

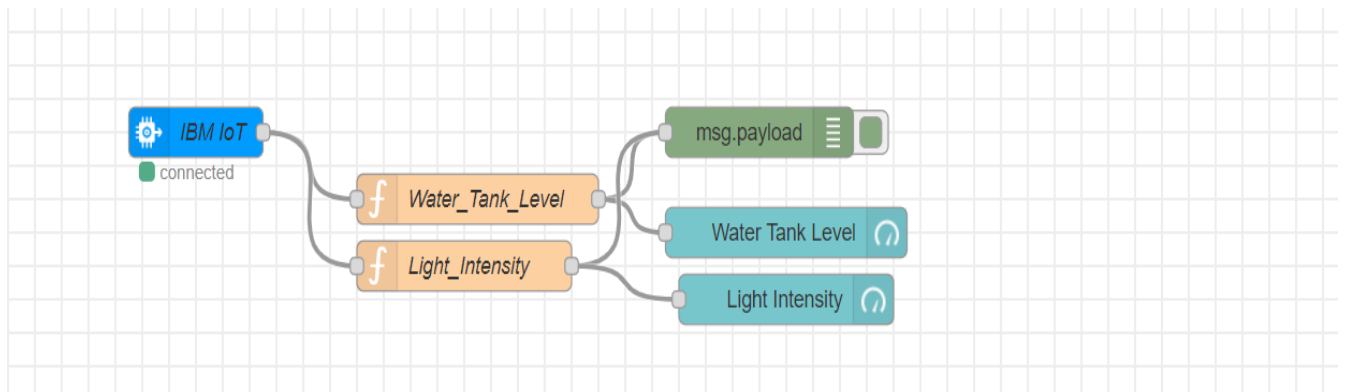
Assignment 3

Develop a code to upload the water tank level and light intensity values to the IBM IoT platform and visualize them in the web application.

IoT Device Id and type:

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
>	12345	Disconnected	VITharish	Device	Jul 8, 2021 6:49 PM	
>	Assignment3	Disconnected	VITharish	Device	Jul 14, 2021 4:00 PM	

After opening node red app and then go to Node red flow editor:



IBM IoT properties:

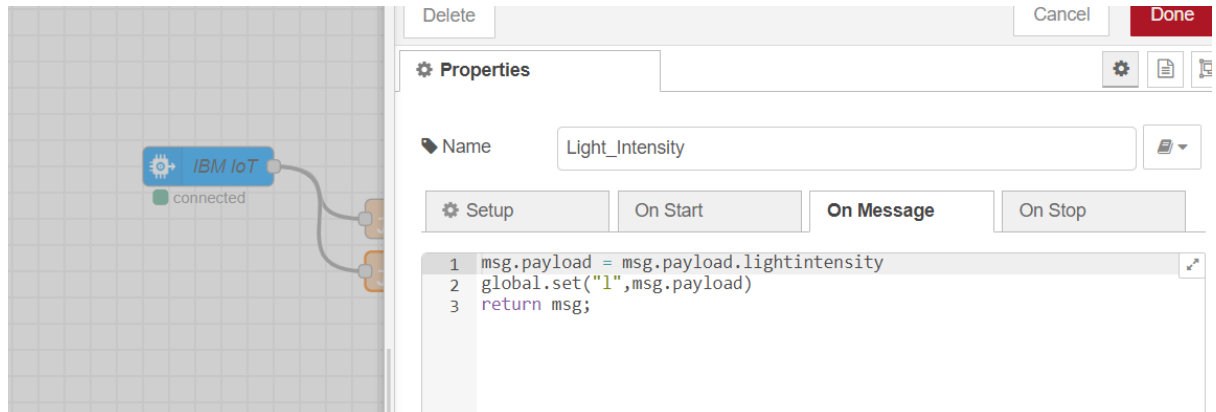
Properties

- Authentication: API Key
- API Key: Vit IoT
- Input Type: Device Event
- Device Type: All or VITharish
- Device Id: All or Assignment3
- Event: All or +
- Format: All or json
- QoS: 0
- Name: IBM IoT
- Service: registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status.

Getting Data of Water tank level and light intensity from Python code:

- `msg.payload = msg.payload.WaterTankLevel`
- `global.set("w",msg.payload)`
- `return msg;`



Python code:

```
import wiotp.sdk.device
```

```
import time
```

```
import random
```

```
myConfig = {
```

```
    "identity": {
```

```
        "orgId": "aannkh",
```

```
        "typeId": "VITharish",
```

```
        "deviceId": "Assignment3"
```

```
    },
```

```
    "auth": {
```

```
        "token": "9381628451"
```

```
    }
```

```
}
```

```
def myCommandCallback(cmd):
```

```
    print("Message received from IBM IoT Platform: %s" %  
cmd.data['command'])
```

```
    m=cmd.data['command']
```

```
    if(m == "lighton"):
```

```
        print("....Light is ON....")
```

```
    elif (m == "lightoff"):
```

```
        print("....Light is OFF....")
```

```
    print()
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
```

```
client.connect()
```

```
while True:
```

```
    temp=random.randint(0,100)
```

```
    hum=random.randint(1,100)
```

```
    myData={'WaterTankLevel':temp, 'lightintensity':hum}
```

```
    client.publishEvent(eventId="status", msgFormat="json", data=myData,  
qos=0, onPublish=None)
```

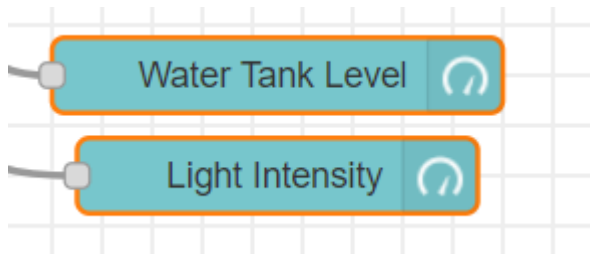
```
    print("Published data Successfully: %s", myData)
```

```
    client.commandCallback = myCommandCallback
```

```
    time.sleep(2)
```

```
client.disconnect()
```

To represent the data in web application using Dashboard flows in node red:

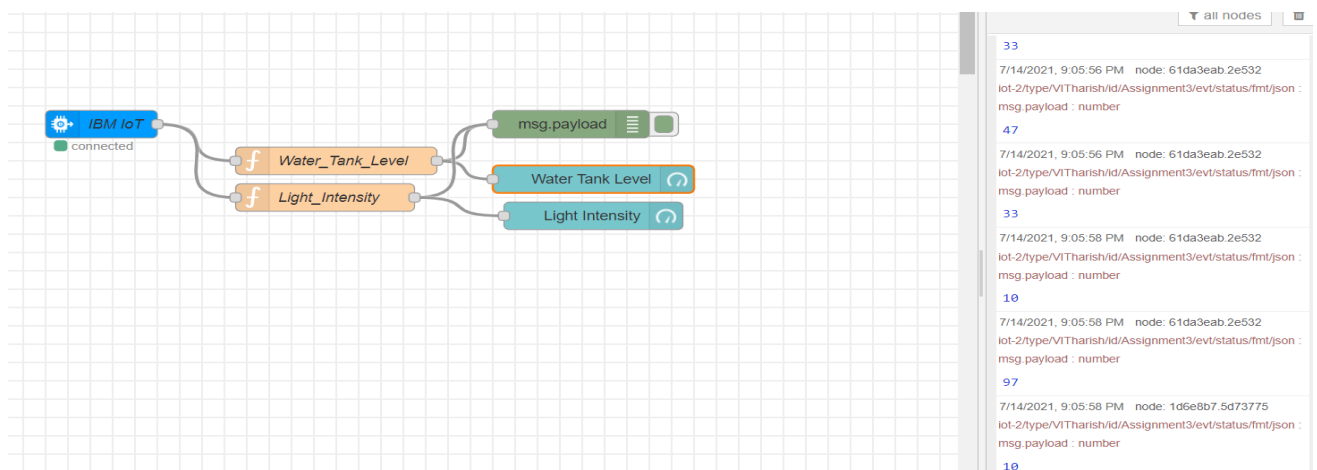


Properties of Gauge:

The image shows the 'Properties' dialog for a gauge node. The settings are as follows:

- Group:** [SmartHome] Home Data
- Size:** 11 x 3
- Type:** Level
- Label:** Water Tank Level
- Value format:** {{value}}
- Units:** %
- Range:** min 0, max 100
- Name:** (empty)

After deploy:



Opening Web application to see the Data:

Web browser link: [Node-RED Dashboard \(mybluemix.net\)](https://haridata.eu-gb.mybluemix.net/ui/#/0?socketid=6u1V78XP2SC5MAX_AAAZ)

The Python Data is sharing with web in realtime.

The image shows a Jupyter Assignment window on the left and a Node-RED Dashboard on the right. The Jupyter window displays a Python script and a list of published data points. The Node-RED Dashboard shows a 'Home Data' section with two gauges: 'Light Intensity' (Brightness) and 'Water Tank Level'.

Jupyter Assignment Code:

```
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
```

Published data Successfully:

- { 'WaterTankLevel': 64, 'lightintensity': 12 }
- { 'WaterTankLevel': 10, 'lightintensity': 20 }
- { 'WaterTankLevel': 19, 'lightintensity': 3 }
- { 'WaterTankLevel': 13, 'lightintensity': 63 }
- { 'WaterTankLevel': 23, 'lightintensity': 85 }
- { 'WaterTankLevel': 28, 'lightintensity': 62 }
- { 'WaterTankLevel': 90, 'lightintensity': 100 }
- { 'WaterTankLevel': 66, 'lightintensity': 9 }
- { 'WaterTankLevel': 36, 'lightintensity': 60 }
- { 'WaterTankLevel': 1, 'lightintensity': 87 }
- { 'WaterTankLevel': 11, 'lightintensity': 82 }
- { 'WaterTankLevel': 23, 'lightintensity': 14 }
- { 'WaterTankLevel': 2, 'lightintensity': 34 }
- { 'WaterTankLevel': 16, 'lightintensity': 48 }
- { 'WaterTankLevel': 15, 'lightintensity': 57 }
- { 'WaterTankLevel': 71, 'lightintensity': 48 }
- { 'WaterTankLevel': 57, 'lightintensity': 34 }
- { 'WaterTankLevel': 83, 'lightintensity': 89 }
- { 'WaterTankLevel': 51, 'lightintensity': 53 }
- { 'WaterTankLevel': 50, 'lightintensity': 3 }

Node-RED Dashboard:

- Light Intensity (Brightness):** 3
- Water Tank Level:** 50%

The image shows a Node-RED Dashboard window with a 'Home Data' section. The gauges show updated values: 'Light Intensity' (Brightness) is 73 and 'Water Tank Level' is 63%.

Home Data:

- Light Intensity (Brightness):** 73
- Water Tank Level:** 63%