

SPRINT 2

CREATING IBM CLOUD SERVICE AND IBM WATSON IOT PLATFORM

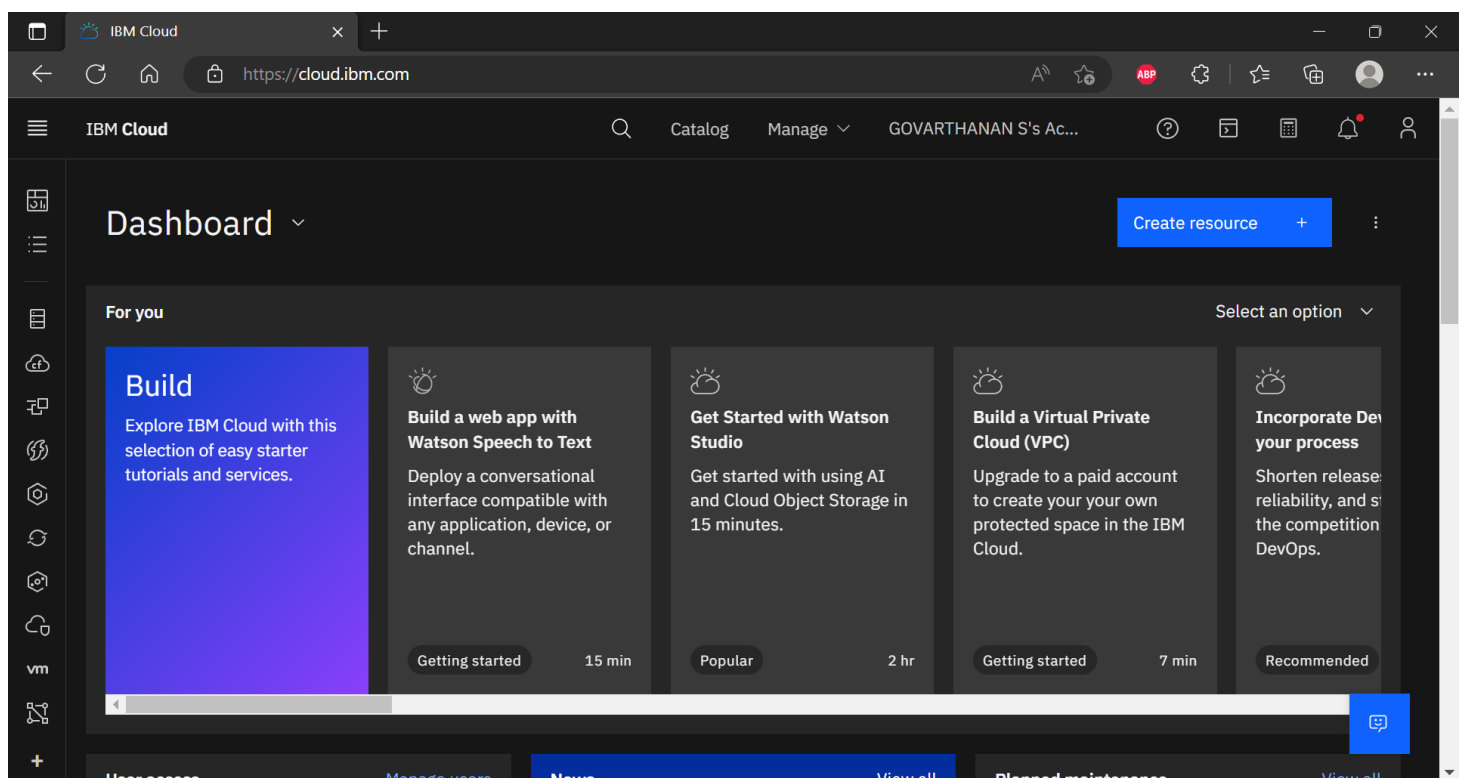
Team ID	PNT2022TMID04590
Project Name	Project - IoT Based Safety Gadget for Child Safety Monitoring and Notification

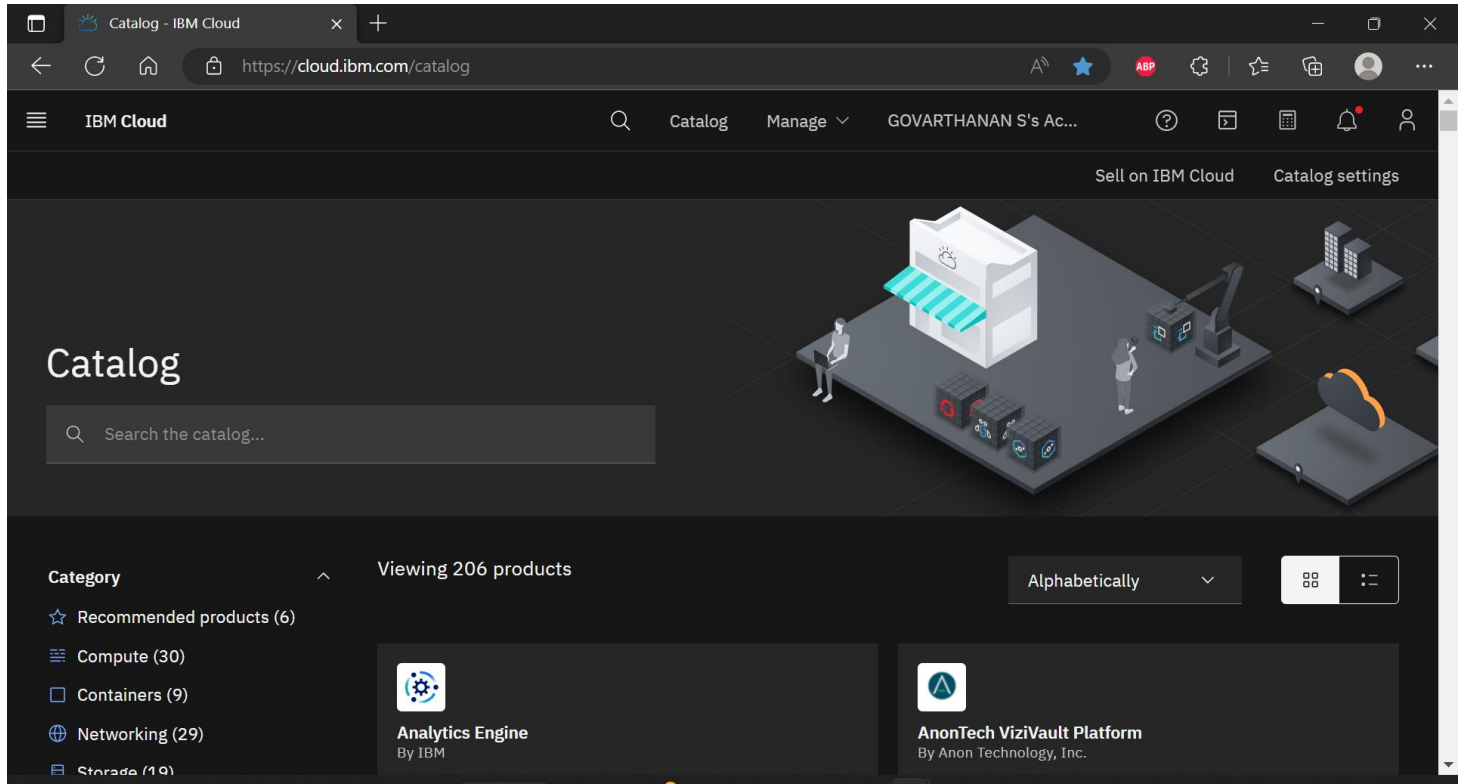
Aim:

To create account and configure in IBM cloud and IBM Watson iot platform.

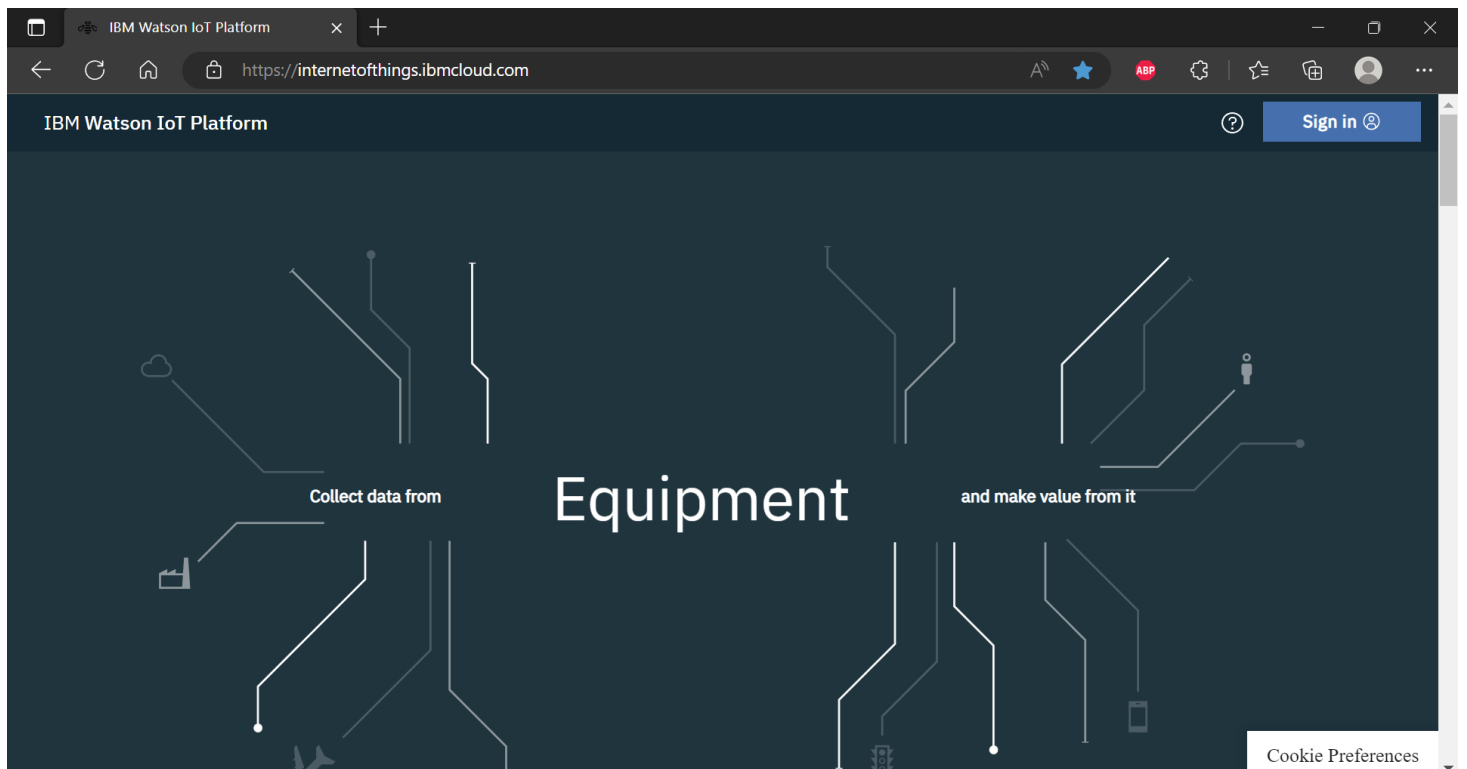
Steps:

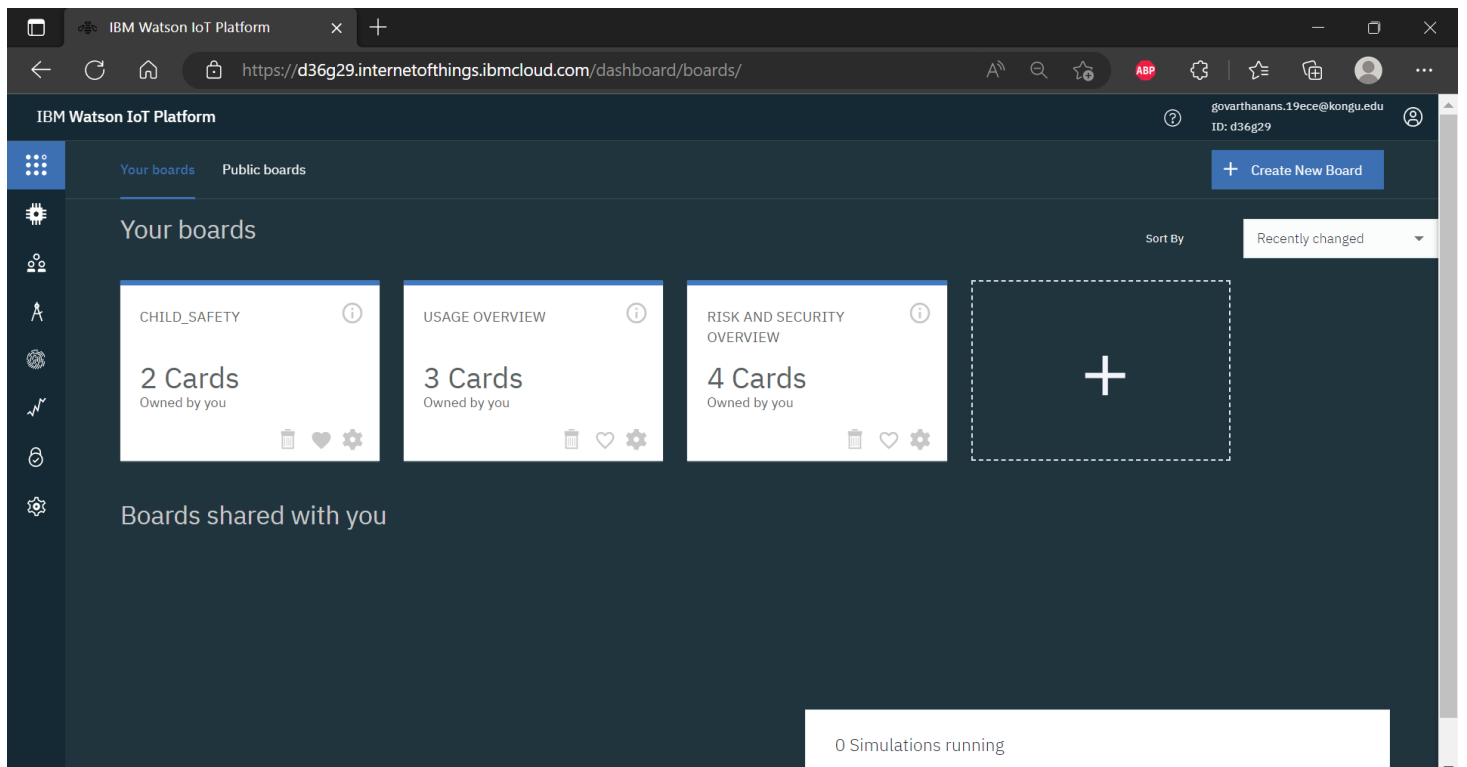
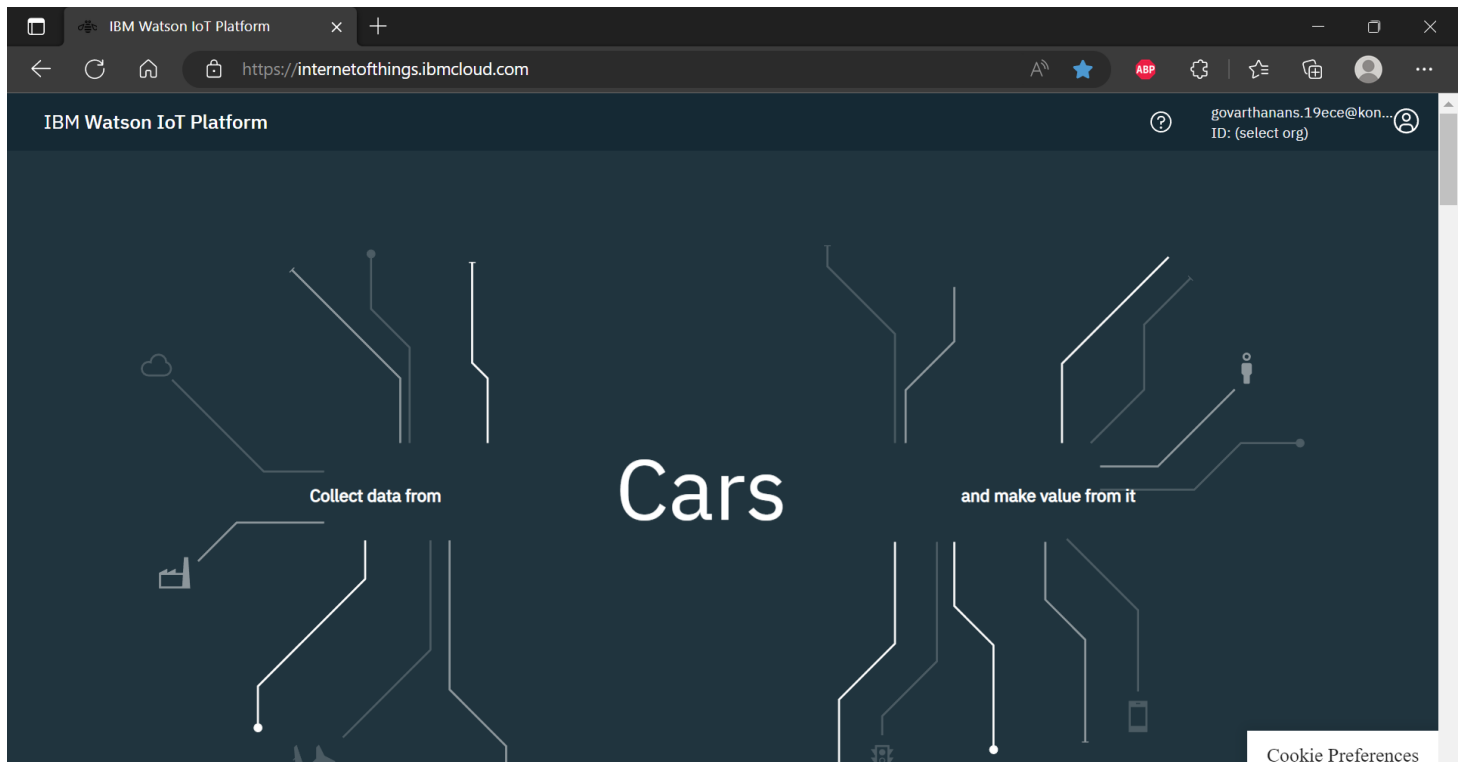
IBM CLOUD CONFIGURATION:





IBM WATSON IOT CONFIGURATION:





The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. A table lists devices with columns: Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. Two devices are listed: ID 194590 (Tracker, Disconnected) and ID 4590 (NodeMCU, Disconnected). The device ID 4590 is selected, and its details are shown in a modal window. The details include: Device ID (4590), Device Type (NodeMCU), Date Added (Nov 9, 2022 11:34 AM), Added By (govarthanans.19ece@kongu.edu), and Connection Status (Disconnected). A button 'Add Device' is visible in the top right corner.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
194590	Disconnected	Tracker	Device	Nov 19, 2022 3:48 PM	
4590	Disconnected	NodeMCU	Device	Nov 9, 2022 11:34 AM	

Identity	Device Information	Recent Events	State	Logs
Device ID	4590			
Device Type	NodeMCU			
Date Added	Nov 9, 2022 11:34 AM			
Added By	govarthanans.19ece@kongu.edu			
Connection Status	Disconnected			

0 Simulations running

The screenshot shows a Visual Studio Code editor with a file named 'geofence.py'. The script is a Python program that connects to the IBM Watson IoT Platform. It imports necessary modules like json, wiotp.sdk.device, time, sys, ibmiotf.application, and ibmiotf.device. It sets up variables for organization, deviceType, deviceId, authMethod, and authToken. The script then attempts to create a device client and connect to the platform. If an exception occurs, it prints an error message and exits. Otherwise, it prints a success message and enters a loop where it updates the device's location.

```

1 import json
2 import wiotp.sdk.device
3 import time
4 import sys
5 import ibmiotf.application
6 import ibmiotf.device
7
8 organization = "d36g29"
9 deviceType = "NodeMCU"
10 deviceId = "4590"
11 authMethod = "token"
12 authToken = "(LBfkwoyWYXONmJevC"
13
14 try:
15     deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
16     deviceCli = ibmiotf.device.Client(deviceOptions)
17 except Exception as e:
18     print("Caught exception connecting device: %s" % str(e))
19     sys.exit()
20
21 print("CHECKING CONNECTION TO IBM WATSON. ")
22
23 time.sleep(2)
24 deviceCli.connect()
25 print("dear user welcome to IBM-IOT ")
26 while True:
27     name= "Smartsafety"
28     #area_location
  
```

Result:
create account and configure in IBM cloud and IBM Watson iot platform successful...