

SPRINT-2

Date	5 November 2022
Team ID	PNT2022TMID46387
Project Name	Personal Assistance for Seniors Who Are Self Reliant

TASK :-

To create a device in the IOT Watson Platform, Workflow for IOT scenarios using Node-RED.

DESCRIPTION: -

- ❖ We have used **IoT Watson platform** for the creation of IoT device.
- ❖ The web application is built using **Node-RED** for collecting the medicine details from the users.
- ❖ We have used the **cloudant DB** for storing the collected data.
- ❖ The web application will send the medicine details to the created IoT device.
- ❖ The IoT device on receiving the details, it make use of TTS to remind the user about the medicine.
- ❖ By using **TTS** (Text to Speech) service from the IBM platform, the medicinal information will be notified to the users in the form of voice commands.
- ❖ Following are the screenshots that demonstrate the device creation and workflow of the IoT scenarios.

1) IBM WATSON – DEVICE CREATION

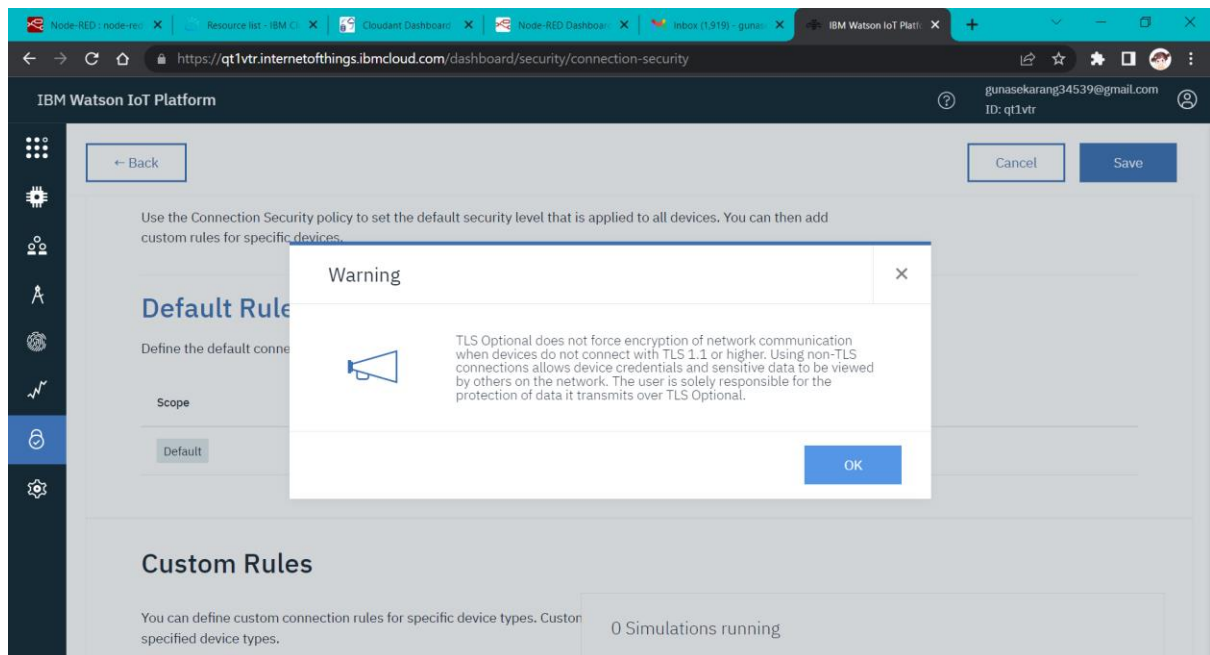
DEVICE TYPE CREATION:

The screenshot shows the 'Device Types' creation page in the IBM Watson IoT Platform. The browser address bar indicates the URL: `https://qt1vtr.internetofthings.ibmcloud.com/dashboard/devices/types/add`. The page header shows the user's email as `gunasekarang34539@gmail.com` and ID as `qt1vtr`. The main content area is titled 'Device Information' and includes a description: 'Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.' The form has three fields: 'Type' with a dropdown menu showing 'Device' and 'Gateway' options, 'Name' with the value 'loginsignup', and 'Description' which is empty. A note below the 'Name' field states: 'The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use.' At the bottom right of the form are 'Cancel' and 'Next' buttons. The page footer shows 'Device Types' and '0 Simulations running'.

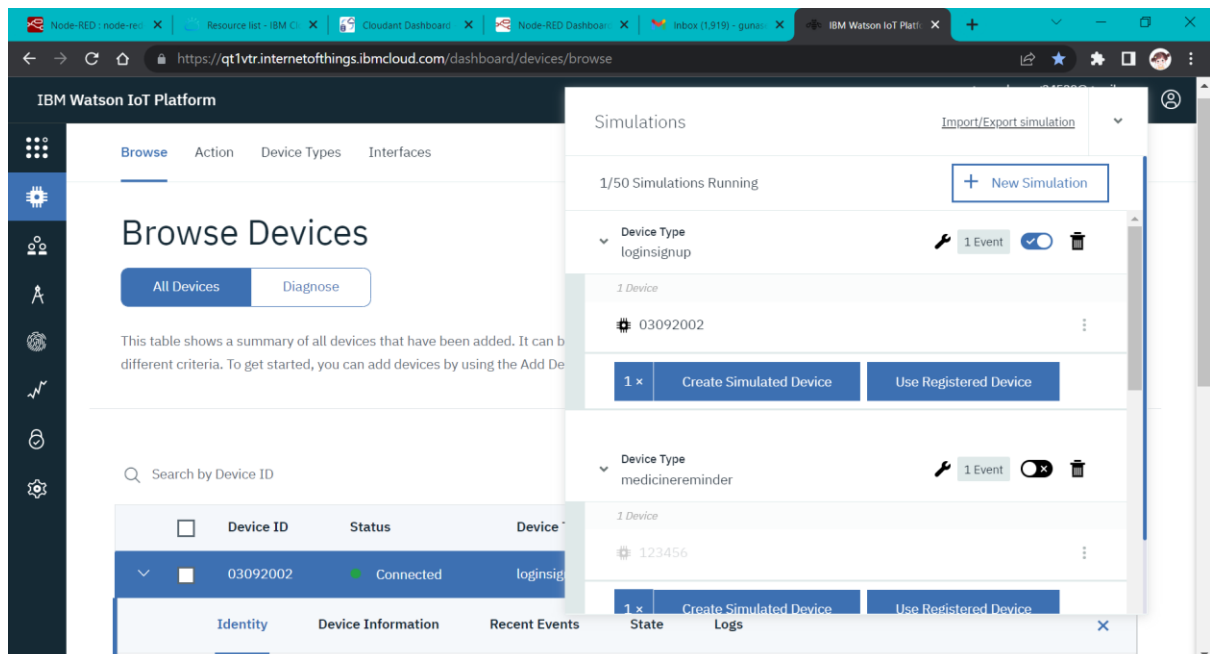
DEVICE CREATION:

The screenshot shows the 'Device Creation' page in the IBM Watson IoT Platform. The browser address bar indicates the URL: `https://qt1vtr.internetofthings.ibmcloud.com/dashboard/devices/browse/add`. The page header shows the user's email as `gunasekarang34539@gmail.com` and ID as `qt1vtr`. The main content area is titled 'Device Information' and includes a description: 'Select a device type for the device that you are adding and give the device a unique ID.' The form has two fields: 'Device Type' with the value 'loginsignup' and 'Device ID' with the value '03092002'. At the bottom right of the form are 'Cancel' and 'Next' buttons. The page footer shows 'Browse Devices' and '0 Simulations running'.

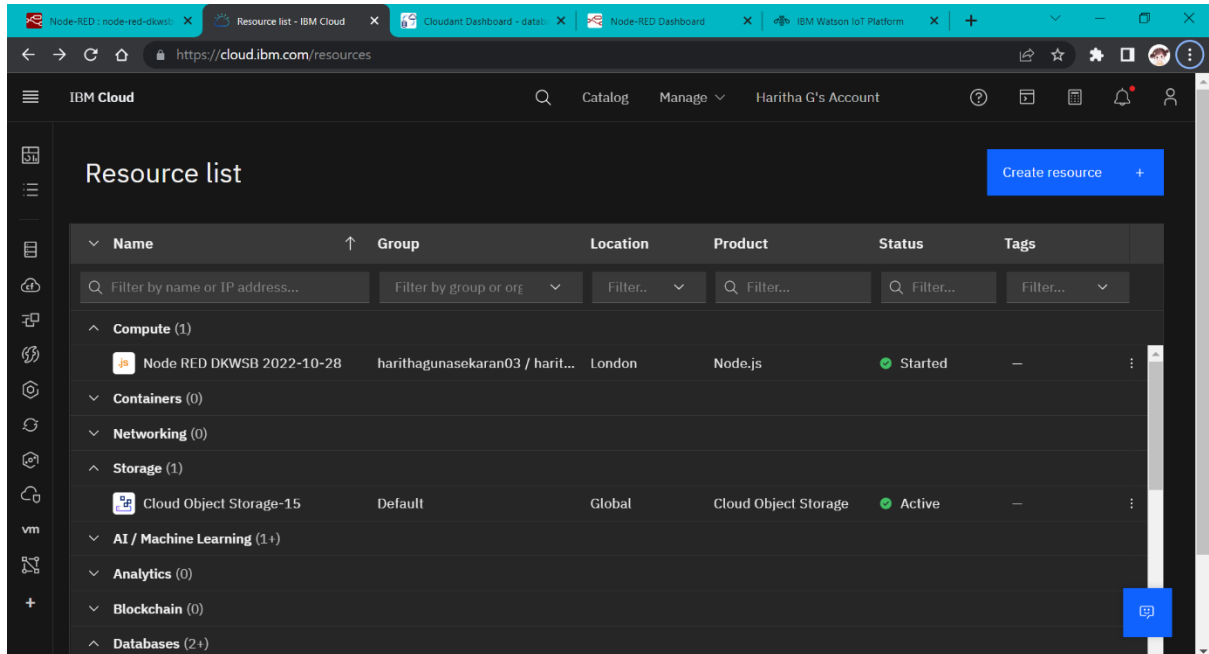
CONFIGURE SECURITY POLICY:



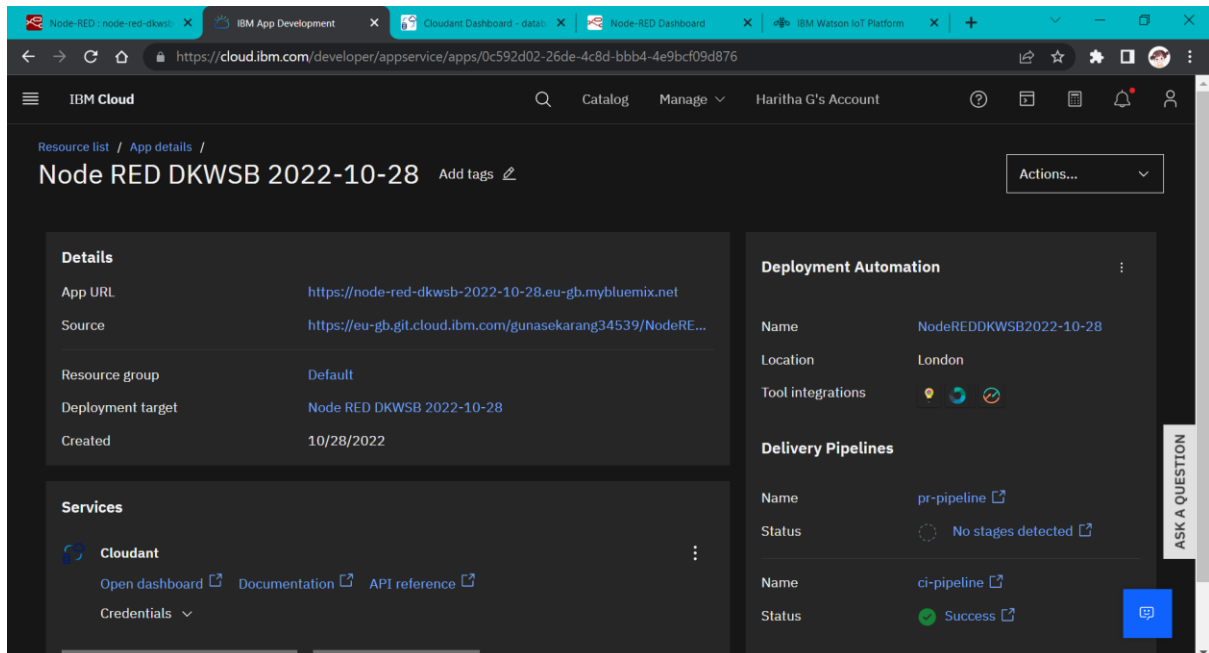
SIMULATE IOT DEVICE:

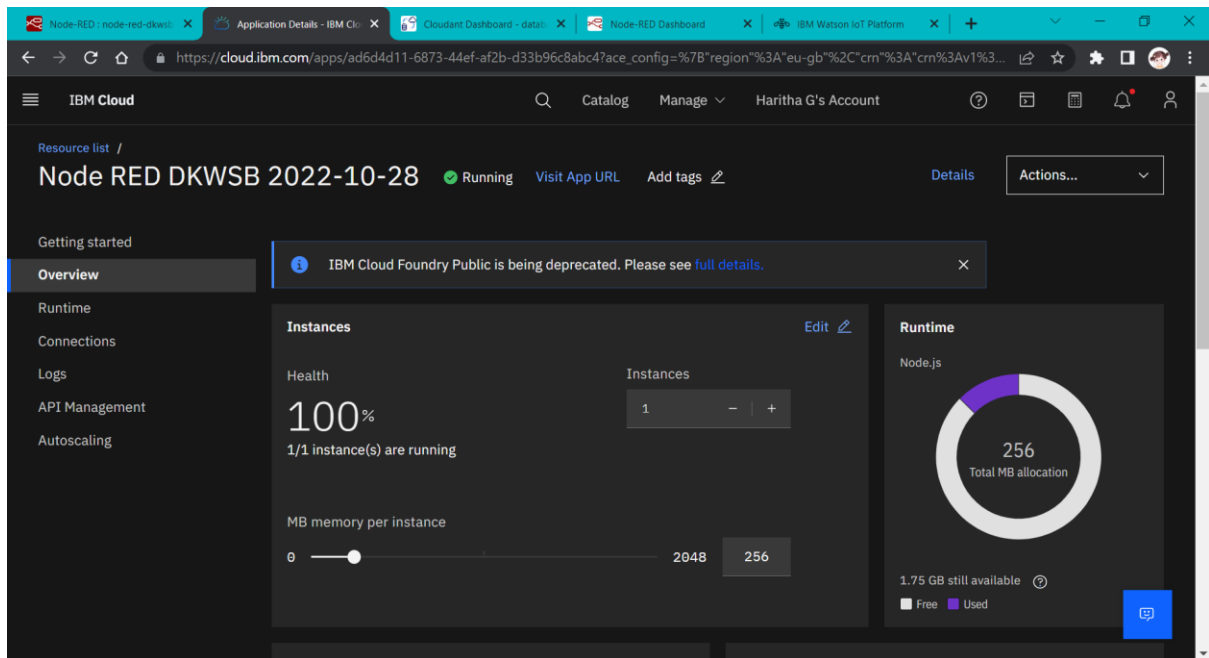


2) NODE-RED CREATION:



DEPLOYING NODE-RED WEB APP:





NODE-RED FLOW EDITOR:

The screenshot shows the Node-RED flow editor interface on IBM Cloud. The browser address bar displays the URL: <https://node-red-dkwsb-2022-10-28.eu-gb.mybluemix.net>. The page features a large red banner with the text 'Node-RED' and 'Flow-based programming for the Internet of Things'. Below the banner, there is a section with the following text:

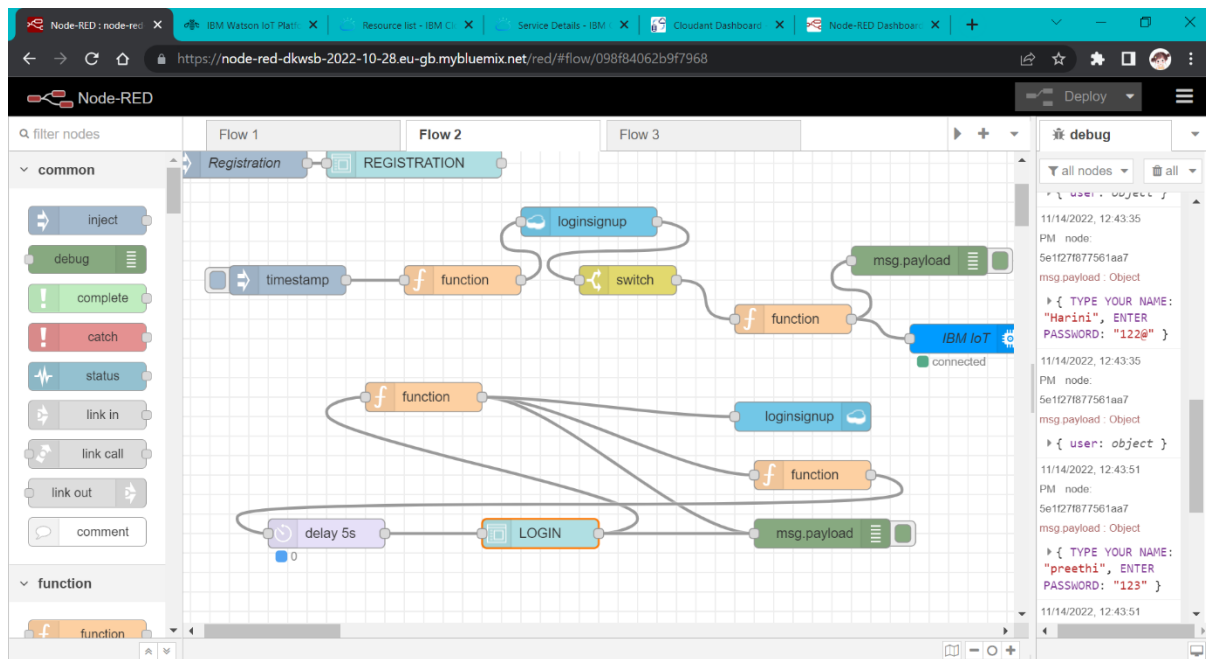
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at nodered.org.

On the right side, there is a button labeled 'Go to your Node-RED flow editor' and a link labeled 'Learn how to customise Node-RED'.

CREATION OF NODES FOR THE WORKFLOW:

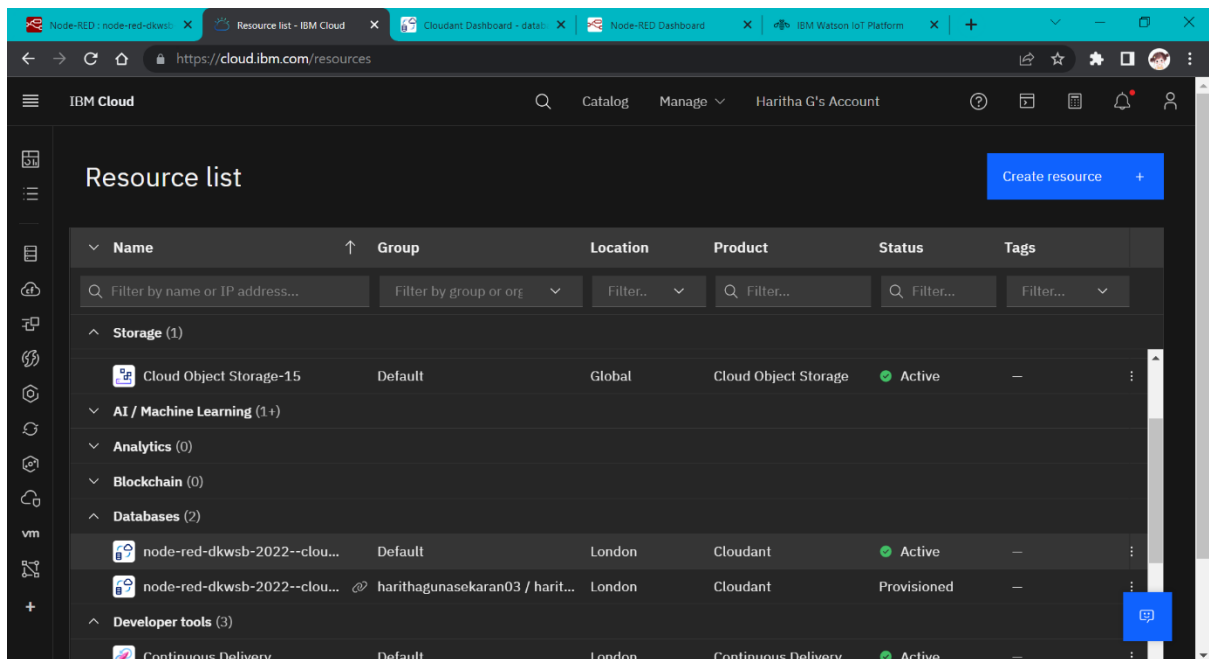


3) CLOUDANT CREATION:

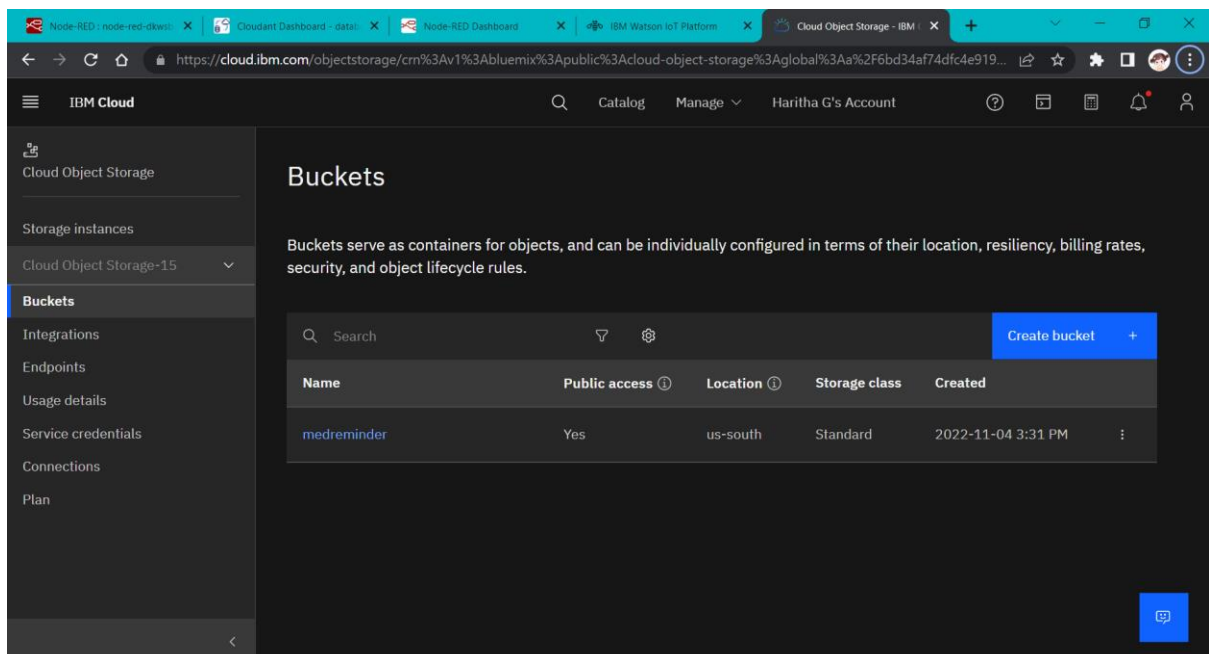
The screenshot shows the IBM Cloud console interface. The top navigation bar includes the IBM Cloud logo, a search bar, and the user's account name 'Haritha G's Account'. The main content area displays the details of a Cloudant database instance named 'node-red-dkwsb-2022--cloudant-1666974514582'. The instance is in the 'Active' state. The 'Overview' tab is selected, showing deployment details such as CRN, Location, External endpoint, and Authentication methods. A 'Launch Dashboard' button is visible in the top right corner.

Deployment details	
CRN	crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/6bd34af74dfc4e919b494a89d99a1279:2f422b8e-4c41-4348-b9fe-c1bb7cc9ef7c::
Location	London
External endpoint	https://e0ca5cfe-2984-4eb5-b604-dc654ceb3930-bluemix.cloudant.com
External endpoint (preferred)	https://e0ca5cfe-2984-4eb5-b604-dc654ceb3930-bluemix.cloudantnosqldb.appdomain.cloud
Authentication methods	IBM Cloud IAM Cloudant credentials Migrate to IAM Only

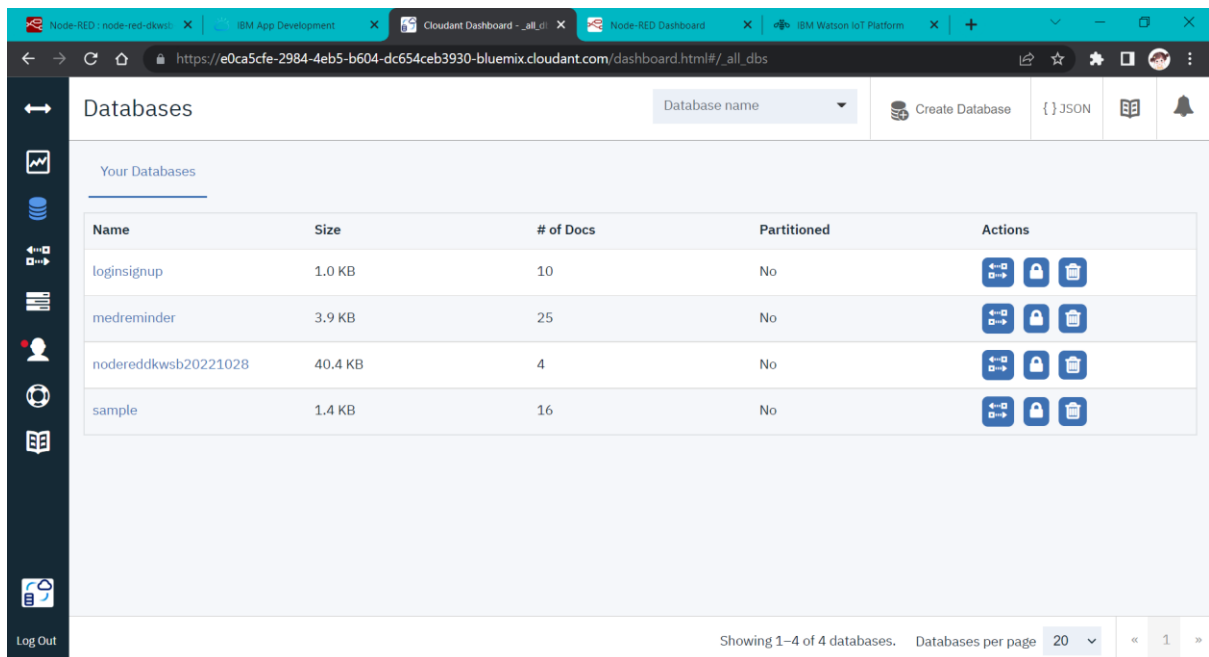
BUCKET CREATION:



❖ Here we created an bucket in as cloud object storage for our project.



4) DATABASE CREATION:



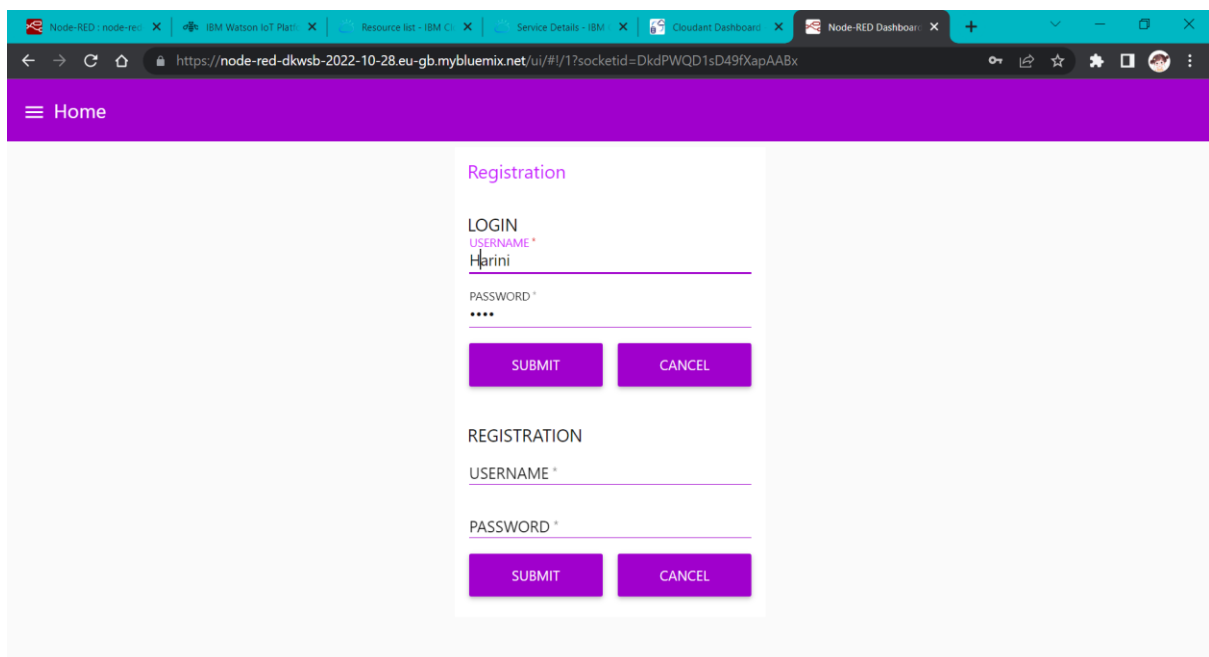
The screenshot shows the Cloudant Databases dashboard. The header includes a navigation bar with icons for databases, documents, and settings. The main content area is titled 'Your Databases' and contains a table with the following data:

Name	Size	# of Docs	Partitioned	Actions
loginsignup	1.0 KB	10	No	[Icons for edit, delete, and other actions]
medreminder	3.9 KB	25	No	[Icons for edit, delete, and other actions]
nodreddkwsb20221028	40.4 KB	4	No	[Icons for edit, delete, and other actions]
sample	1.4 KB	16	No	[Icons for edit, delete, and other actions]

At the bottom of the dashboard, it says 'Showing 1-4 of 4 databases. Databases per page 20'.

FINAL EXECUTION:

- ❖ When the user enter the login credentials, it get stored in db.
- ❖ After successfull login user get redirected to the home scren.



The screenshot shows the application's home screen. The header is a purple bar with the text 'Home'. The main content area is white and contains two forms: 'LOGIN' and 'REGISTRATION'. The 'LOGIN' form has fields for 'USERNAME' (with the value 'harini') and 'PASSWORD' (with the value '****'). Below these fields are 'SUBMIT' and 'CANCEL' buttons. The 'REGISTRATION' form has fields for 'USERNAME' and 'PASSWORD'. Below these fields are 'SUBMIT' and 'CANCEL' buttons.

❖ The data is getting coming in the IOT device platform.

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 various icons for navigation. The main content area displays a table of devices. The first device, '03092002', is highlighted, and its 'Recent Events' tab is active. This tab shows a live stream of data with the following table:

Event	Value	Format	Last Received
event_1	{"username":"harini","password":"12@"}	json	a few seconds ago

Below the 'Recent Events' tab, a list of devices is shown, including '12345' (Disconnected, UltraSonDistance) and '123456' (Disconnected, medicin). A status message '1 Simulation running' is visible at the bottom right of the device list.

❖ Now, we can see the data is getting stored in DB.

The screenshot shows the IBM Watson IoT Platform database view. The top navigation bar includes 'Node-RED', 'IBM Watson IoT Platform', 'Resource list', 'Service Details', 'Cloudant Dashboard', and 'Node-RED Dashboard'. The main content area displays a JSON document for the device 'loginsignup' with ID '6999da3f69f9e3d0fb307bf60fc9f28b'. The document is shown in a code editor with the following JSON structure:

```
1 {
2   "_id": "6999da3f69f9e3d0fb307bf60fc9f28b",
3   "_rev": "1-dc21b1d2ad91369e8181ae4efe1a6680",
4   "payload": {
5     "user": {
6       "TYPE YOUR NAME": "Harini",
7       "ENTER PASSWORD": "122@"
8     }
9   },
10   "socketId": "C9K09UYaqzLwHESHAABz"
11 }
```

The interface includes a 'Save Changes' button, a 'Cancel' button, and an 'Upload Attachment' button. The document is displayed in a table format with a 'JSON' tab selected.

RESULT:

At the end of the sprint 2, we created the software for our project and tested successfully.