

CREATE AND CONFIGURE IBM CLOUD SERVICES

CREATE A NODE – RED

Team ID	PNT2022TMID21246
Project Name	Smart Farmer – IoT based smart farming application

Team Leader: LAVANYA R

Team Member: ABIRAMIPRIYA J

Team Member: KALAISELVI S

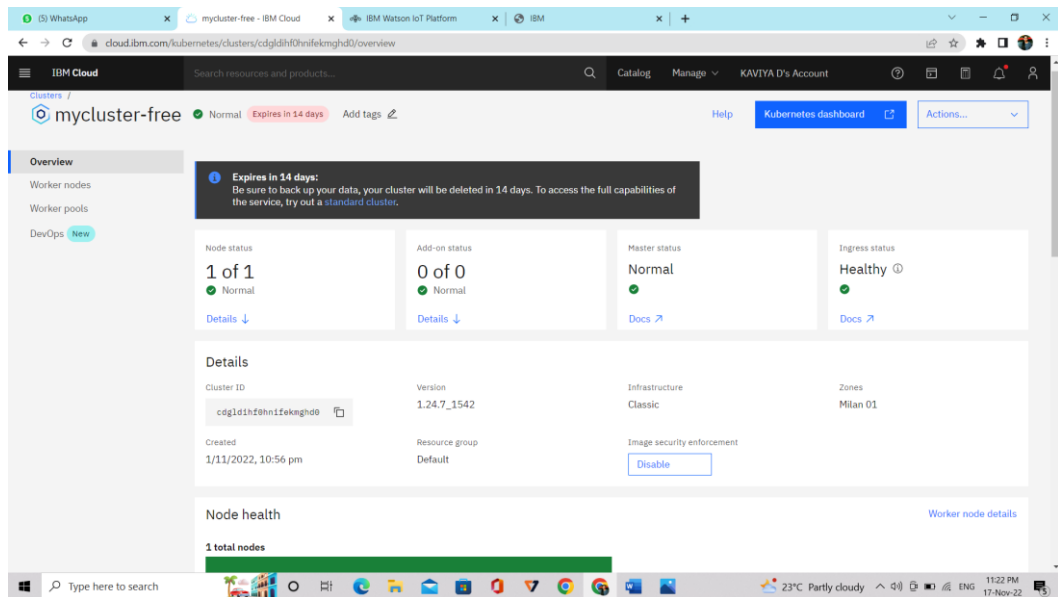
Team Member: KAVIYA D

LOGIN TO THE IBM CLOUD ACCOUNT AND CREATE AN APP:

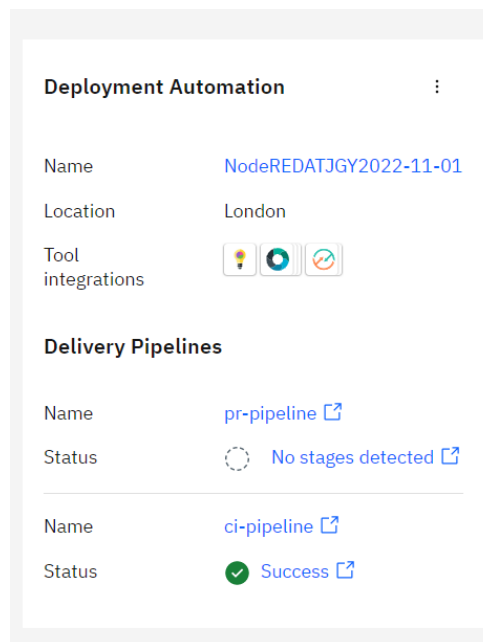
The screenshot displays the IBM Cloud console interface for a specific application. The main header shows the application name 'Node RED ATJGY 2022-11-01' and an 'Add tags' button. The left sidebar contains navigation options like 'Resource list', 'App details', 'Catalog', 'Manage', and 'KAVIYA D's Account'. The main content area is divided into several sections: 'Details' showing app URL, source, resource group, deployment target, and creation date; 'Services' with links to Cloudant, Open dashboard, Documentation, and API reference; 'Deployment Automation' showing the name, location, tool integrations, and delivery pipelines; and a 'Getting started quickly' sidebar with a list of steps for configuring the app. The bottom of the screen shows a Windows taskbar with various application icons and system information.

CREATE A CLUSTER IN KUBERNETES INORDER TO DEPLOY THE APP:

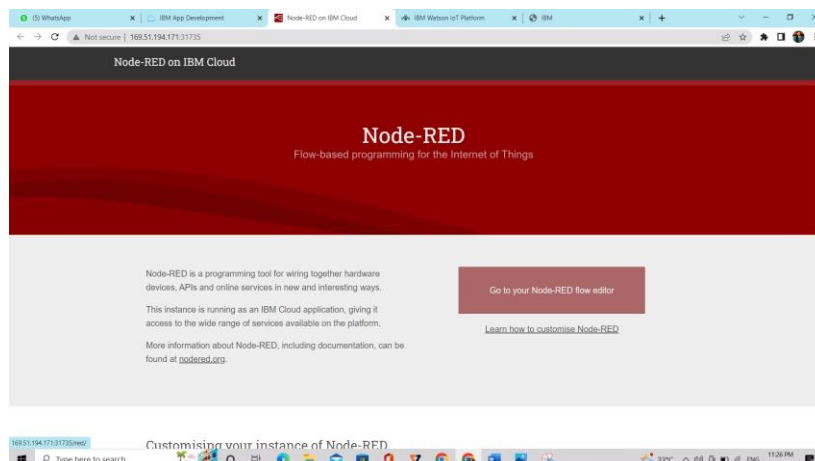
The screenshot shows the IBM Cloud console's 'Kubernetes clusters' page. The left sidebar has a navigation menu with 'Kubernetes', 'Clusters', 'Reservations', 'Helm catalog', and 'Container Registry'. The main content area displays a table of existing clusters. The table has columns for Name, State, Location, Worker count, Created, Version, and Infrastructure. A single cluster named 'mycluster-free' is listed with a 'Normal' state, located in 'Paris 01', with 1 worker, created on 11/11/2022, and version 1.24.7_1542. A 'Create cluster' button is visible in the top right corner. The bottom of the screen shows a Windows taskbar with various application icons and system information.



DEPLOY THE APP USING THE CLUSTER INFORMATION:



CLICK THE APP URL AND NODE RED WILL OPEN:



TESTING USING NODE RED:

The screenshot shows the Node-RED web interface in a browser. The left sidebar contains a palette of nodes categorized into 'common' and 'function'. The main workspace, labeled 'Flow 1', contains a flow with two nodes: an 'inject' node (labeled 'AKKL TEST') and a 'msg.payload' node. The 'inject' node is connected to the 'msg.payload' node. The right sidebar shows the 'debug' console, which displays a series of log messages. Each message is a JSON object with a timestamp, a node ID, and a message payload. The messages are:

```
{ "time": "11/8/2022, 12:07:55 AM", "node": "2f2543a.0d0d98", "msg": { "payload": "Hello IBM" } }
```

 The browser's address bar shows the URL: `169.51.194.171:31735/red/#flow/ebfc732bd80cc0db`. The bottom status bar shows the system clock as 12:10 AM on 08-Nov-22.

INSTALL IBM IOT NODE IN THE MANAGE PALLETTE AND USING THE NODE ALONG WITH THE DEVICE CREATED IN THE IBM WATSON ACCOUNT:

The screenshot shows the Node-RED web interface with the 'manage' tab selected in the left sidebar. The 'function' category is expanded, showing various nodes. The 'IBM IoT' node is highlighted. The main workspace, labeled 'Flow 1', contains a flow with two nodes: an 'IBM IoT' node (labeled 'connected') and a 'msg.payload' node. The 'IBM IoT' node is connected to the 'msg.payload' node. The right sidebar shows the 'debug' console, which displays log messages. The messages are:

```
{ "time": "11/8/2022, 8:54:58 PM", "node": "2f2543a.0d0d98", "msg": { "payload": { "randomNumber": 64, "temp": 96, "hum": 92 } } }
```

 and

```
{ "time": "11/8/2022, 8:55:07 PM", "node": "2f2543a.0d0d98", "msg": { "payload": { "randomNumber": 26, "temp": 107, "hum": 79 } } }
```

 The browser's address bar shows the URL: `169.51.194.171:31735/red/#flow/ebfc732bd80cc0db`. The bottom status bar shows the system clock as 8:55 PM on 08-Nov-22.

OUTPUT OF THE DEVICE AFTER CONNECTING IT WITH NODE RED:

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area shows a table of devices. The first device listed is '1234', which is 'Disconnected'. Below the table, a modal window titled 'Identity' is open, showing 'Recent Events' for the selected device. The events are listed in a table with columns: Event, Value, Format, and Last Received. The events are 'eventflow' with values like '{\"randomNumber\":1,\"temp\":102,\"hum\":98}' and '{\"randomNumber\":100,\"temp\":107,\"hum\":74}', all in 'json' format. A status box at the bottom right indicates '1 Simulation running'. The bottom of the image shows a Windows taskbar with various application icons and system information like '26°C Mostly cloudy' and '7:56 PM 08-Nov-22'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	abcd	Device	Nov 8, 2022 7:39 PM	

Event	Value	Format	Last Received
eventflow	{\"randomNumber\":1,\"temp\":102,\"hum\":98}	json	a few seconds ago
eventflow	{\"randomNumber\":100,\"temp\":107,\"hum\":74}	json	a minute ago
eventflow	{\"randomNumber\":45,\"temp\":108,\"hum\":87}	json	a minute ago
eventflow	{\"randomNumber\":87,\"temp\":108,\"hum\":93}	json	2 minutes ago

1 Simulation running