

**ponents & Technologies:**

S.No	Component	Description	Technology
1	User Interface	How user interacts with application	HTML ,CSS, Javascript, React Js
2	Application Logic-1	Logic for the application	Python
3	Application Logic-2	Logic for the application	IBM Watson IOT service
4	Application Logic-3	Logic for the application	MySQL,NoSQL,etc.
5	Cloud Database	Database service on cloud	IBM Cloud
6	File Storage	It store the required files	IBM Block Storage or local filesystem
7	External API-1	Purpose of external application has been used in the application	IBM Weather API
8	Machine Learning Model	Purpose of machine learning model	Object recognition model
9	Infrastructure	Local Server and cloud server configuration	Local, cloud foundry

**Application Characteristics:**

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the frameworks used	Opensource frameworks
2	Security Implementations	Sensitive and private data must be protected from their production until the decision making and storage stages	Node red, open weather API, MIT app Inventor
3	Scalable Architecture	Scalability is a major concern for IOT platforms. It is choices of these platforms that would affect the scalability and that automatic real time decision making is feasible in an environment.	Technology used

## APPLICATION PLATFORM AND CLOUD:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays a table of devices. One device, with ID '12345', is highlighted. Below the table, a detailed view of the device is shown, including its identity, device information, recent events, state, and logs. The device is connected and was added on Nov 23, 2022, 9:11 PM. The connection status is 'Connected' with a connection time of Nov 24, 2022, 11:41 AM and a client address of 157.49.150.32.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	new	Device	Nov 23, 2022 9:11 PM	

**Device Information:**

- Device ID: 12345
- Device Type: new
- Date Added: Nov 23, 2022 9:11 PM
- Added By: 1919102096@smartinternz.com
- Connection Status: Connected
- Connection Time: Nov 24, 2022 11:41 AM
- Client Address: 157.49.150.32
- SecureToken: [redacted]

The screenshot shows the IBM Cloud developer console. The top navigation bar includes 'Catalog', 'Manage', and 'Nisha D's Account'. The main content area displays the details of a resource named 'Node RED UTBYX 2022-11-23'. The details section includes the App URL, Source, Resource group, Deployment target, and Created date. The Services section shows the Cloudant service. The Deployment Automation section shows the Name, Location, Tool integrations, and Delivery Pipelines. The Getting started quickly section provides instructions on how to configure the app.

**Details:**

- App URL: <https://node-red-utbyx-2022-11-23.eu-gb.mybluemix.net>
- Source: <https://eu-gb.git.cloud.ibm.com/1919102096/NodeREDUTBYX...>
- Resource group: Default
- Deployment target: Node RED UTBYX 2022-11-23
- Created: 11/23/2022

**Services:**

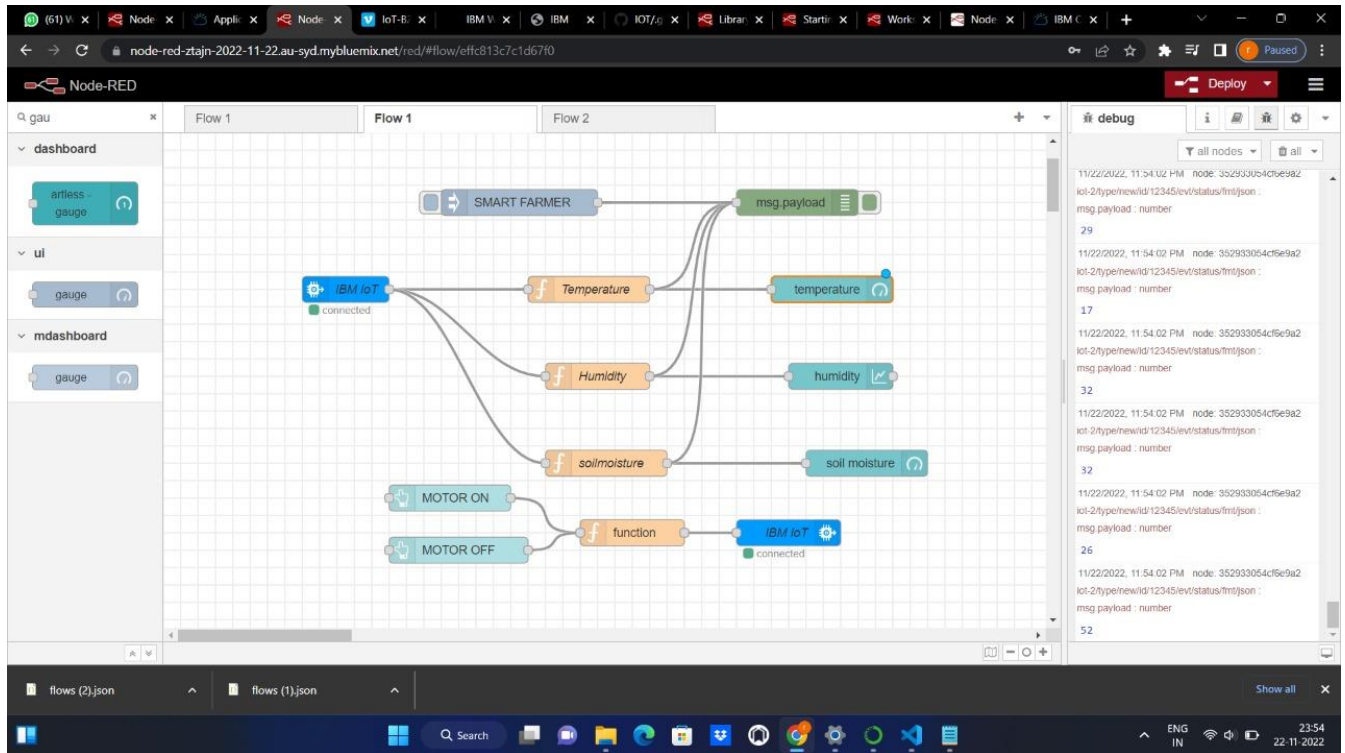
- Cloudant: Open dashboard, Documentation, API reference
- Credentials: [redacted]
- Connect existing services: +
- Create service: +

**Deployment Automation:**

- Name: NodeREDUTBYX2022-11-23
- Location: London
- Tool integrations: [redacted]
- Delivery Pipelines:
  - Name: pr-pipeline, Status: No stages detected
  - Name: ci-pipeline, Status: Success

**Getting started quickly:**

- Use the **Services** card to connect a service to your app. Select an existing service instance, or create a new one. [Learn more.](#)
- If you want to view the code before your app is deployed, click **Download code** to obtain the .zip file.
- Click **Deploy your app** in the **Deployment Automation** card to select the deployment target and configure the Continuous Delivery service. The deployment begins automatically.
- After the deployment begins, you can view the status of the deployment, modify your app, view your repo, or view the



```
File Edit Selection View Go Run Terminal Help
smf1.py - Visual Studio Code

C:\Users\rames> smf1.py
1 #IBM Watson IoT Platform
2 #pip install wiotp-sdk
3 import wiotp.sdk.device
4 import time
5 import random
6 myConfig = {
7     "identity": {
8         "orgid": "oldrpd",
9         "typeid": "new",
10        "deviceid": "12345"
11    },
12    "auth": {
13        "token": "12345678"
14    }
15 }
16
17 def myCommandCallback(cmd):
18     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
19     m=cmd.data['command']
20
21 client = wiotp.sdk.device.DeviceClient(config=myConfig, loghandlers=None)
22 client.connect()
23
24 while True:
25     temp=random.randint(-20,125)
26     hum=random.randint(0,100)
27     myData={'temperature':temp, 'humidity':hum}
28     client.publishEvent(eventId="status", msgformat="json", data=myData, qos=0, onPublish=None)
29     print("Published data Successfully: %s", myData)
30     client.commandCallback = myCommandCallback
31     time.sleep(2)
32     client.disconnect()
```

