

PRACTICAL-7

AIM

Demonstrate message publish & subscribe mechanism of MQTT protocol using node red.

THEORY

Node Red:

- Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
- It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.
- Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click.
- JavaScript functions can be created within the editor using a rich text editor.
- A built-in library allows you to save useful functions, templates or flows for re-use.
- The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.
- With over 225,000 modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities.
- The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others.
- An online flow library allows you to share your best flows with the world.

MQTT:

- MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT).
- It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.
- MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.
- Features of MQTT:
 - Lightweight and Efficient
 - MQTT clients are very small, require minimal resources so can be used on small microcontrollers. MQTT message headers are small to optimize network bandwidth.
 - Bi-directional Communications
 - MQTT allows for messaging between device to cloud and cloud to device. This makes for easy broadcasting messages to groups of things.
 - Scale to Millions of Things
 - MQTT can scale to connect with millions of IoT devices.

PRACTICAL

- We can install node red on windows by following command.
- `npm install -g --unsafe-perm node-red`
- Then we can run “node-red” command in cmd to start the node-red.

```

Welcome to Node-RED
=====

19 Mar 09:59:11 - [info] Node-RED version: v1.2.9
19 Mar 09:59:11 - [info] Node.js version: v14.15.4
19 Mar 09:59:11 - [info] Windows_NT 10.0.19042 x64 LE
19 Mar 09:59:12 - [info] Loading palette nodes
19 Mar 09:59:31 - [info] Settings file : C:\Users\jilsa\.node-red\settings.js
19 Mar 09:59:31 - [info] Context store : 'default' [module=memory]
19 Mar 09:59:31 - [info] User directory : C:\Users\jilsa\.node-red
19 Mar 09:59:31 - [warn] Projects disabled : editorTheme.projects.enabled=false
19 Mar 09:59:31 - [info] Flows file : C:\Users\jilsa\.node-red\flows_DESKTOP-SSUT150.json
19 Mar 09:59:31 - [info] Creating new flow file
19 Mar 09:59:31 - [warn]

-----

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

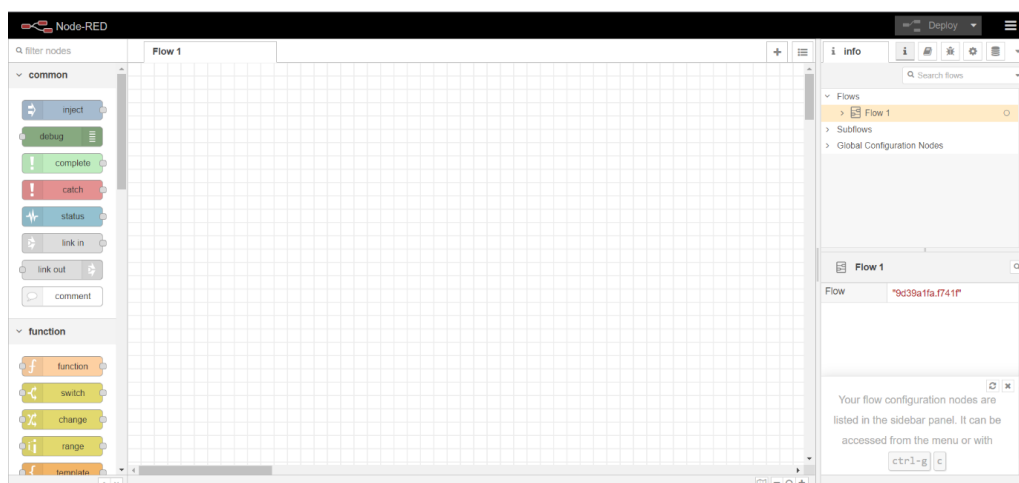
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.

-----

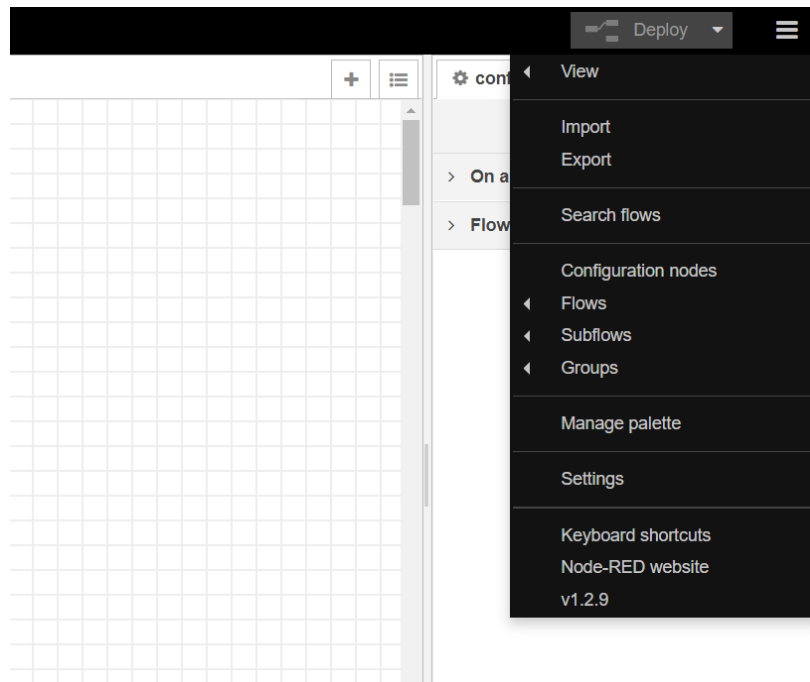
19 Mar 09:59:31 - [info] Server now running at http://127.0.0.1:1880/
19 Mar 09:59:31 - [info] Starting flows
19 Mar 09:59:31 - [info] Started flows

```

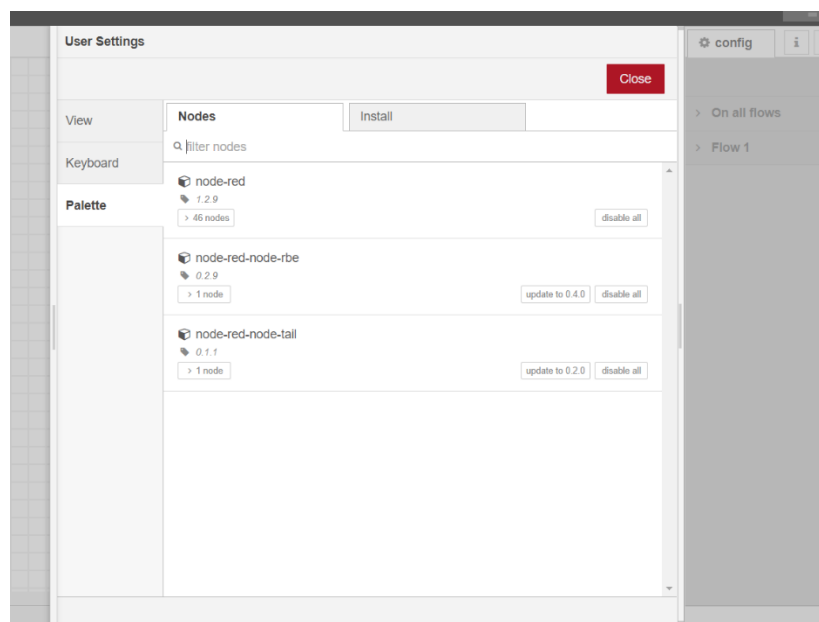
- It will give us an IP address to use web-based node-red.



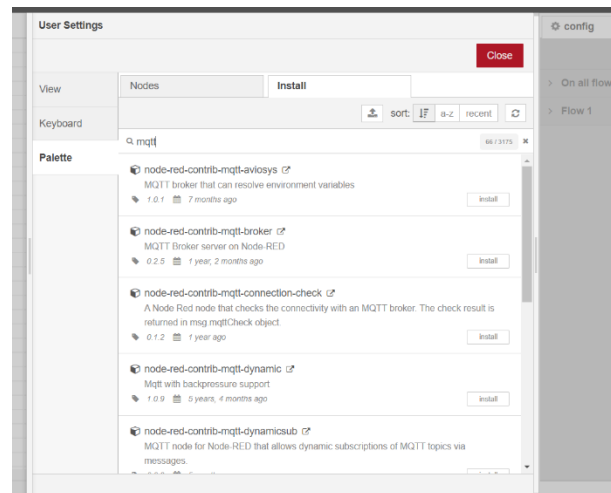
- First, we will install broker for MQTT protocol.
- For that, we will go to menu situated at top right corner.
- We will see Manage Palette option there.



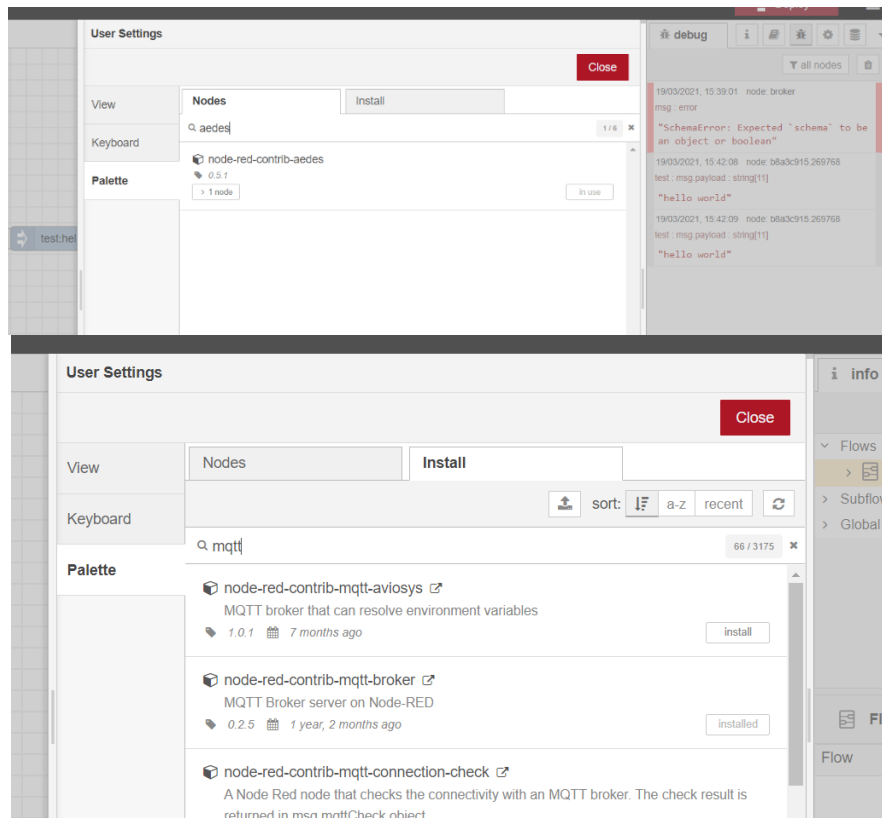
- By clicking on it, User settings will be opened.



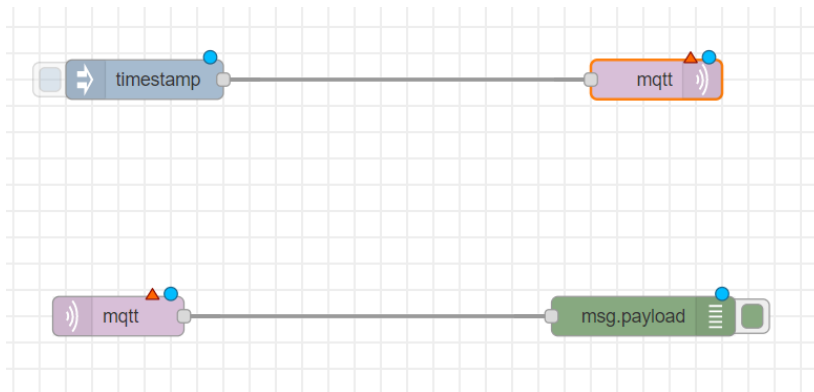
- We go to Install tab and search MQTT.



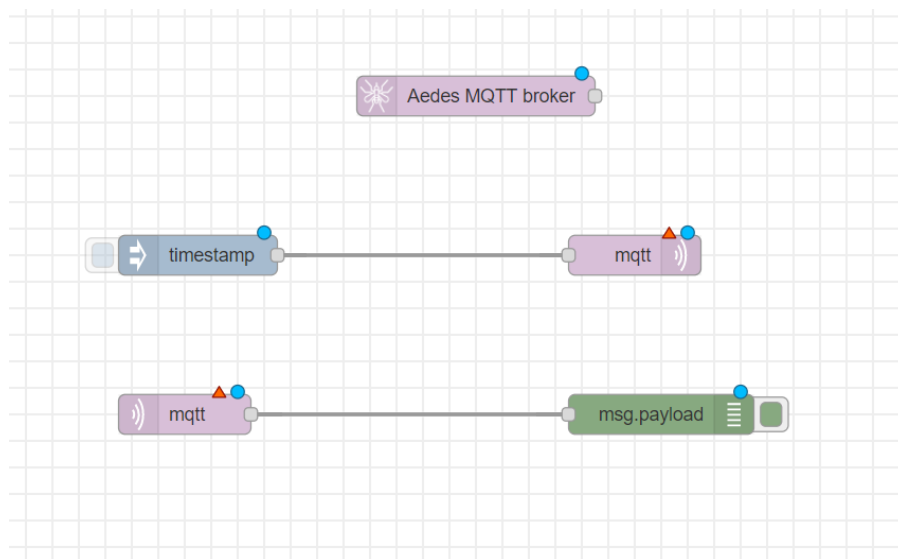
- We will install “node-red-contrib-mqtt-broker” or “node-red-contrib-aedes”



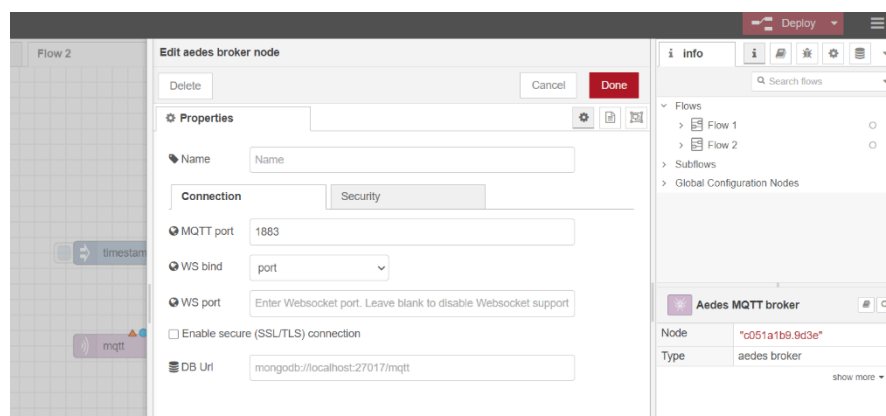
- Then we will add 4 nodes.
 1. Inject node, which will be renamed as Timestamp.
 2. Debug node, which will be renamed as msg.payload.
 3. MQTT in
 4. MQTT out
- We will connect timestamp node to MQTT out and debug node to MQTT in.



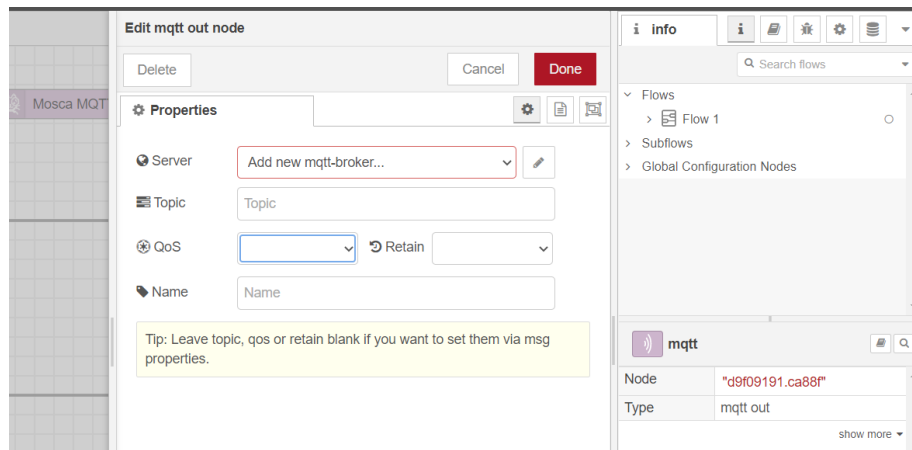
- Then we will add MOSCA MQTT broker or AEDES MQTT broker.



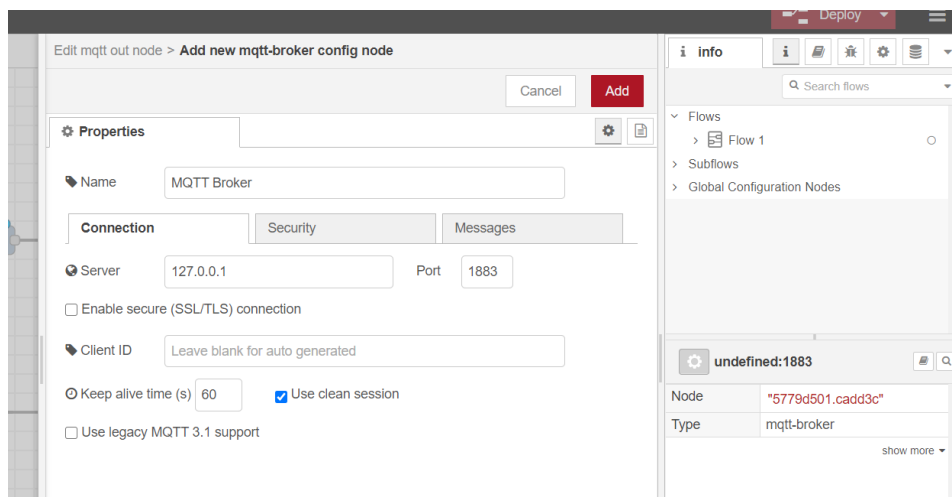
- We can double click on node to see and change their properties.



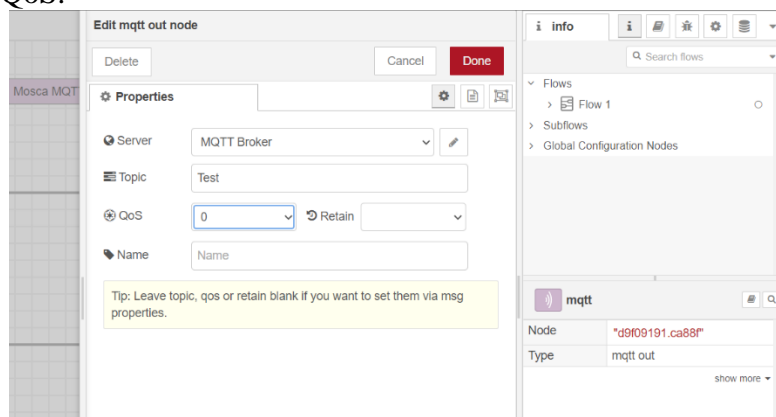
- We will first configure MQTT out node.



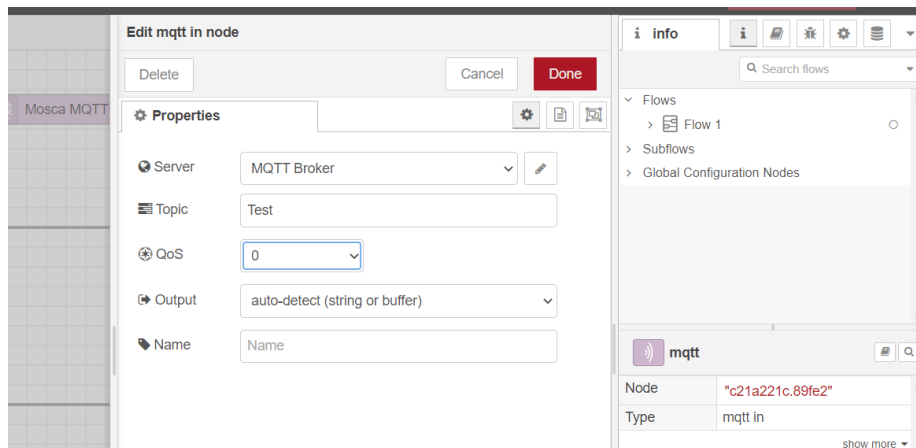
- We will add new mqtt-broker. So we click on the button beside it.
- We will add name and server IP address.
- We can add additional settings like security too if we want.



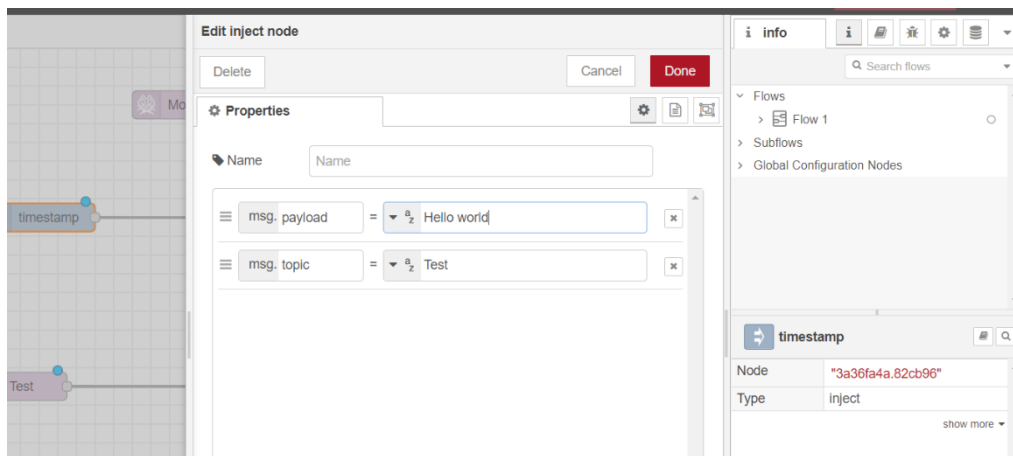
- When we click add, the server will be created and we will fill a couple of fields there like topic and QoS.



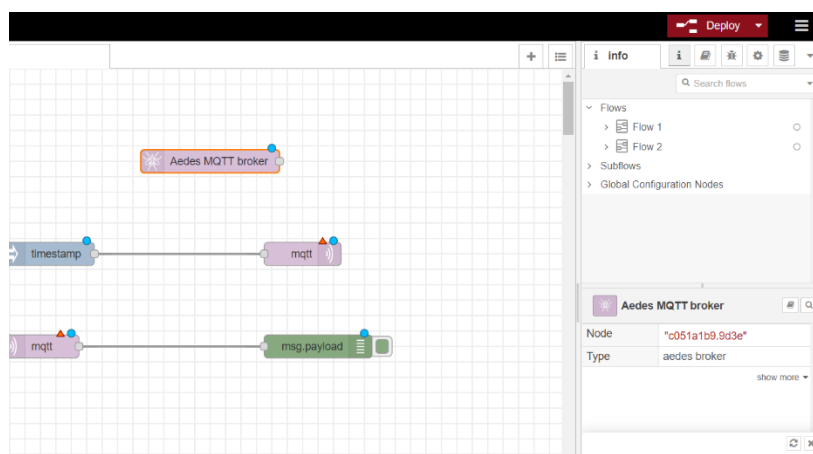
- We will provide same configuration for MQTT in node but we don't need to create MQTT Broker again.



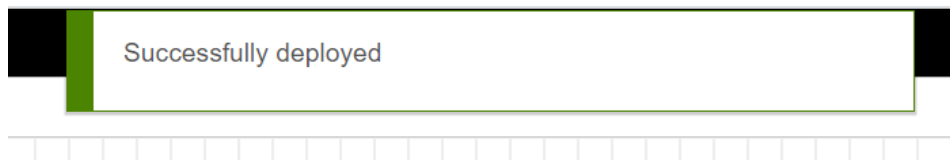
- We will give input string in timestamp node. We change the timestamp to string of “Hello world” and give it the topic name same as MQTT nodes.



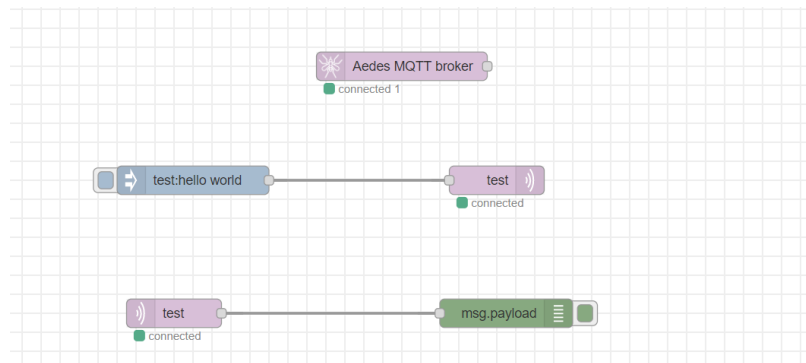
- Now we can deploy our model by “Deploy” button on top right corner.



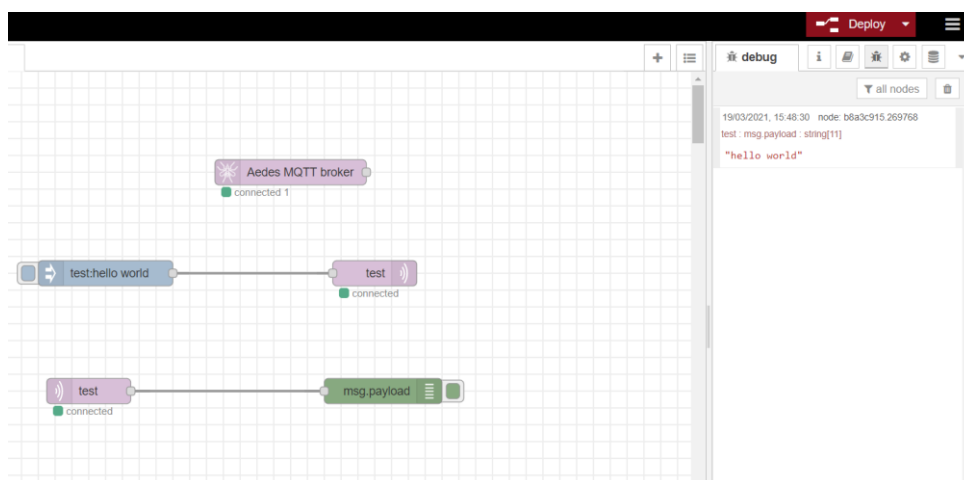
- We can see “Successfully deployed” message.



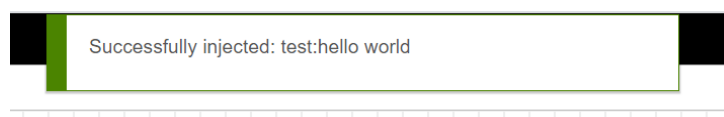
- And in a moment, we will be able to see connected status if there is no error.



- In debug console, which can be opened from right panel, we can see the output received by debug node after clicking inject node to send the text.



- We can see the status on top too.



CONCLUSION

In this practical, we learned about Node red and MQTT. We implemented the MQTT connection using Node red.