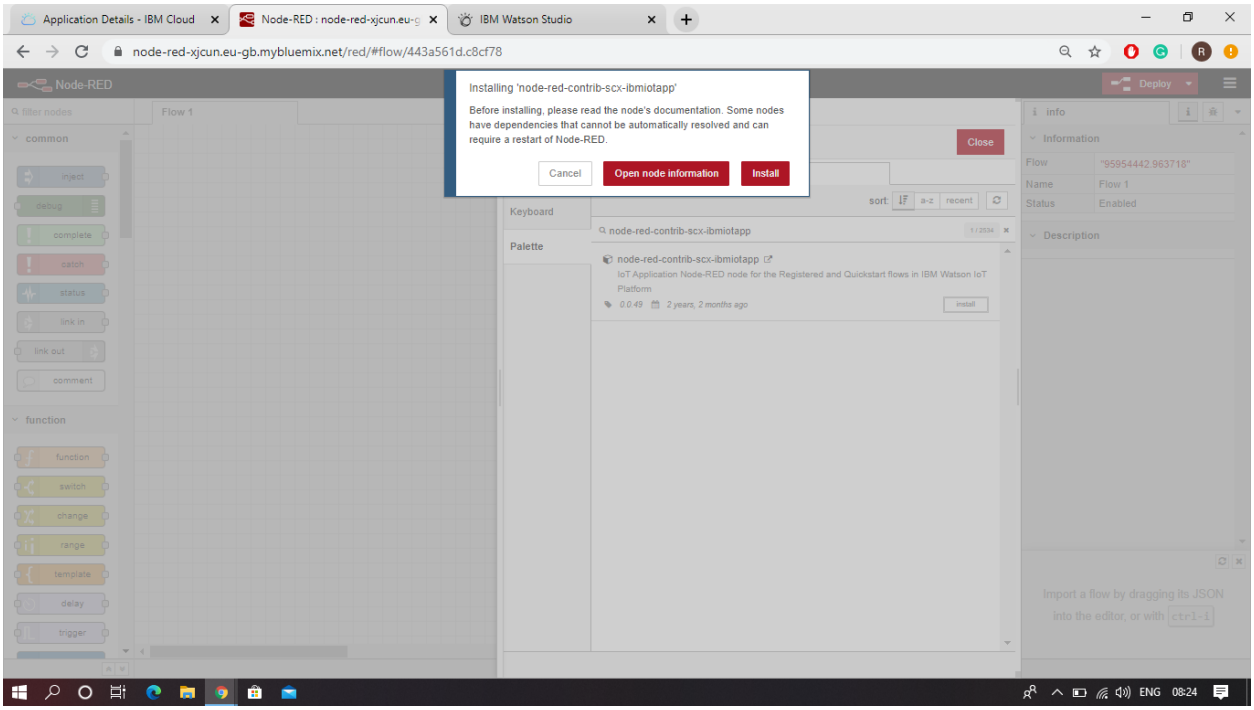


Installing Nodes

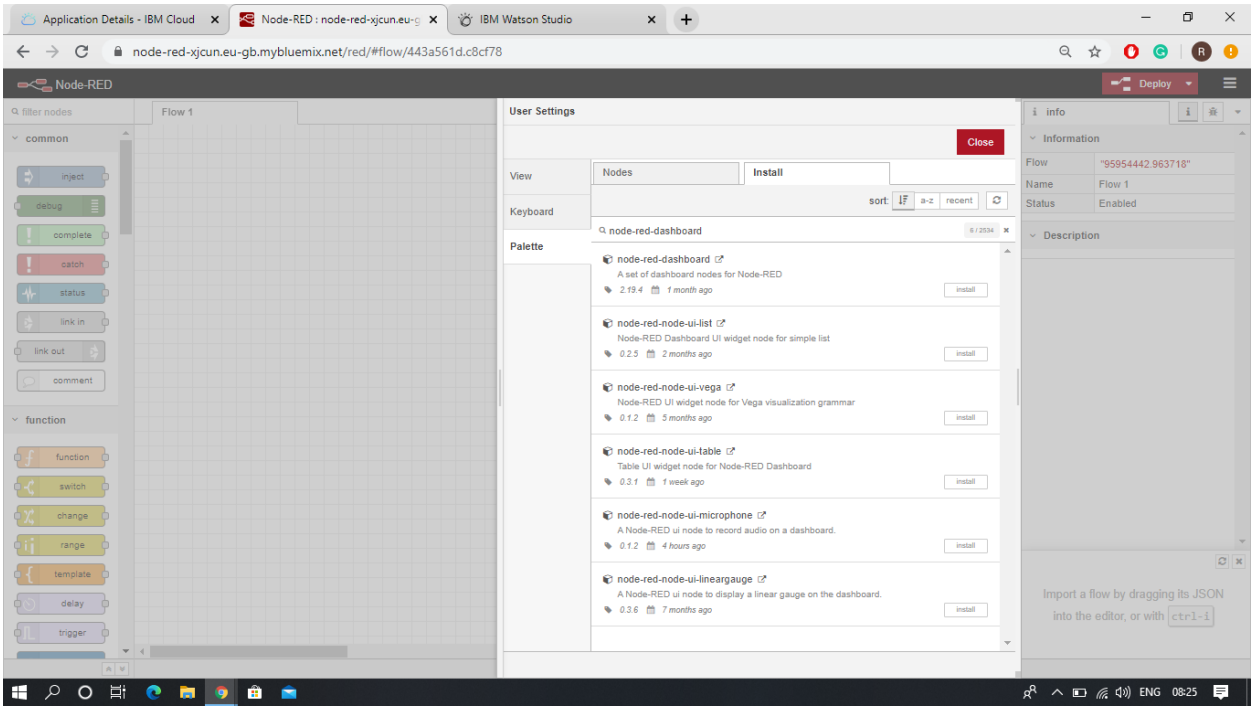
- Search for node-red-scx-ibmiotapp for installing the ibm iot app nodes used to interface with IBM IoT Watson Platforms. Click on Install.



- It will pop up to confirm the installation click on install.

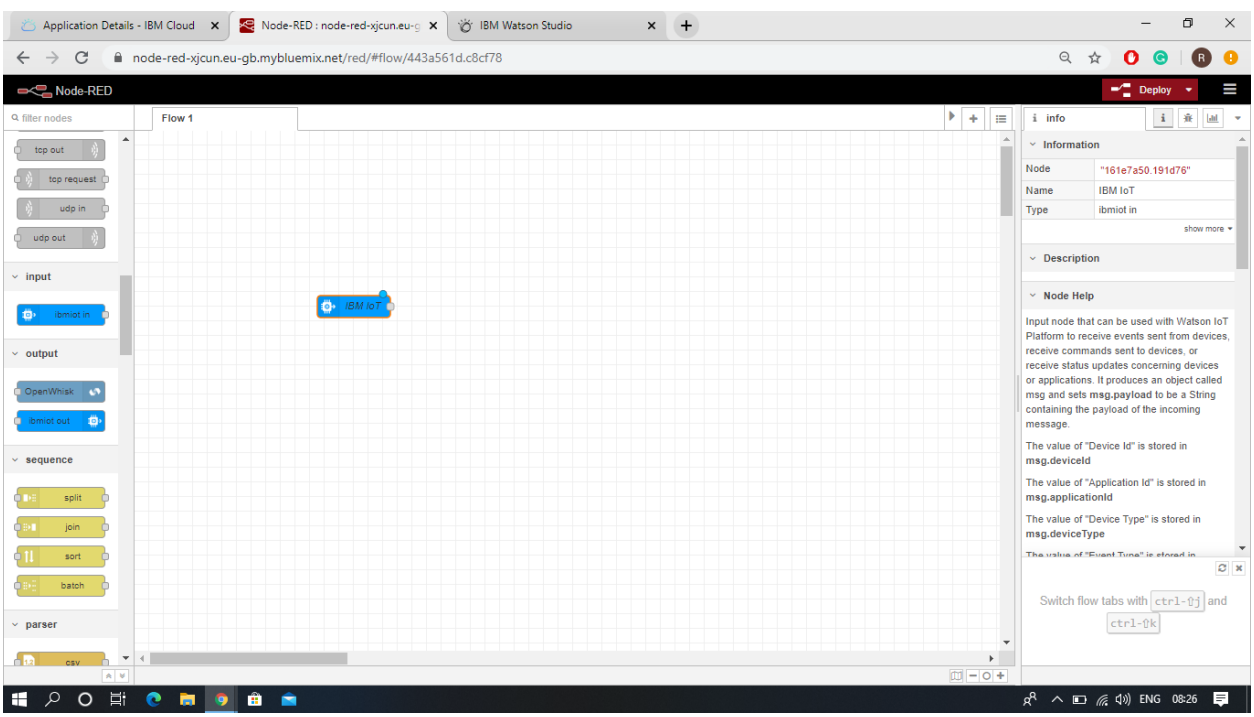


- In same manner we need to install Node-Red-Dashboard.

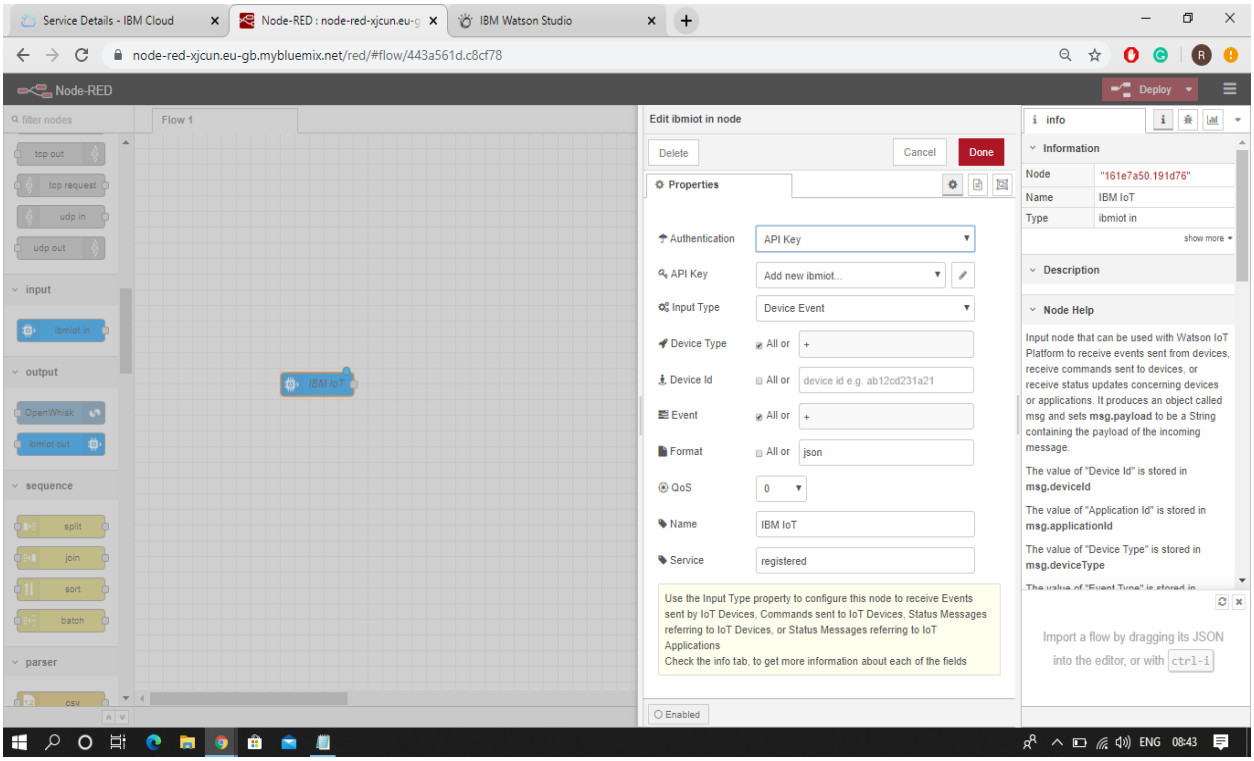


Connecting to IBM IOT Platform using Node Red

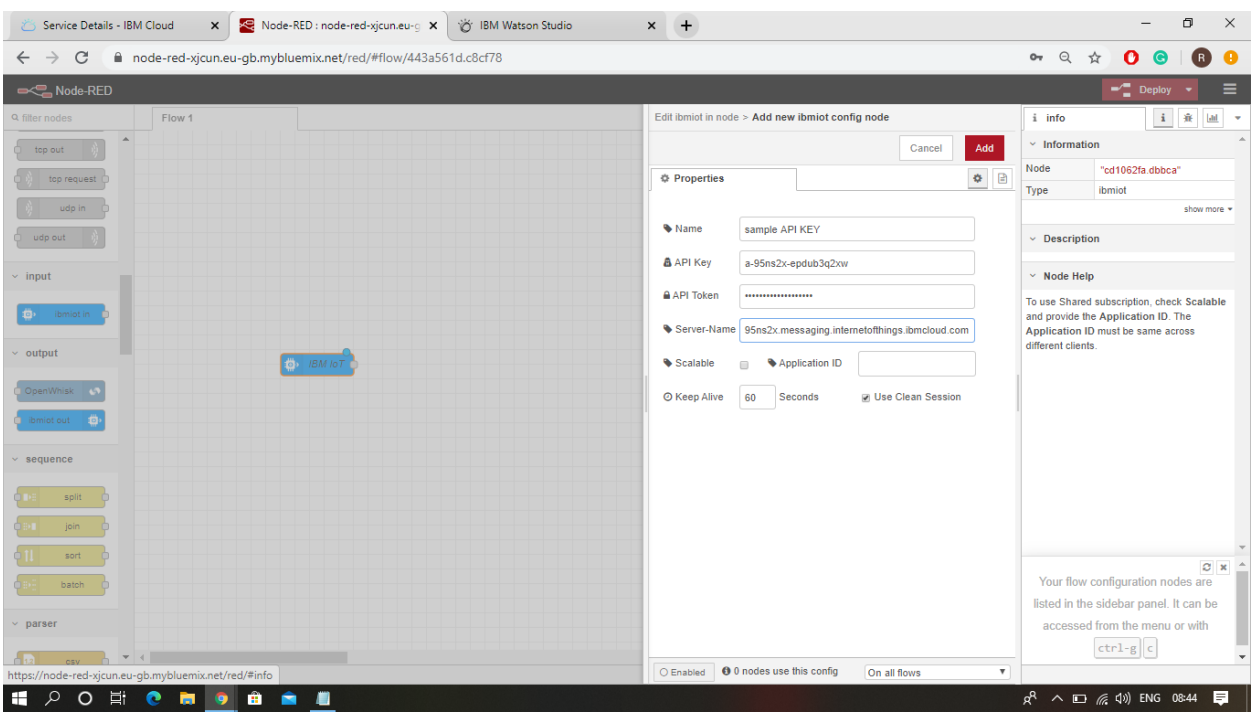
- Drag the IBM IoT node on to flow editor and double click to configure it.



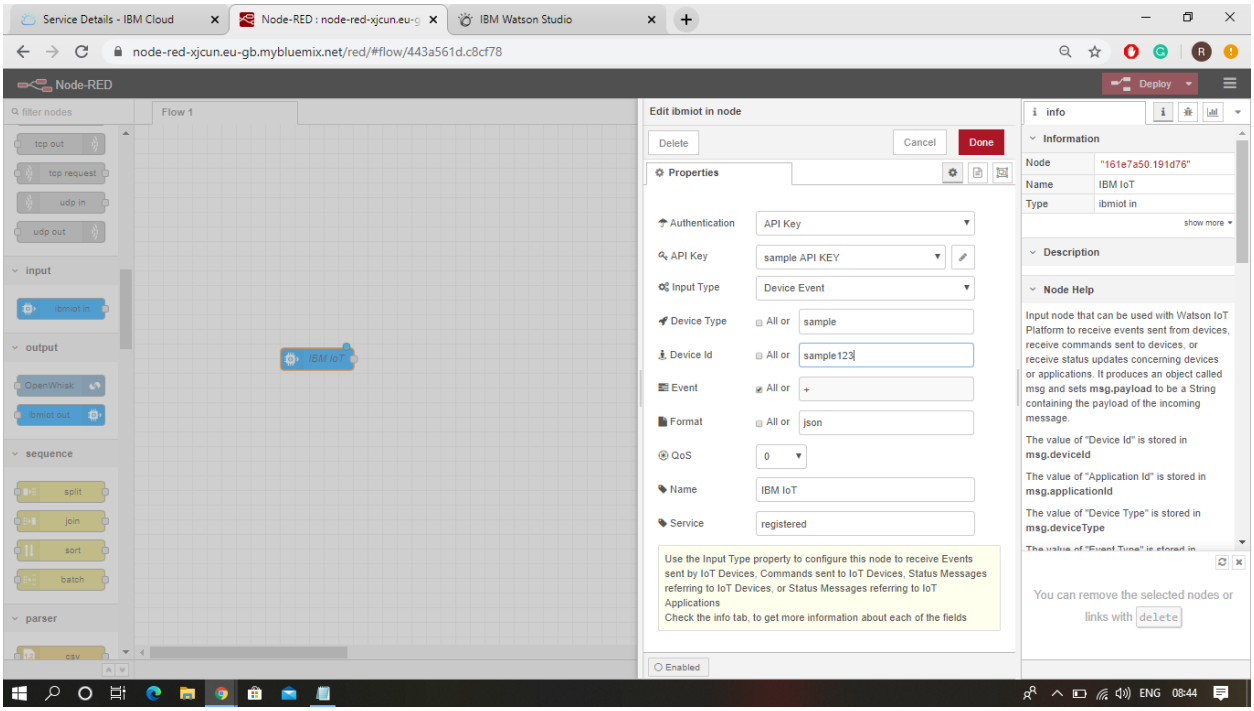
- Goto properties and Select API Key as Authentication.



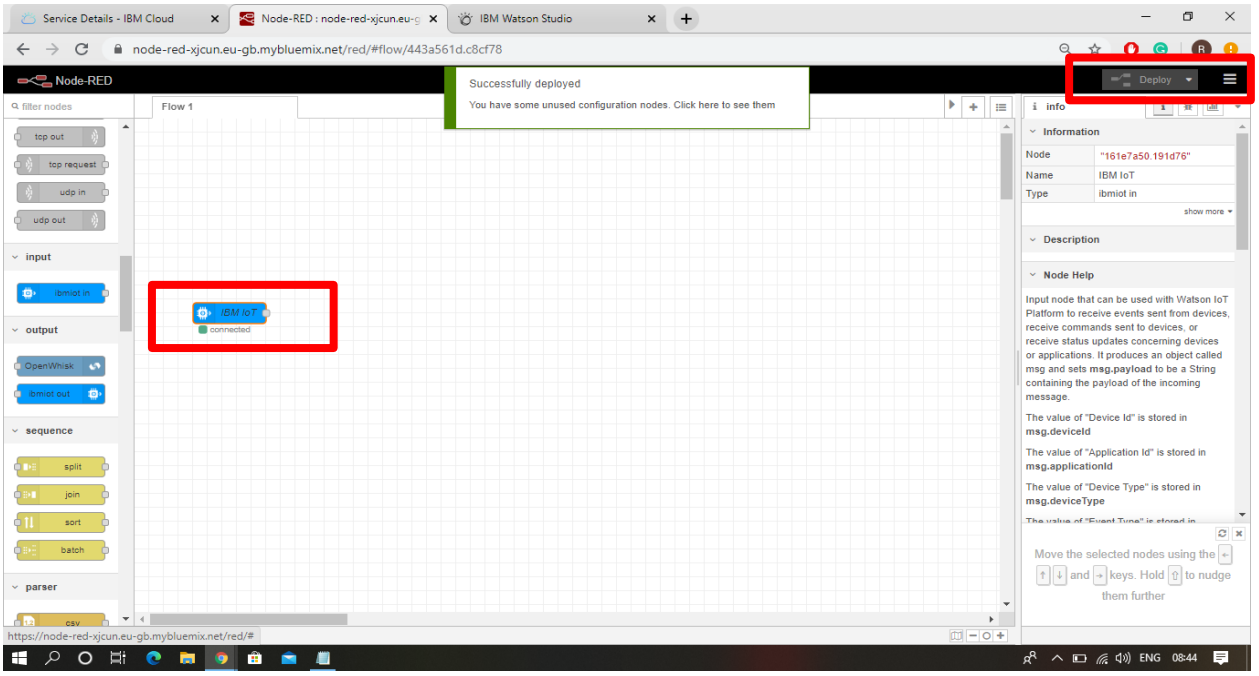
- In API Key paste API Key, API Token and server name and update it (or) click on add.



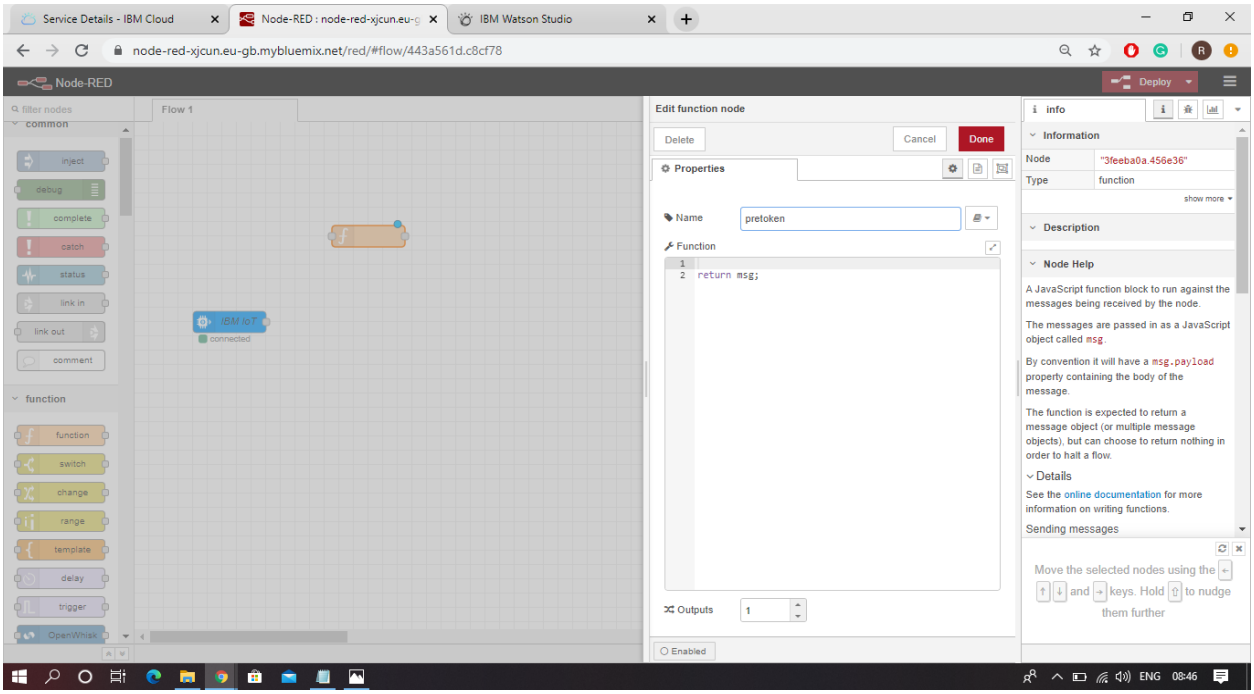
- Enter the other device credentials like device type and device Id, Format. And click Done



- Once done with providing the credentials,click on Deploy. We need to get the status under IBM IoT node as connected. If it is disconnected then you need to check with your Credentials.



- Drag the function node on to flow editor and configure function node.

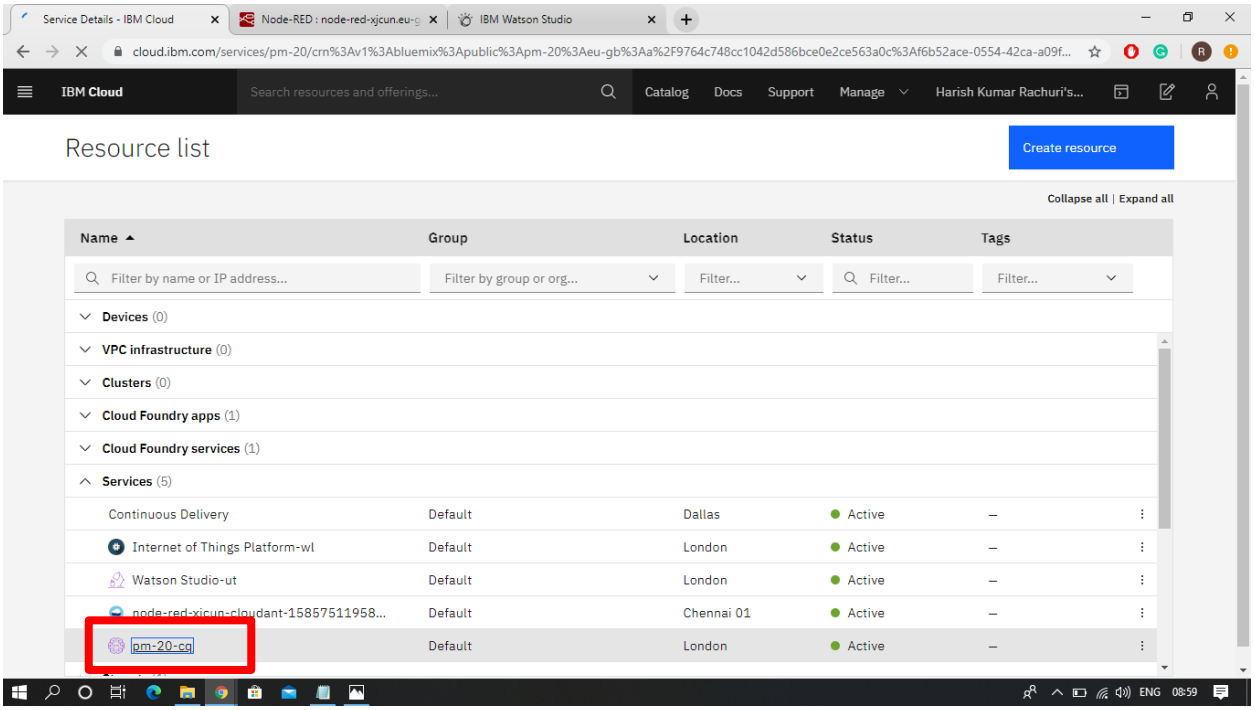


- Edit the code as follows

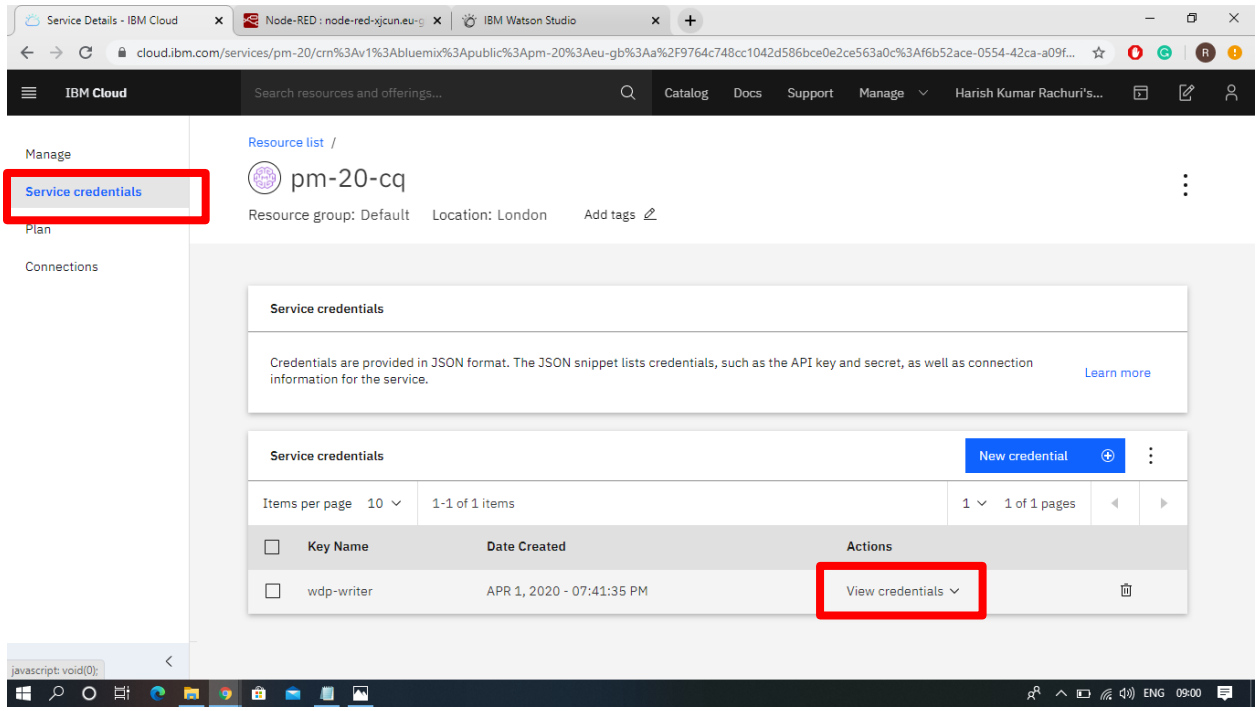
```
global.set ("age", msg.payload.d.age);

global.set ("Temperature", msg.payload.d.Temperature);

 global.set ("Systolic", msg.payload.d.Systolic);

global.set ("Diastolic", msg.payload.d.Diastolic);

global.set ("Pulse”, msg.payload.d.Pulse);

var apikey="RbSYuZU0Jk5WX_VvV9LqZztBF7pmgMLMsefzjwS0Mxh1";

msg.headers= {"content-type”: application/x-www-form-urlencoded"}

msg.payload= {"grant_type":"urn:ibm:params:oauth:grant-type:apikey","apikey":apikey}

return msg;
```

Note :You need to provide your API key of ML instance.

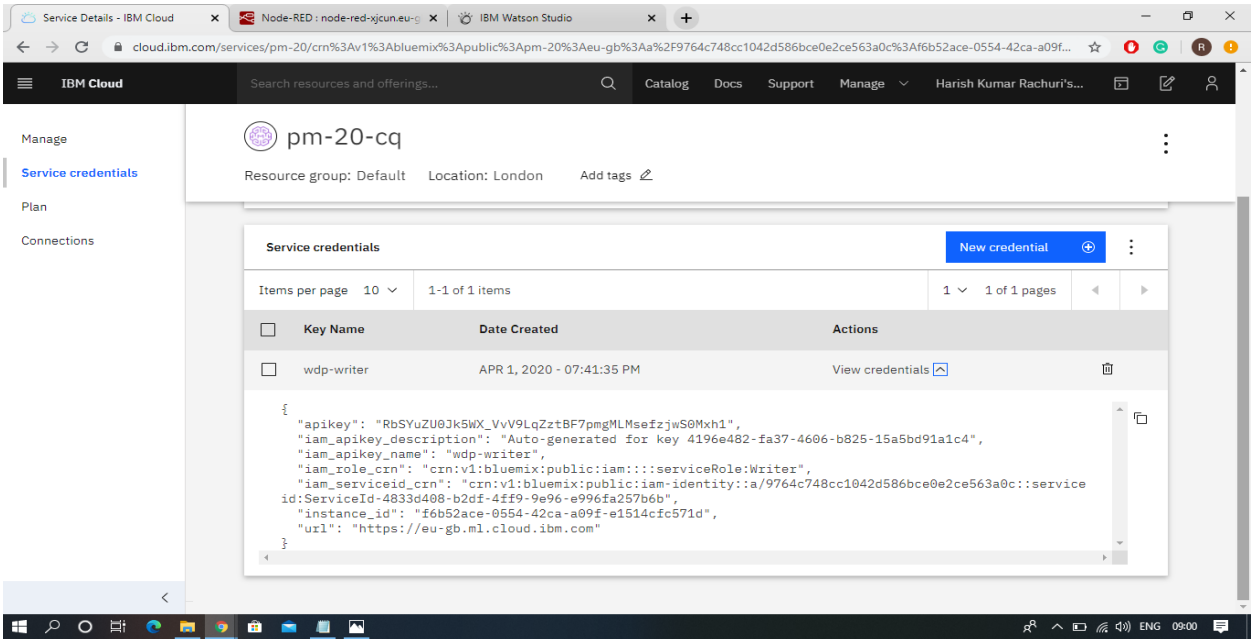
- In the above function you need to provide your api key, For API KEY and Instance id of Machine learning model navigate to dash Board and click on machine learning model



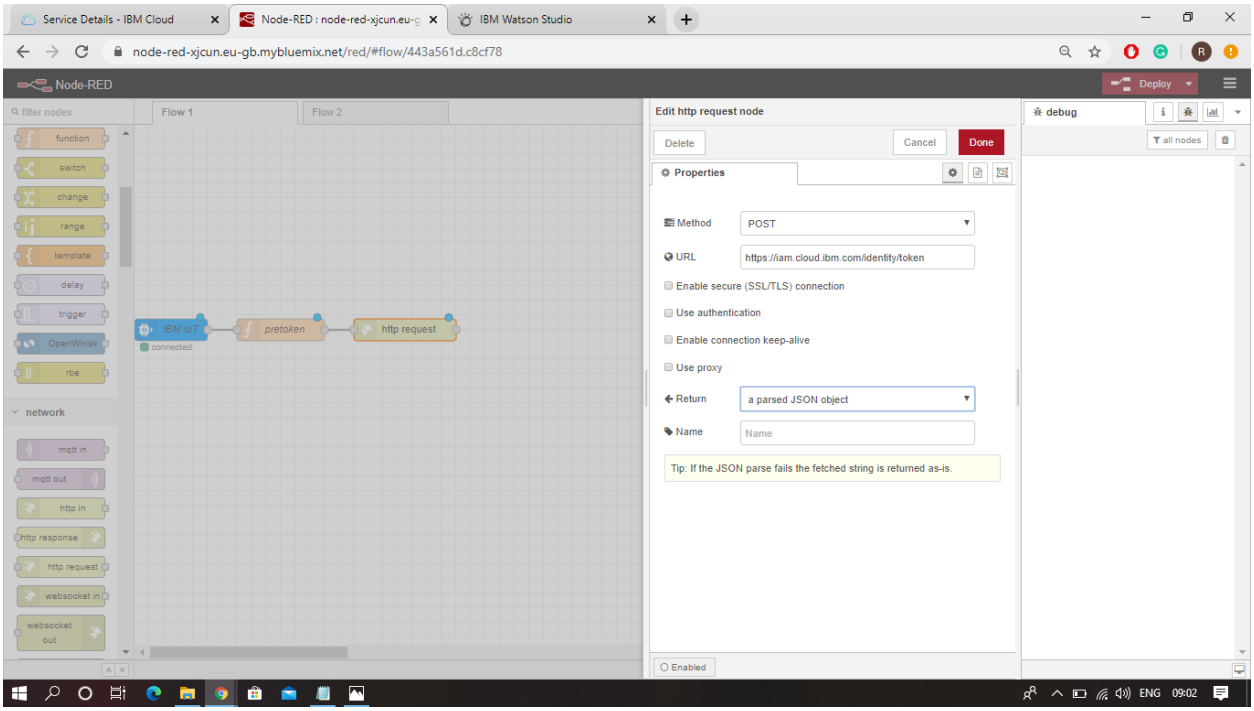
- It will open the screen. Select service credentials and click on view credentials.



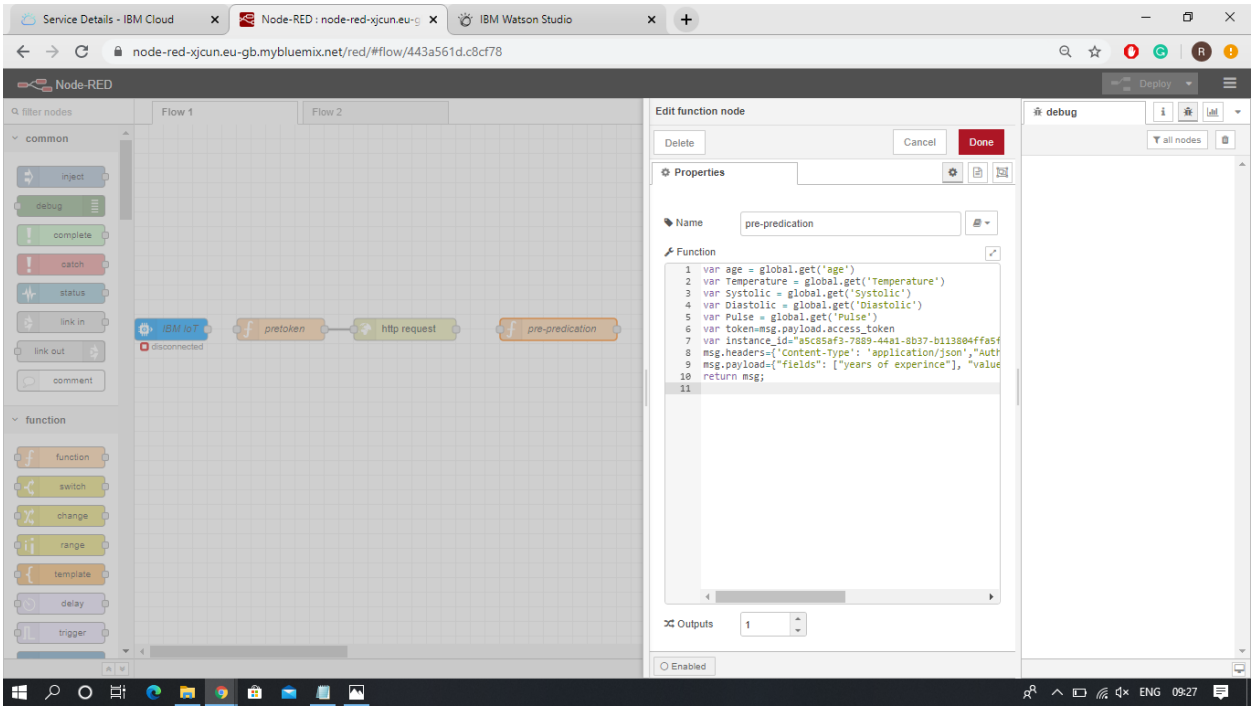
- We can fine the Credentials like API key, Instance ID. Copy it and save for future reference.



- Drag and drop Http Request node on to the flow editor and click to configure. Select the method as Post and enter the URL as https://iam.cloud.ibm.com/identity/token and return as "a parsed JSON Object"



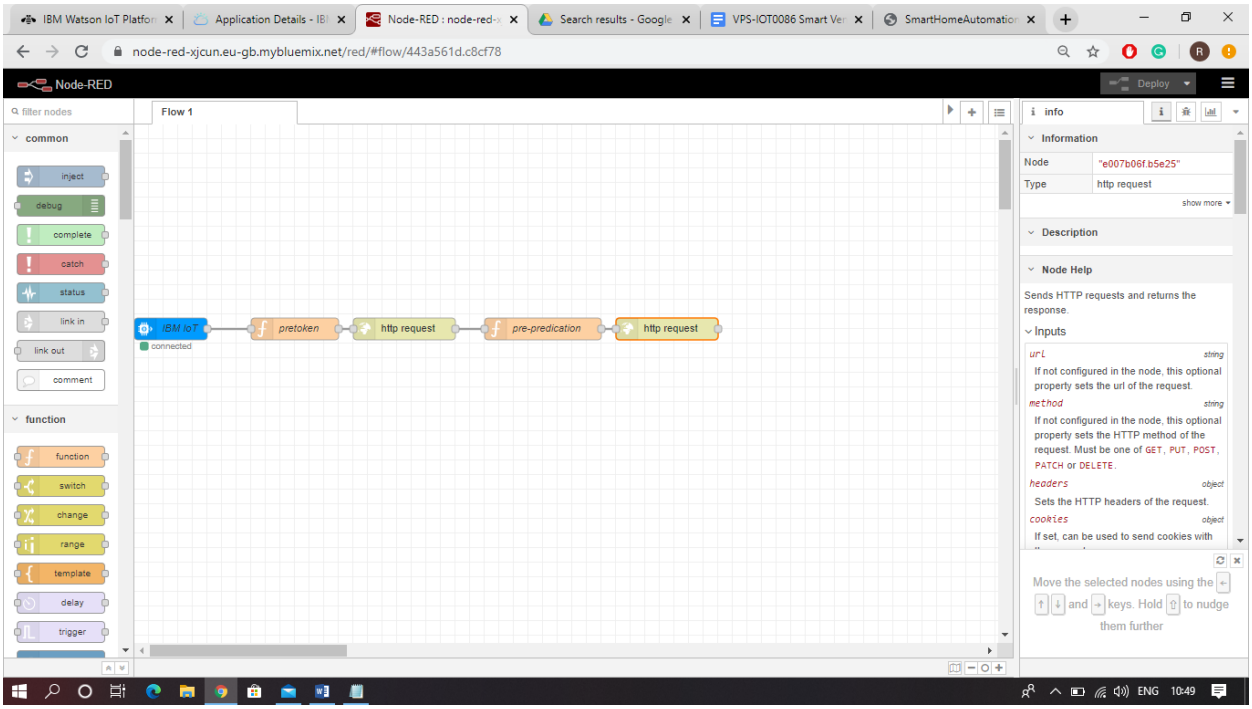
- Drag and Drop function node and configure with



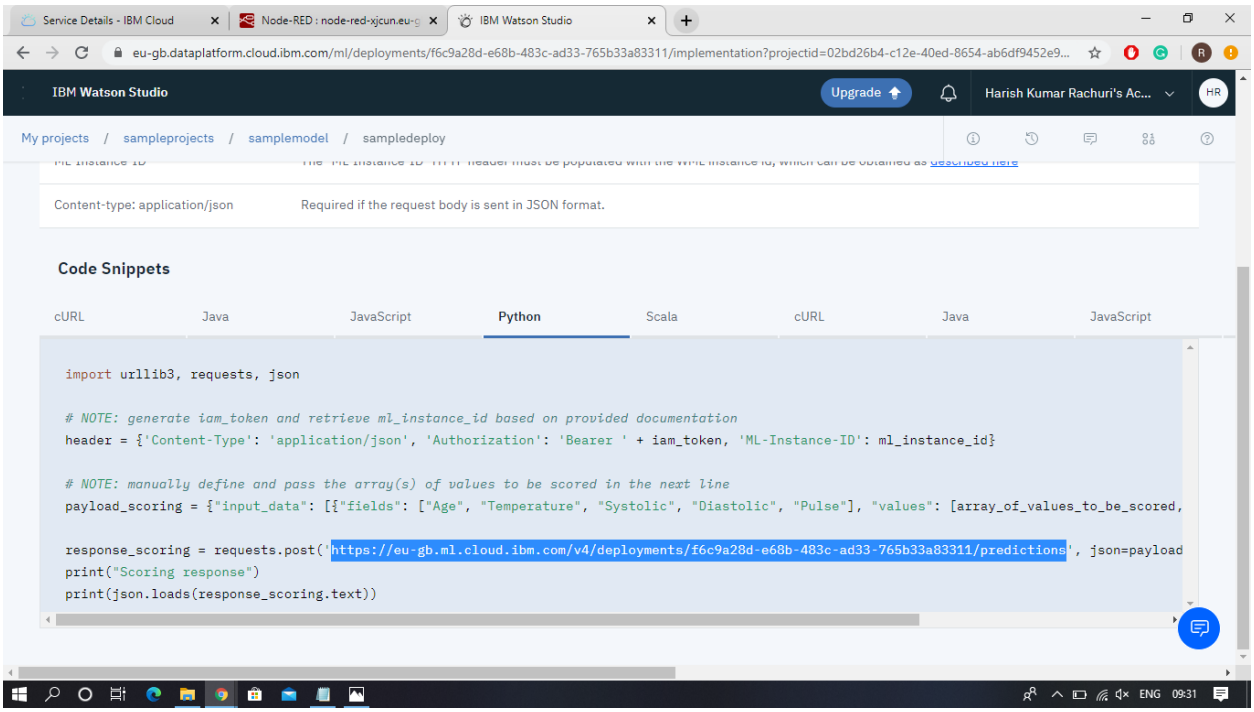
- Edit the code with your Instance ID and msg.payload with your input parameters.

```
var age = global.get('age')

var Temperature = global.get('Temperature')

var Systolic = global.get('Systolic')

var Diastolic = global.get('Diastolic')

var Pulse = global.get('Pulse')

var token=msg.payload.access_token

var instance_id="f6b52ace-0554-42ca-a09f-e1514cfc571d"

msg.headers={'Content-Type': 'application/json","Authorization":"Bearer "+token,"ML-Instance-ID":instance_id}

msg.payload={"input_data": [{"fields": ["Age", "Temperature", "Systolic", "Diastolic", "Pulse"], "values": [[age,Temperature,Systolic,Diastolic,Pulse]]}]}

return msg;
```

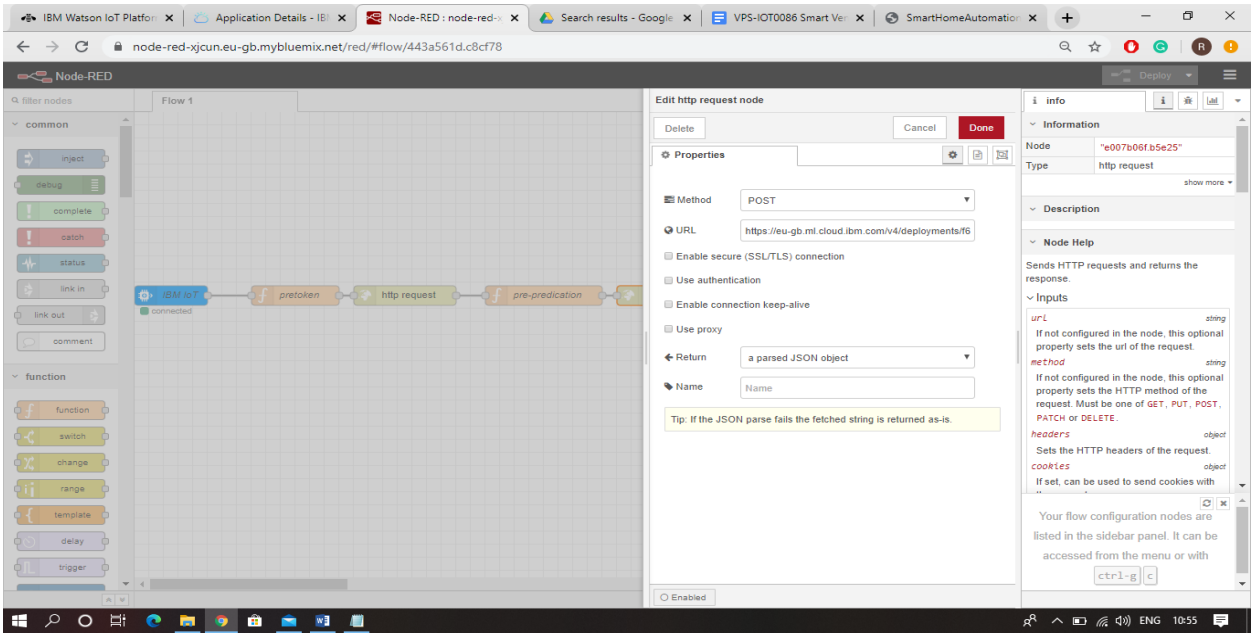
- Drag and Drop the second Http Request node



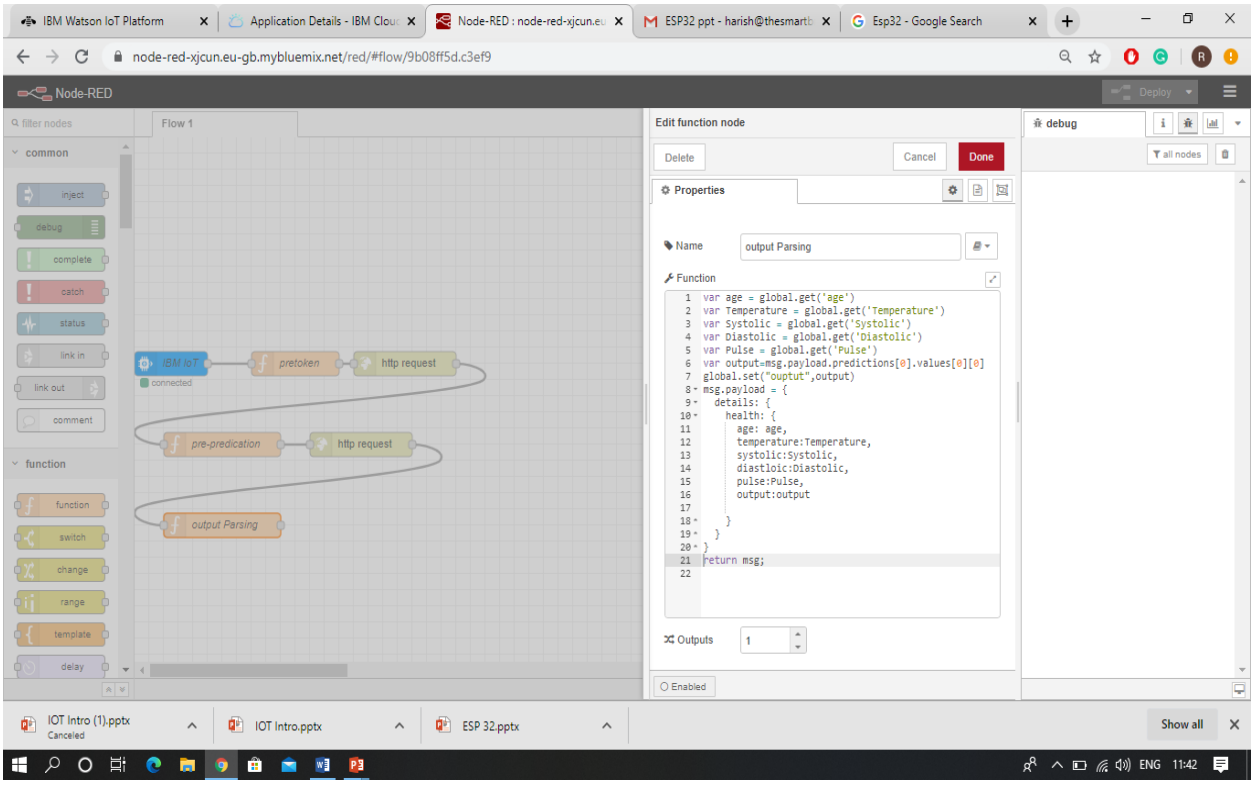
- copy the prediction URL from IBM Watson Studio python code.



- paste the URL in Http Request node and select the method as “post” and returns as “a parsed JSON object”.



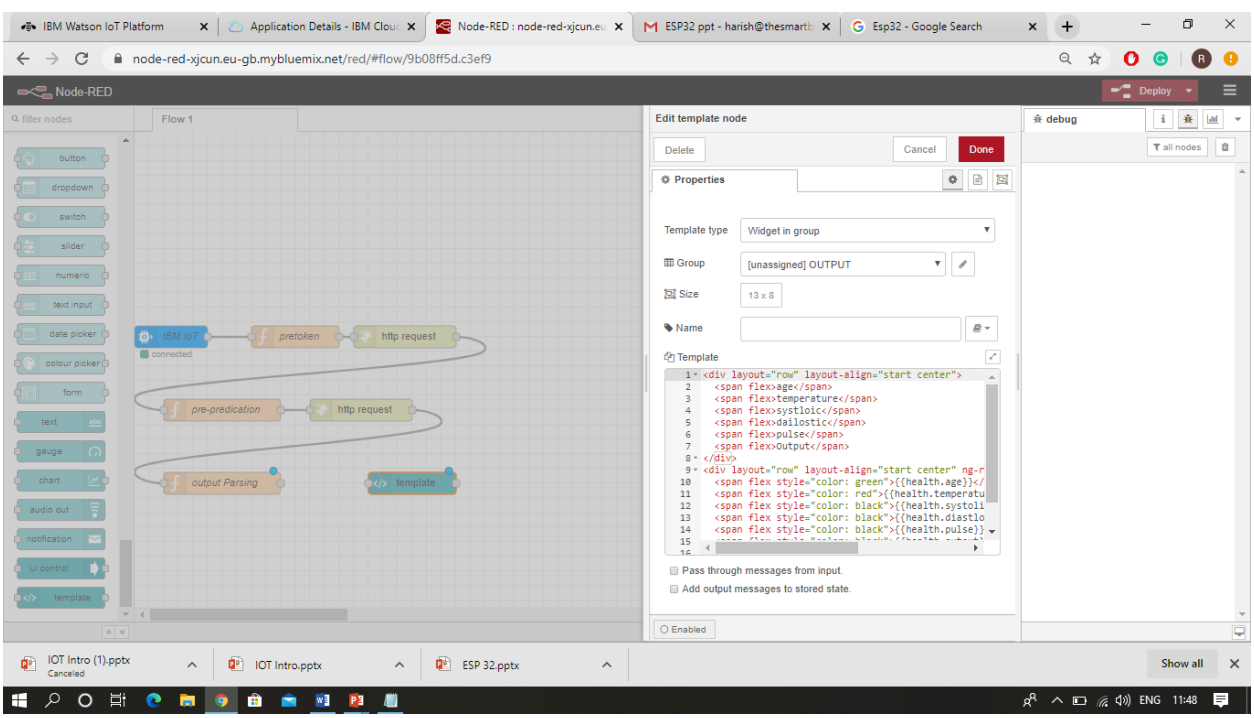
- Drag and drop the function node for parsing the Received Output from machine learning model.
- set the output as global variables.



```
var age = global.get ('age')
var Temperature = global.get ('Temperature')
var Systolic = global.get ('Systolic')
var Diastolic = global.get ('Diastolic')
var Pulse = global.get ('Pulse')
var output=msg.payload.predictions[0].values[0][0]
global.set("ouptut",output)
msg.payload = {
  details: {
    health: {
      age: age,
      temperature: Temperature,
      systolic: Systolic,
      diastloic:Diastolic,
      pulse: Pulse,
      output: output
    }
  }
}
return msg;
```

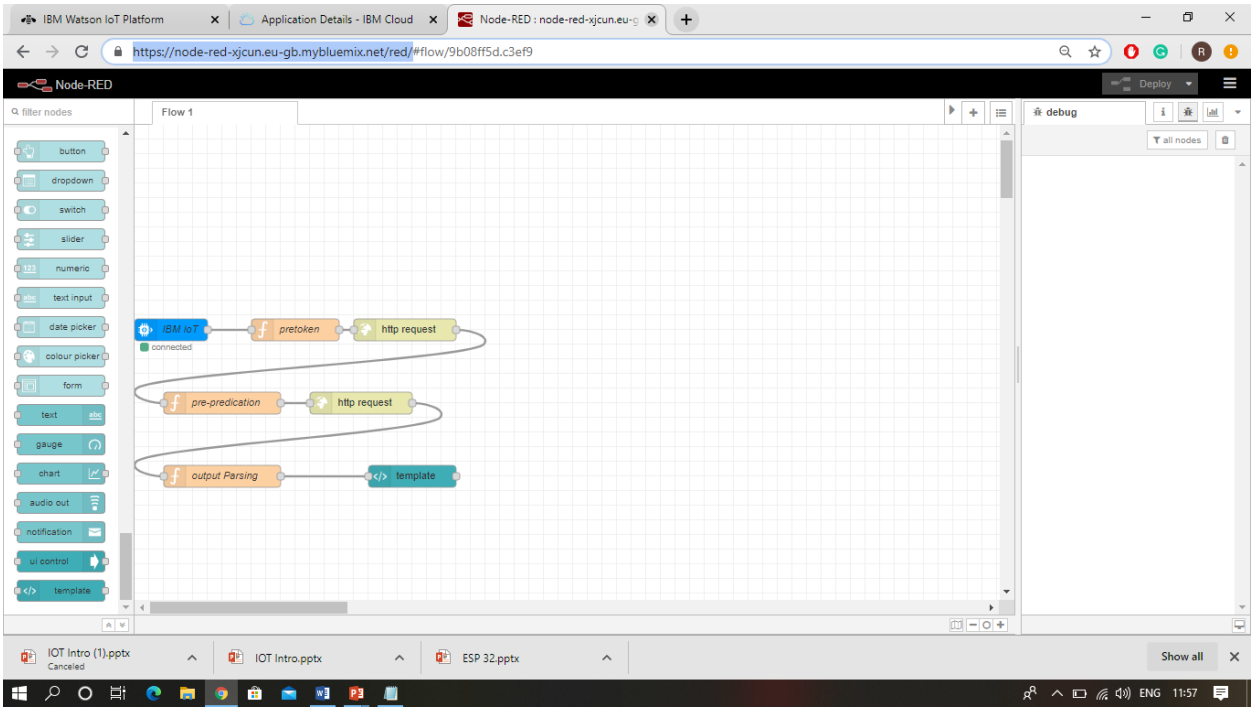
Task 7: Creating the Web UI to visualize the data

- From the dashboard nodes, Drag and Drop the template node. Double click to configure the node, we need to write some html Coding to display the value on the web UI.

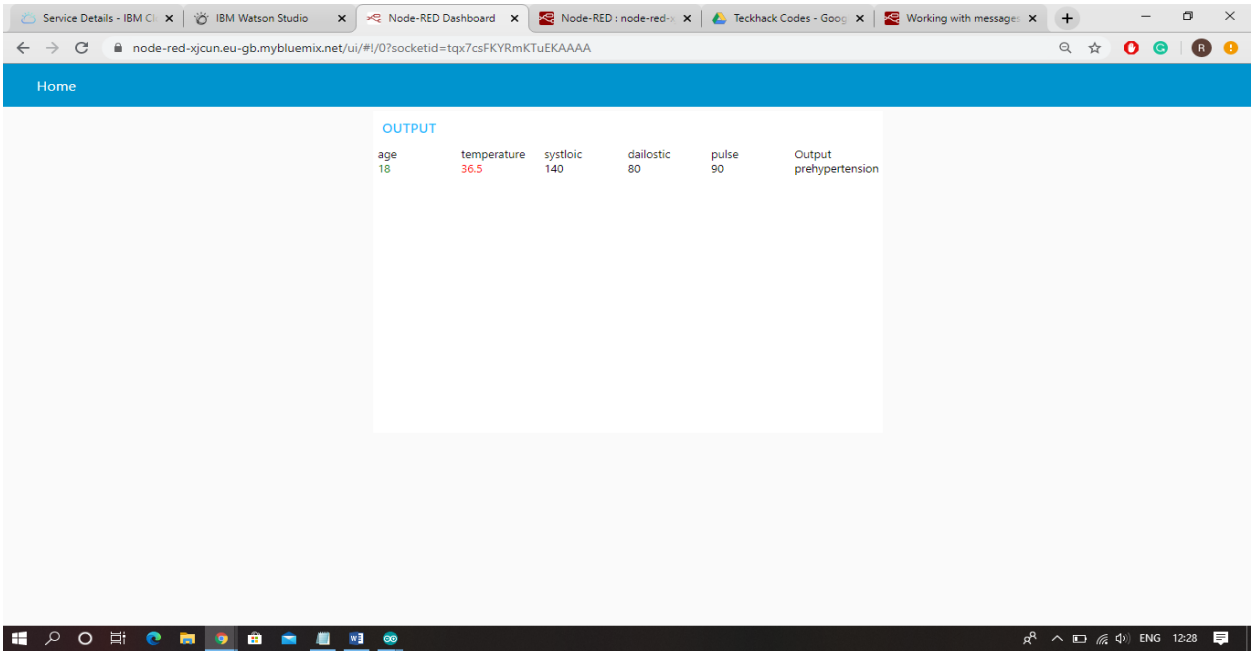


```
<div layout="row" layout-align="start center">
 <span flex>age</span>
 <span flex>temperature</span>
 <span flex>systloic</span>
 <span flex>dailostic</span>
 <span flex>pulse</span>
 <span flex>Output</span>
</div>
<div layout="row" layout-align="start center" ng-repeat="health in msg.payload.details">
 <span flex style="color: green">{{health.age}} </span>
 <span flex style="color: red"> {{health.temperature}}</span>
 <span flex style="color: blue"> {{health.systolic}} </span>
 <span flex style="color: yellow"> {{health.diastloic}} </span>
 <span flex style="color: black"> {{health.pulse}} </span>
 <span flex style="color: Brown"> {{health.output}} </span>
 </div>
```

- copy the Node Red URL till ".net" or ".cloud" and append "/ui" and paste it in new tab and click enter to get the WEB UI page.

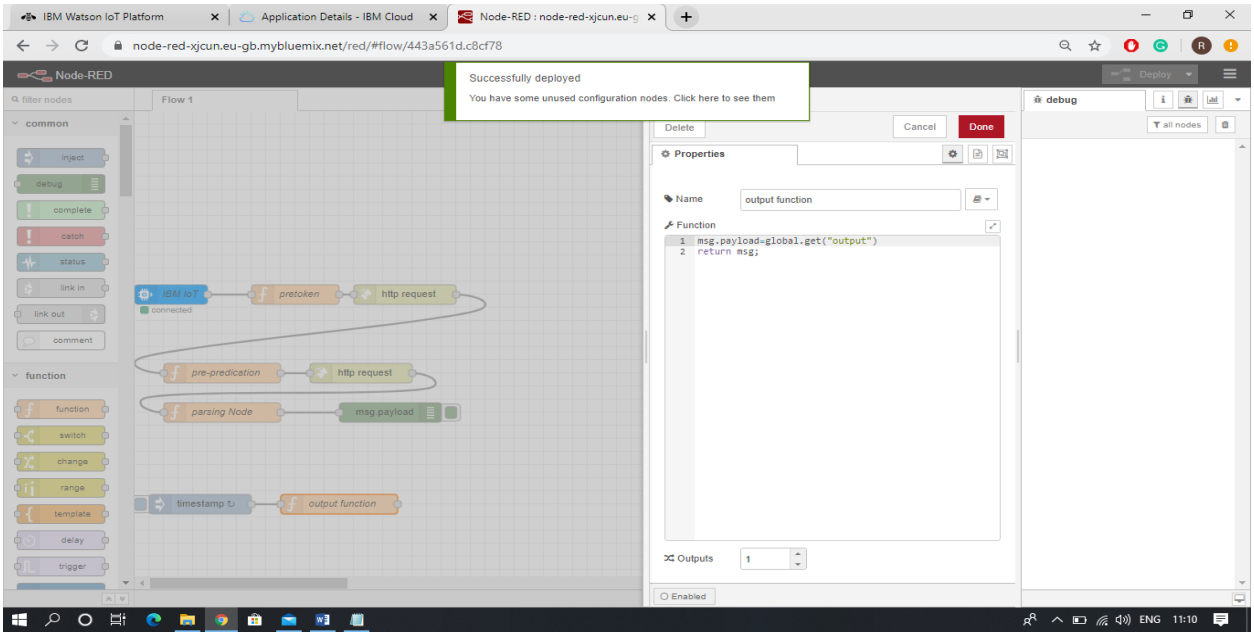


- it will display the web UI in Web browser.



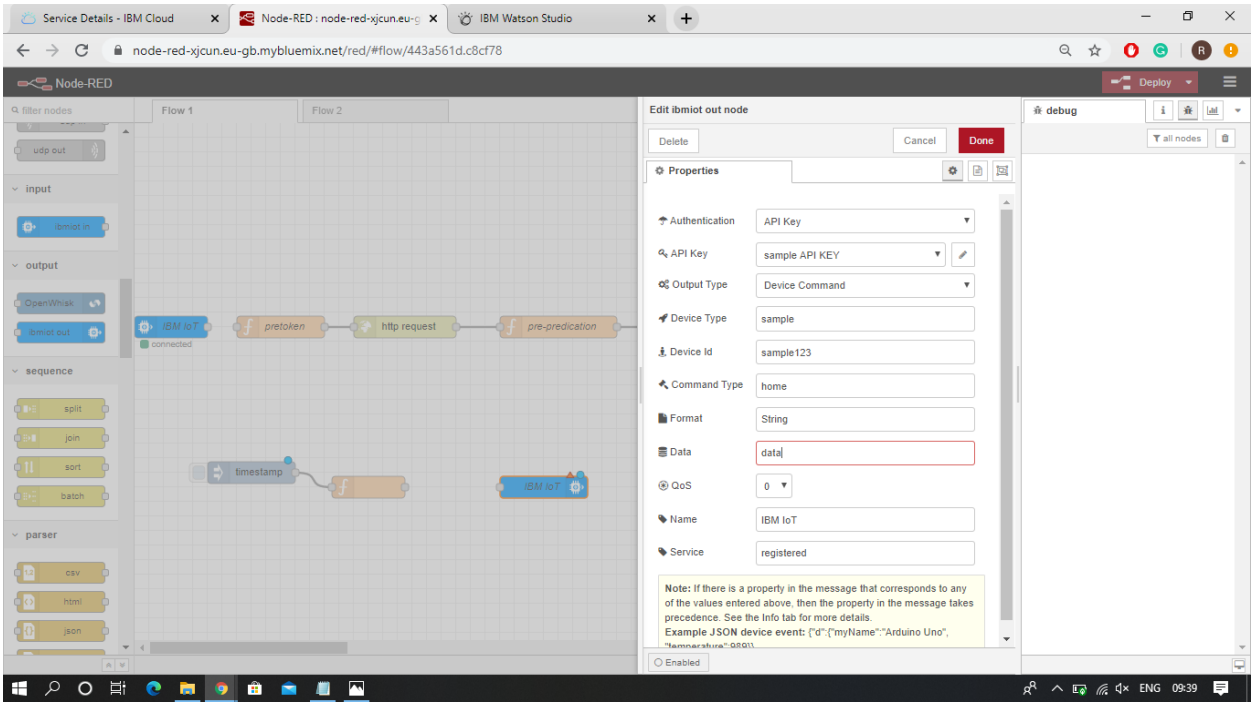
Sending the predicted output of model to IBM IOT Device.

- Take the inject node and function node on to the flow editor. write function in function node for getting the global variables. Which was set in previous function Node.

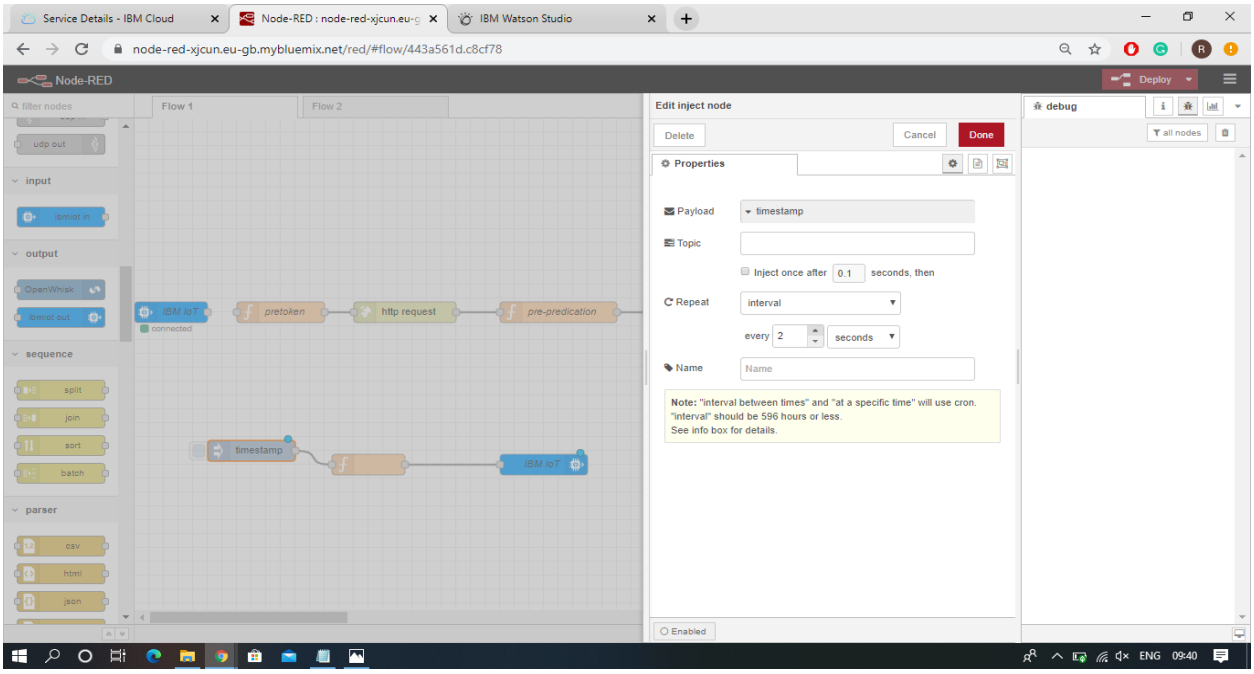


- Drag the IBM IoT output node and configure it. select the authentication as API key, Select the API which was configured for IBM IoT Input node or you can again provide the

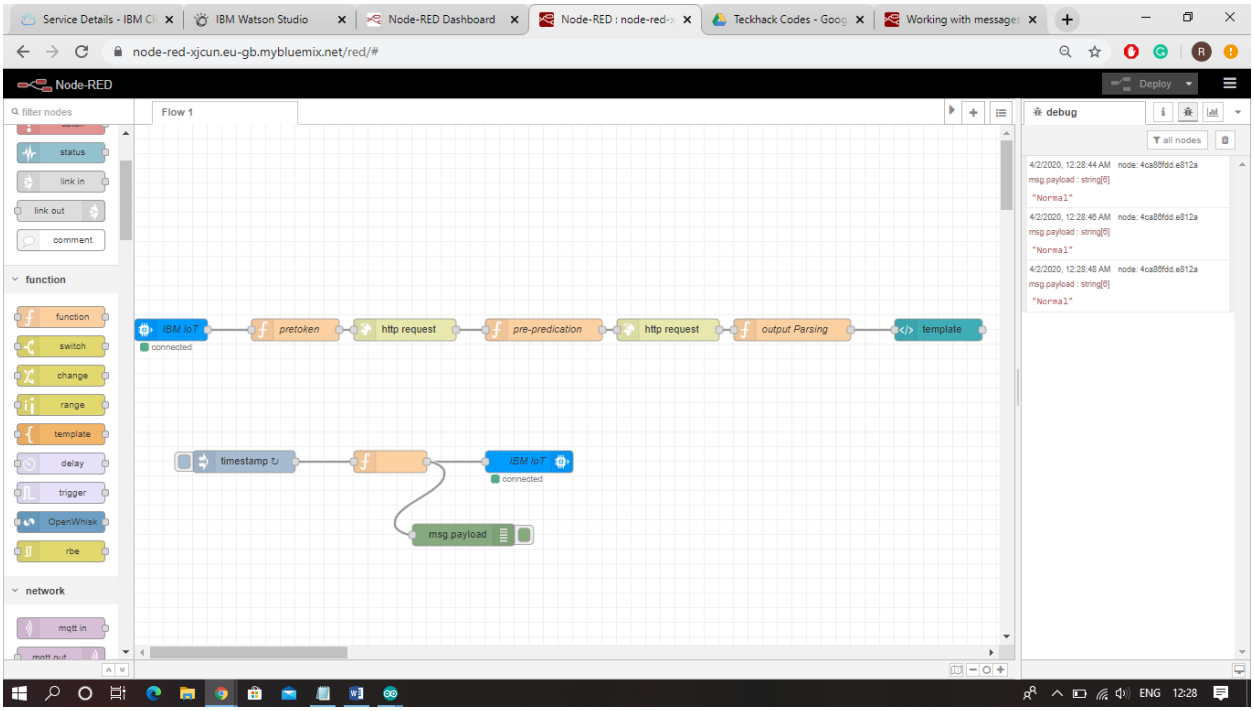
credentials. Then select the output type as Device command and provide the other credentials like device id, device type. Give command type as “home” and format as “String” and data as “Data”.



- Configure the inject node to inject (time Stamp) for every two seconds.



- Drag and drop the debug node and attach it to function node for checking the output.


Service Details - IBM Cl
IBM Watson Studio
Node-RED Dashboard
Node-RED : node-red-
Teckhack Codes - Goog
Working with messages
node-red-xjcun.eu-gb.mybluemix.net/red/#
Node-RED
Deploy
filter nodes
Flow 1
status
link in
link out
comment
function
function
switch
change
range
template
delay
trigger
OpenWhisk
rbe
network
mqtt in
IBM IoT
connected
pretoken
http request
pre-predication
http request
output Parsing
template
timestamp
IBM IoT
connected
msg.payload
debug
all nodes
4/2/2020, 12:28:44 AM node: 4ca88fdd.e812a
msg.payload : string[6]
"Normal"
4/2/2020, 12:28:46 AM node: 4ca88fdd.e812a
msg.payload : string[6]
"Normal"
4/2/2020, 12:28:48 AM node: 4ca88fdd.e812a
msg.payload : string[6]
"Normal"
ENG
12:28