

# AWS Device Testing

Node Red

Team Emertxe



# AWS Device Testing

## Node Red

The screenshot shows the AWS IoT console interface. On the left, there is a navigation menu with options like Monitor, Onboard, Manage, Secure, Act, and Test. The main content area displays the 'Certificates' page, which includes a search bar and a list of certificates. One certificate is visible with the ID '7c72dc34-acf55c148...' and a status of 'ACTIVE'. A terminal window titled 'adil : bash — Konsole <2>' is overlaid on the console, showing the prompt 'adil@wisdom:~]'. A yellow callout box with the text 'Press Ctrl Atl t' and an arrow points to the terminal window.

# AWS Device Testing

## Node Red - Installation

```
user@user:~] sudo apt install npm
```

```
user@user:~] sudo install npm  
[sudo] password for user:
```

```
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed:  
0 upgraded, 74 newly installed, 1 to remove and 263 not upgraded  
Need to get 9,745 kB of archives.  
After this operation, 41.3 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Press **Enter**

```
Get:1 http://in.archive.ubuntu.com/ubuntu cosmic/universe  
5.8.0+ds-2 [1,185 kB]  
Fetched 9,745 kB in 18s (609 kB/s)  
Selecting previously unselected package npm.  
Unpacking npm (5.8.0+ds-2) ...  
Processing triggers for man-db (2.8.4-2) ...  
Setting up npm (5.8.0+ds-2) ...  
user@user:~]
```

- NOTE: The installation output on screen may differ on different systems based on the existing installation

# AWS Device Testing

## Node Red - Installation



```
user@user:~] sudo npm install -g --unsafe-perm node-red
```



```
/usr/local/bin/node-red-pi -> /usr/local/lib/node_modules/node-red/bin/node-red-pi
/usr/local/bin/node-red -> /usr/local/lib/node_modules/node-red/red.js
node-pre-gyp WARN Using request for node-pre-gyp https
[bcrypt] Success: "/usr/local/lib/node_modules/node-red/lib/binding/bcrypt_lib.node" is installed via remote
+ node-red@0.20.5
added 353 packages from 347 contributors in 22.089s
user@user:~]
```

- NOTE: The installation output on screen may differ on different systems based on the existing installation



# AWS Device Testing

## Node Red - Installation

```
user@user:~] node-red
```

```
17 May 18:22:48 - [info]
```

```
Welcome to Node-RED
```

```
=====
```

```
17 May 18:22:48 - [info] Node-RED version: v0.20.5
```

```
17 May 18:22:48 - [info] Node.js version: v8.11.4
```

```
17 May 18:22:48 - [info] Linux 4.18.0-20-generic x64 LE
```

```
17 May 18:22:50 - [info] Loading palette nodes
```

```
17 May 18:22:51 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
```

```
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.

Select and Press Ctrl Shift c

```
17 May 18:22:56 - [info] Starting flows
```

```
17 May 18:22:56 - [info] Started flows
```

```
17 May 18:22:56 - [info] Server now running at http://127.0.0.1:1880/
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/certificatehub

Search

Services Resource Groups

Syed Adil Mumbai Support

**AWS IoT**

Monitor

Onboard

Manage

**Secure**

- Certificates**
- Policies
- CAs
- Role Aliases
- Authorizers

Act

Test

Software

Settings

Learn

### Certificates

Search certificates

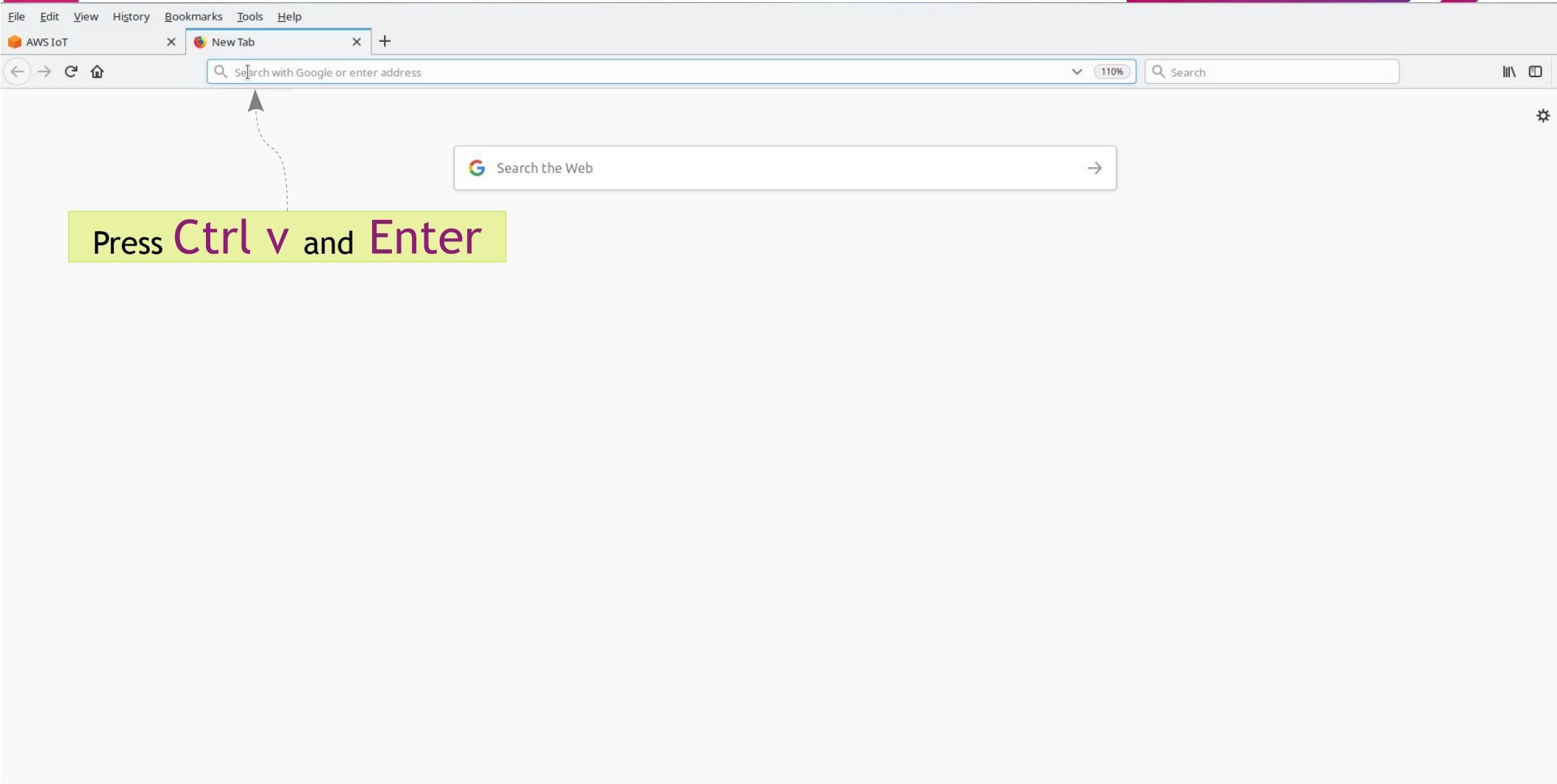
7c72dc344acf55c148...  
ACTIVE

Create

Card

# AWS Device Testing

## Node Red

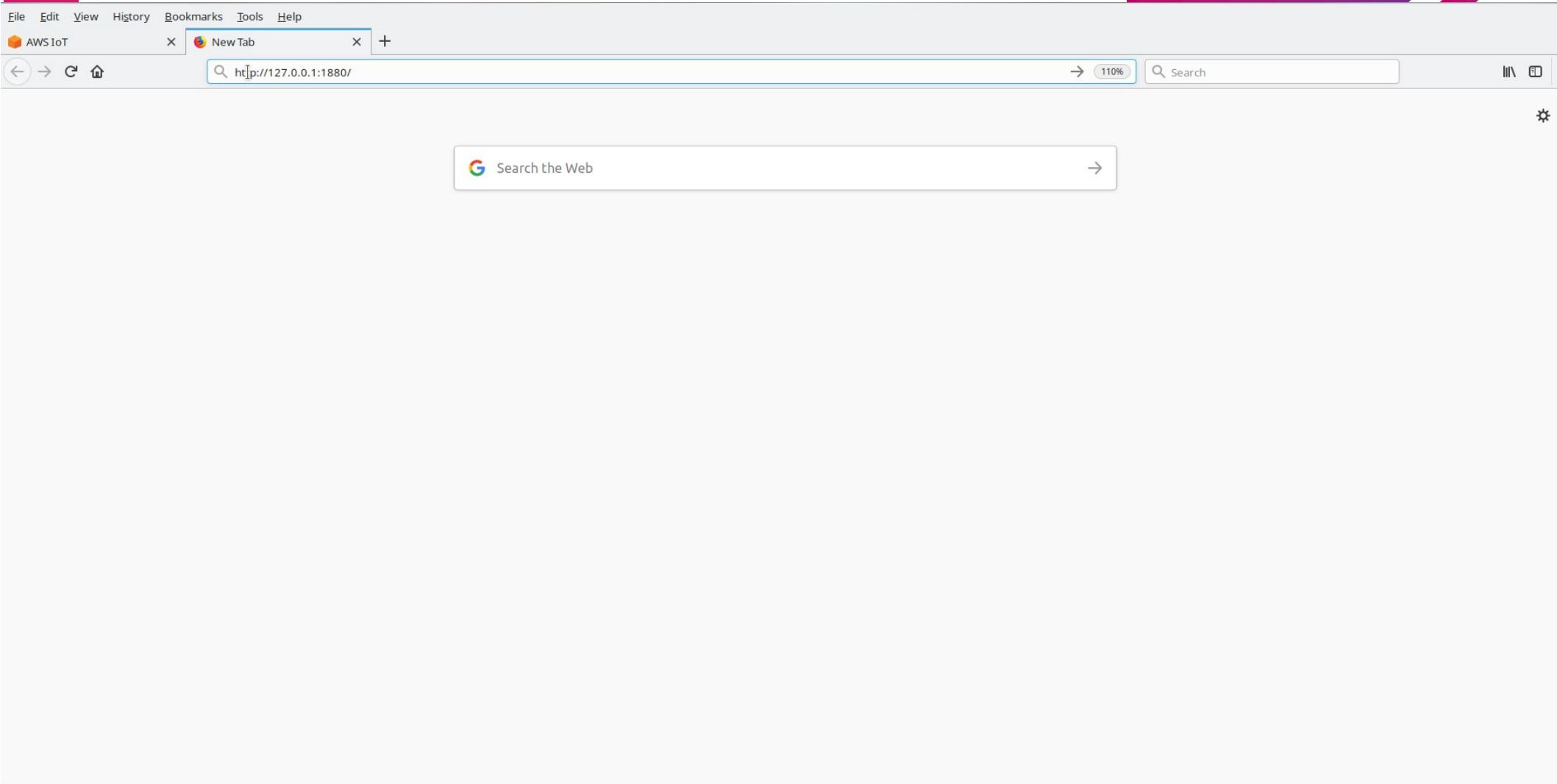


A Firefox Account is your key to new apps and continued awesomeness from Firefox. Make your online experience seamless.

[Sign up now](#)

# AWS Device Testing

## Node Red



A Firefox Account is your key to new apps and continued awesomeness from Firefox. Make your online experience seamless.

[Sign up now](#)

# AWS Device Testing

## Node Red

Node-RED

Flow 1

- 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
- Simple drag, drop and connect interface
- Built on Node.js
- Can run
  - Locally
  - On a Device
  - On Cloud

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Show the Info tab with `ctrl-g i`  
the Debug tab with `ctrl-g d`

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. The browser's address bar displays the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a top menu bar with 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. Below the menu is a tab bar with 'AWS IoT' and 'Node-RED'. The main workspace is titled 'Flow 1' and contains a grid for building flows. On the left, the 'input' node palette is visible, listing nodes: inject, catch, status, link, mqtt, http, websocket, tcp, and udp. A dashed circle highlights the 'inject' node, with a callout box containing the text 'Drag here and release'. The right sidebar shows the 'info' panel with 'Information' and 'Description' tabs. The 'Information' tab displays details for the current flow: Flow ID '43f6e27.1be399c', Name 'Flow 1', and Status 'Enabled'. At the bottom of the sidebar, there is a note: 'Enable or disable these tips from the option in the settings'.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Import a flow by dragging its JSON into the editor, or with `ctrl-i`

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Hold down **ctrl** when you click on a node to add or remove it from the current selection



# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. The browser's address bar displays the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface has a top bar with a "Deploy" button. On the left, a sidebar lists various nodes categorized into "input", "output", and "function". The "output" category is expanded, and the "debug" node is circled. A dashed line connects the "debug" node to a "timestamp" node already placed on the main workspace grid. A yellow callout box with the text "Drag here and release" points to the "timestamp" node. On the right, an "info" panel shows details for the selected flow, including its ID, name, and status.

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

Drag here and release

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Hold down **ctrl** when you click on a node to add or remove it from the current selection

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser window. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a top menu bar with options like File, Edit, View, History, Bookmarks, Tools, and Help. Below the menu is a toolbar with navigation icons and a search bar. The main workspace is a grid where nodes are placed and connected. In this flow, a 'timestamp' node is connected to a 'debug' node. The left sidebar is divided into three sections: 'input' (with nodes like inject, catch, status, link, mqtt, http, websocket, tcp, udp), 'output' (with nodes like debug, link, mqtt, http response, websocket, tcp, udp), and 'function'. The right sidebar shows the 'info' panel for the selected flow, displaying its ID, name, and status.

**Flow 1**

**input**

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

**output**

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

**function**

**info**

**Information**

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

**Description**

You can confirm your changes in the node edit tray with `ctrl-enter` or cancel them with `ctrl-escape`

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Move the selected nodes using the ↑ ↓ ← → keys. Hold ⇧ to nudge them further

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

ctrl-space will toggle the view of this sidebar

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Enable or disable these tips from the option in the settings

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

You can confirm your changes in the node edit tray with `ctrl-enter` or cancel them with `ctrl-escape`

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

click and drag on a node port to move all of the attached wires or just the selected one

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

Drag here and release

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with `ctrl-g`



# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Search for nodes using

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

ctrl click in the workspace to open the quick-add dialog

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Enable or disable these tips from the option in the settings

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Pressing `enter` will edit the first node in the current selection

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

click and drag on a node port to move all of the attached wires or just the selected one

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Hold down **ctrl** when you **click** on a node port to enable quick-wiring

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Double Click

ctrl click in the workspace to open the quick-add dialog

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser window. The address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The interface includes a top menu bar with File, Edit, View, History, Bookmarks, Tools, and Help. Below the menu is a toolbar with navigation icons and a search bar. The main workspace is titled "Flow 1" and contains a flow diagram with a "timestamp" node connected to "msg.payload" and an "mqtt" node. The "mqtt" node is selected, and its configuration panel is open on the right. The panel has tabs for "Properties" and "Info". The "Properties" tab is active, showing fields for "Server" (a dropdown menu with "Add new mqtt-broker..." selected), "Topic" (a text input field), "QoS" (a dropdown menu), "Retain" (a checkbox), and "Name" (a text input field). A tip box at the bottom of the properties panel states: "Tip: Leave topic, qos or retain blank if you want to set them via msg properties." The "Info" tab is also visible, showing information about the "mqtt out" node, including its ID "ed96660e.b7d83", type "mqtt out", and a description: "Connects to a MQTT broker and publishes messages." The right sidebar also contains an "Inputs" section with details about the "payload" property.

Node-RED interface showing a flow with a timestamp node connected to msg.payload and an mqtt node. The mqtt node configuration panel is open, showing fields for Server, Topic, QoS, Retain, and Name. The right sidebar displays the node's information, including its ID, type, and description.



# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface. On the left, the 'input' and 'output' node palettes are visible. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. A yellow box with the text 'Click Here' is positioned over the 'mqtt' node. On the right, the 'Edit mqtt out node' panel is open, showing the 'Properties' tab. The 'Server' dropdown is highlighted with a red box and a dashed arrow pointing to it from the 'Click Here' box. The 'Topic', 'QoS', and 'Retain' fields are also visible. A tip at the bottom of the panel reads: 'Tip: Leave topic, qos or retain blank if you want to set them via msg properties.'

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

function

timestamp

msg.payload

mqtt

Click Here

Edit mqtt out node

Delete Cancel Done

Properties

Server Add new mqtt-broker...

Topic Topic

QoS Retain

Name Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

info

Information

Node "ed96660e.b7d83"

Type mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload string | buffer  
most users prefer simple text payloads, but binary buffers can also be published.

topic string  
the MQTT topic to publish to.

qos number  
0, fire and forget - 1, at least once - 2, once and once only. Default 0.

retain boolean  
set to true to retain the message on the broker. Default false.

Details

msg.payload is used as the payload of the published message. If it contains an Object will be converted to a JSON string before

Hold down when you click on node to also select all of its connected nodes

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection Security Messages

Server e.g. localhost Port 1883

☐ Enable secure (SSL/TLS) connection

Client ID Leave blank for auto generated

☐ Keep alive time (s) 60 ☒ Use clean session

☒ Use legacy MQTT 3.1 support

info

Information

Node "69163c4f.344324"

Type mqtt-broker

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the

Pressing **enter** will edit the first node in the current selection

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The browser address bar shows '127.0.0.1:1880/#flow/43f6e27.1be399c'. The main workspace displays a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. A yellow callout box with the text 'Click Here' points to the 'mqtt' node in the left sidebar. The right sidebar shows the configuration for the selected 'mqtt' node, titled 'Edit mqtt out node > Add new mqtt-broker config node'. The configuration includes fields for 'Name', 'Server' (e.g. localhost), 'Port' (1883), 'Client ID' (Leave blank for auto generated), 'Keep alive time (s)' (60), and checkboxes for 'Enable secure (SSL/TLS) connection', 'Use clean session', and 'Use legacy MQTT 3.1 support'. The bottom right sidebar shows the 'info' tab with details about the node, including its ID '69163c4f.344324' and type 'mqtt-broker'.

Click Here

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/certificatehub

Search

aws Services Resource Groups

Syed Adil Mumbai Support

AWS IoT

Monitor

Onboard

Manage

Secure

- Certificates
- Policies
- CAs
- Role Aliases
- Authorizers

Act

Test

Software

Settings

Learn

### Certificates

Search certificates

7c72dc344acf55c148...  
ACTIVE

Create

Card

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/certificatehub

Search

Services Resource Groups

Syed Adil Mumbai Support

**AWS IoT**

Monitor

Onboard

Manage

**Secure**

- Certificates**
- Policies
- CAs
- Role Aliases
- Authorizers

Act

Test

**Certificates**

Create

Card

Search certificates

7c72dc344acf55c148...  
ACTIVE

Click Here

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Syed Adil Mumbai Support

Connected to Device Gateway on client ID 'iotconsole-1557657765090-0'.

### MQTT client

**Subscriptions**

[Subscribe to a topic](#)  
[Publish to a topic](#)

**Subscribe**  
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**  
Specify a topic to subscribe to, e.g. myTopic/1 [Subscribe to topic](#)

**Max message capture** [?](#)  
100

**Quality of Service** [?](#)  
☒ 0 - This client will not acknowledge to the Device Gateway that messages are received  
☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**  
☒ Auto-format JSON payloads (improves readability)  
☐ Display payloads as strings (more accurate)  
☐ Display raw payloads (in hexadecimal)

**Publish**  
Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1 [Publish to topic](#)

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as iotconsole-1557657765090-0

### MQTT client

Subscriptions

[Subscribe to a topic](#)

[Publish to a topic](#)

**Subscribe**

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**

**Subscribe to topic**

**Max message capture**

**Quality of Service**

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

**Publish**

Specify a topic and a message to publish with a QoS of 0.

**Publish to topic**

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as **iotconsole-1557657765090-0**

### MQTT client

**Subscriptions**

[Subscribe to a topic](#)

[Publish to a topic](#)

**Subscribe**

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**

Specify a topic to subscribe to, e.g. myTopic/1

**Subscribe to topic**

**Max message cap**

100

**Click Here**

**Quality of Service**

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

**Publish**

Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1

**Publish to topic**

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```



# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as **iotconsole-1557657765090-0**

Disconnect  
Clear all messages  
View endpoint

### MQTT client

**Subscriptions**

[Subscribe to a topic](#)  
[Publish to a topic](#)

**Subscribe**  
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**  
Specify a topic to subscribe to, e.g. myTopic/1 [Subscribe to topic](#)

**Max message capture** [?](#)  
100

**Quality of Service** [?](#)  
☒ 0 - This client will not acknowledge to the Device Gateway that messages are received  
☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**  
☒ Auto-format JSON payloads (improves readability)  
☐ Display payloads as strings (more accurate)  
☐ Display raw payloads (in hexadecimal)

**Publish**  
Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1 [Publish to topic](#)

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as **iotconsole-1557657765090-0**

Disconnect  
Clear all messages  
View endpoint

### MQTT client

**Subscriptions**

[Subscribe to a topic](#)  
[Publish to a topic](#)

**Subscribe**  
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**  
Specify a topic to subscribe to, e.g. myTopic/1 [Subscribe to topic](#)

**Max message capture** [?](#)  
100

**Quality of Service** [?](#)  
☒ 0 - This client will not acknowledge to the Device Gateway that messages are received  
☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**  
☒ Auto-format JSON payloads (improves readability)  
☐ Display payloads as strings (more accurate)  
☐ Display raw payloads (in hexadecimal)

**Publish**  
Specify a topic and a message to publish with a QoS of 0.  
Specify a topic to publish to, e.g. myTopic/1 [Publish to topic](#)

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

The screenshot displays the AWS IoT console's MQTT client interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'Syed Adil' from 'Mumbai'. The left sidebar lists navigation options: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main content area is titled 'MQTT client' and features a 'Subscriptions' tab. Under 'Subscriptions', there are links for 'Subscribe to a topic' and 'Publish to a topic'. The 'Subscribe' section includes a text input for 'Subscription topic' (placeholder: 'Specify a topic to subscribe to, e.g. myTopic/1'), a 'Max message capture' dropdown set to '100', and 'Quality of Service' radio buttons. The '0' option is selected, with a description: '0 - This client will not acknowledge to the Device Gateway that messages are received'. The '1' option is '1 - This client will acknowledge to the Device Gateway that messages are received'. Below this is the 'MQTT payload display' section with three radio buttons: 'Auto-format JSON payloads (improves readability)' (selected), 'Display payloads as strings (more accurate)', and 'Display raw payloads (in hexadecimal)'. The 'Publish' section has a text input for 'Specify a topic to publish to, e.g. myTopic/1'. At the bottom, a terminal window shows a JSON message: 

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

 A 'Help' sidebar on the right is open, showing the 'Custom endpoint' section, which is 'Enabled'. It explains that the custom endpoint allows connecting to AWS IoT via a REST API and is provisioned and ready to use. The endpoint value shown is 'a16lpc9okbq2a-ats.iot.ap-sou'.

# AWS Device Testing

## Node Red

The screenshot displays the AWS IoT console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar contains navigation links: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main content area is titled 'MQTT client' and features a 'Subscriptions' tab. Under 'Subscriptions', there are links for 'Subscribe to a topic' and 'Publish to a topic'. The 'Subscribe' section includes a 'Subscription topic' field, a 'Max message cap' field (highlighted with a yellow box and the text 'Double click Here'), and 'Quality of Service' options. The 'MQTT payload display' section offers three radio button options: 'Auto-format JSON payloads (improves readability)' (selected), 'Display payloads as strings (more accurate)', and 'Display raw payloads (in hexadecimal)'. The 'Publish' section at the bottom has a field to 'Specify a topic to publish to, e.g. myTopic/1' and a code editor showing a JSON message: 

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

. On the right, a 'Help' panel for 'Custom endpoint' is open, showing the 'Endpoint' field with the value 'a161pc9okbq2a-ats.iot.ap-sou'.

# AWS Device Testing

## Node Red

The screenshot shows the AWS IoT console interface for the MQTT client. The left sidebar contains navigation links: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main content area is titled "MQTT client" and includes a "Subscriptions" section with "Subscribe to a topic" and "Publish to a topic" links. The "Subscribe" section contains a "Subscription topic" input field, a "Max message capture" dropdown set to 100, and "Quality of Service" radio buttons. The "Publish" section contains a "Specify a topic to publish to" input field. A yellow callout box with the text "Press Ctrl c" points to the "Endpoint" field in the "Custom endpoint" help panel on the right. The "Custom endpoint" panel shows the endpoint "a161p19okbq2a-ats.iot.ap-sou" and a "Connect" button.

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Help

Custom endpoint

Enabled

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.

Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

Endpoint

a161p19okbq2a-ats.iot.ap-sou

Connect

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

Specify a topic to subscribe to, e.g. myTopic/1

Max message capture

100

Quality of Service

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

Publish

Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

# AWS Device Testing

## Node Red

The screenshot shows the AWS IoT console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'Syed Adil' from 'Mumbai'. The left sidebar lists various IoT services: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main content area is titled 'MQTT client' and features a 'Subscriptions' tab. Under this tab, there are links to 'Subscribe to a topic' and 'Publish to a topic'. The 'Subscribe' section includes a text input for 'Subscription topic' (with a placeholder 'Specify a topic to subscribe to, e.g. myTopic/1'), a 'Max message cap.' dropdown set to '100', and a 'Quality of Service' section with two radio buttons: '0 - This client will not acknowledge to the Device Gateway that messages are received' (selected) and '1 - This client will acknowledge to the Device Gateway that messages are received'. Below this is the 'MQTT payload display' section with three radio buttons: 'Auto-format JSON payloads (improves readability)' (selected), 'Display payloads as strings (more accurate)', and 'Display raw payloads (in hexadecimal)'. At the bottom, there is a 'Publish' section with a text input for 'Specify a topic to publish to, e.g. myTopic/1' and a code editor showing a JSON message: 

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

. A yellow box with the text 'Click Here' is positioned over the 'Subscribe' section. A dashed line connects the 'Help' button in the top right corner to the 'Help' sidebar on the right, which displays information about the 'Custom endpoint'.

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Help

MQTT client

Subscriptions

Subscribe to a topic

Publish to a topic

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

Specify a topic to subscribe to, e.g. myTopic/1

Max message cap.

100

Click Here

Quality of Service

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

Publish

Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

Custom endpoint

Enabled

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.

Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

Endpoint

a161pc9okbq2a-ats.iot.ap-sou

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The browser tabs show 'AWS IoT' and 'Node-RED'. The address bar indicates the URL '127.0.0.1:1880/#flow/43f6e27.1be399c'. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The left sidebar contains a 'filter nodes' search bar and categories for 'input', 'output', and 'function' nodes. The right sidebar is open to the 'Edit mqtt out node' configuration, showing the 'Add new mqtt-broker config node' dialog. The 'Properties' tab is active, displaying fields for 'Name', 'Connection' (Server: 'e.g. localhost', Port: '1883'), 'Security', and 'Messages'. The 'Server' field is highlighted. The 'Client ID' field is set to 'Leave blank for auto generated'. The 'Keep alive time (s)' is set to '60'. The 'Use clean session' checkbox is checked. The 'Use legacy MQTT 3.1 support' checkbox is also checked. The right sidebar also shows an 'info' panel with 'Node' information (ID: '69163c4f.344324', Type: 'mqtt-broker') and a 'Description' section explaining the configuration and its behavior.

Node-RED

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection

Server

Port

1883

Enable secure (SSL/TLS) connection

Client ID

Leave blank for auto generated

Keep alive time (s)

60

Use clean session

Use legacy MQTT 3.1 support

info

Information

Node

"69163c4f.344324"

Type

mqtt-broker

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the flow or by clicking the 'Close' button.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Click Here

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection

Server e.g. localhost Port 1883

Enable secure (SSL/TLS) connection

Client ID Leave blank for auto generated

Keep alive time (s) 60 Use clean session

Use legacy MQTT 3.1 support

info

Information

Node "69163c4f.344324"

Type mqtt-broker

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

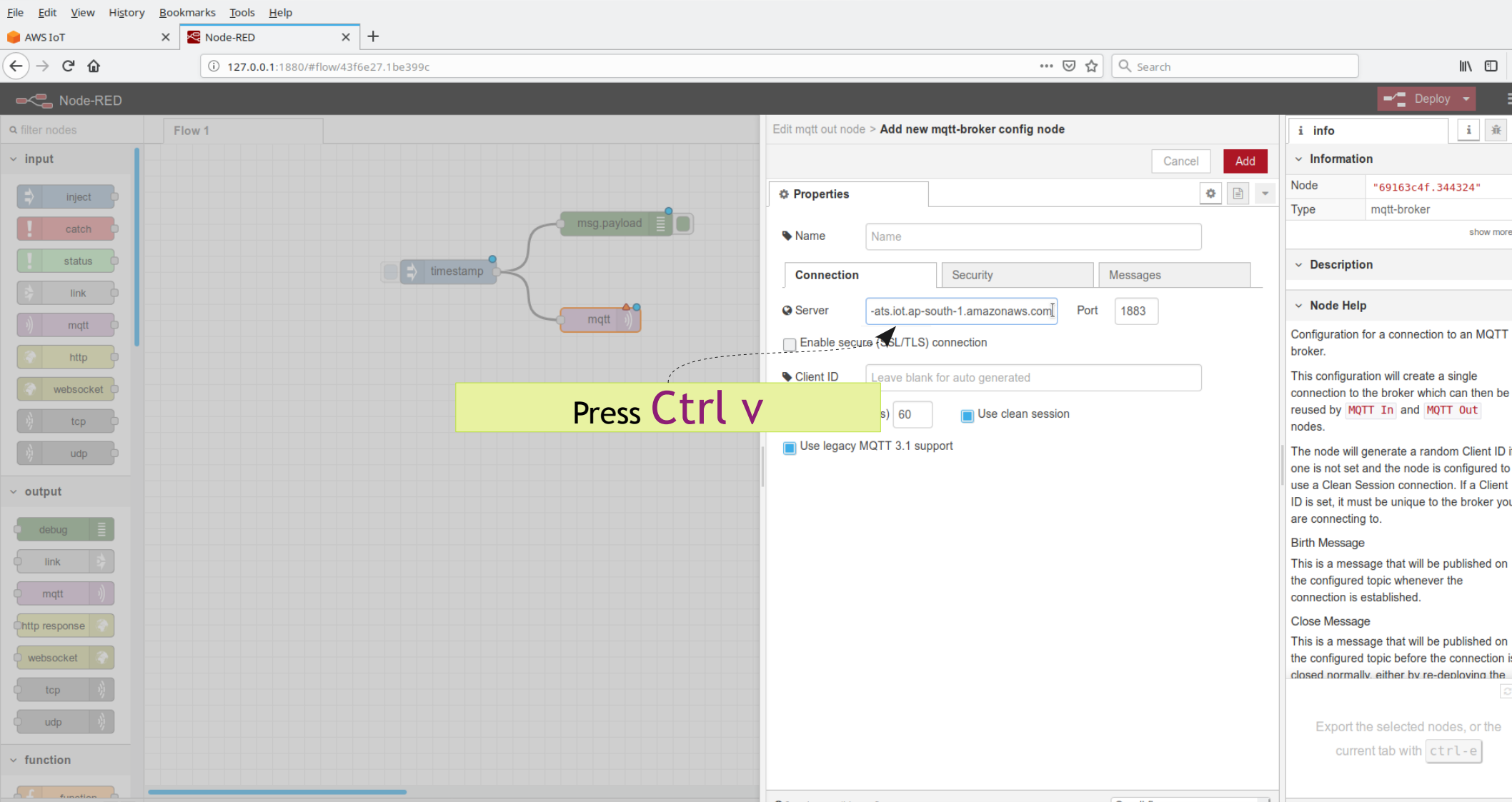
This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the flow or by manually closing the connection.

Import a flow by dragging its JSON into the editor, or with **ctrl-i**



# AWS Device Testing

## Node Red



The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with a 'timestamp' node connected to a 'msg.payload' node and an 'mqtt' node. A yellow callout box with the text 'Press Ctrl v' points to the 'mqtt' node. On the right, the 'Edit mqtt out node' configuration panel is open, showing the 'Add new mqtt-broker config node' form. The 'Server' field is populated with 'ats.iot.ap-south-1.amazonaws.com' and the 'Port' is '1883'. The 'Client ID' field is set to 'Leave blank for auto generated'. The 'Security' tab is selected, and the 'Use legacy MQTT 3.1 support' checkbox is checked. The right sidebar shows the 'info' panel with details about the selected node, including its ID '69163c4f.344324' and type 'mqtt-broker'.

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Deploy

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection Security Messages

Server ats.iot.ap-south-1.amazonaws.com Port 1883

Enable secure (SSL/TLS) connection

Client ID Leave blank for auto generated

60 Use clean session

Use legacy MQTT 3.1 support

info

Information

Node "69163c4f.344324"

Type mqtt-broker

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the

Export the selected nodes, or the current tab with **ctrl-e**

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Click Here

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection

Server

Port

Enable secure (SSL/TLS) connection

Client ID

alive time (s)

Use clean session

Use legacy MQTT 3.1 support

Messages

info

Information

Node

Type

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the node or by clicking on the node.

Hold down **Ctrl** when you **click** on a node to also select all of its connected nodes

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The browser address bar shows the URL '127.0.0.1:1880/#flow/43f6e27.1be399c'. The Node-RED interface shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The 'mqtt' node configuration panel is open, showing the following settings:

- Properties:** Name (empty)
- Connection:** Server: -ats.iot.ap-south-1.amazonaws.com, Port: 8883
- ☐ Enable secure (SSL/TLS) connection
- Client ID:** Leave blank for auto generated
- ☒ Keep alive time (s): 60
- ☒ Use clean session
- ☒ Use legacy MQTT 3.1 support

The right sidebar contains an 'info' panel with the following information:

- Information:** Node: "69163c4f.344324", Type: mqtt-broker
- Description:** Configuration for a connection to an MQTT broker. This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.
- Node Help:** The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.
- Birth Message:** This is a message that will be published on the configured topic whenever the connection is established.
- Close Message:** This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the flow or by manually closing the connection.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection Security Messages

Server -ats.iot.ap-south-1.amazonaws.com Port 8883

☐ Enable secure (SSL/TLS) connection

Client ID Leave blank for auto generated

☐ Keep alive time (s) 60 ☒ Use clean session

☒ Use legacy MQTT 3.1 support

Click Here

info

Information

Node "69163c4f.344324"

Type mqtt-broker

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the flow or by manually closing the connection.

Enable or disable these tips from the option in the settings

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The left sidebar contains a palette of input, output, and function nodes. The right sidebar is open to the configuration page for the 'mqtt' node, titled 'Edit mqtt out node > Add new mqtt-broker config node'. This page includes tabs for 'Properties', 'Connection', 'Security', and 'Messages'. The 'Connection' tab is active, showing fields for 'Server' (ats.iot.ap-south-1.amazonaws.com), 'Port' (8883), and options to 'Enable secure (SSL/TLS) connection' and 'Use legacy MQTT 3.1 support'. The 'info' sidebar on the far right provides details about the node, including its ID ('69163c4f.344324') and type ('mqtt-broker').

Node-RED interface showing a flow with a timestamp node connected to msg.payload and an mqtt node. The right sidebar displays the configuration for the mqtt node, including server, port, and security settings.

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. The top bar includes the AWS IoT logo and the Node-RED tab. The main workspace displays a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. A yellow callout box with the text 'Click Here' points to the 'Add new mqtt-broker config node' button in the configuration panel for the 'mqtt' node. The configuration panel shows the following settings:

- Properties:** Name (empty)
- Connection:** Server (-ats.iot.ap-south-1.amazonaws.com), Port (8883)
- Security:** ☒ Enable secure (SSL/TLS) connection, TLS Configuration (Add new tls-config...)
- Messages:** Client ID (Leave blank for auto generated), Keep alive time (s) (60), ☒ Use clean session, ☒ Use legacy MQTT 3.1 support

The right sidebar shows the 'info' tab for the selected node, displaying its ID ('69163c4f.344324') and type ('mqtt-broker'). The 'Description' section explains that this configuration creates a single connection to the MQTT broker, which can be reused by 'MQTT In' and 'MQTT Out' nodes. It also mentions that the node will generate a random Client ID if one is not set and configured to use a Clean Session connection.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node > Add new tls-config config node

Cancel Add

Properties

☐ Use key and certificates from local files

Certificate

Private Key

Passphrase

CA Certificate

☒ Verify server certificate

Server Name

Name

info

Information

Node "b285b6e.93a2748"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

You can confirm your changes in the node edit tray with **ctrl-enter** or cancel them with **ctrl-escape**

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Properties

Use key and certificates from local files

Certificate Upload

Private Key Upload

Passphrase private key passphrase (optional)

CA Certificate Upload

Verify server certificate

Server Name for use with SNI

Name

Click Here

info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

ctrl click in the workspace to open the quick-add dialog



# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. A 'File Upload' dialog is open, displaying a list of files in the 'Certificates' directory. The files are:

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

The '7c72dc344a-certificate.pem.crt' file is selected. The dialog has 'Cancel' and 'Open' buttons at the bottom right. The background shows the Node-RED interface with a sidebar on the left containing 'input' and 'output' nodes, and a right sidebar with 'info' and 'help' tabs. The 'info' tab is active, showing details for a 'tls-config' node.

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface running in a browser. A 'File Upload' dialog box is open, displaying a list of files in the 'Certificates' directory. The files are:

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

The '7c72dc344a-certificate.pem.crt' file is selected. The dialog has 'Cancel' and 'Open' buttons at the bottom right. The background shows the Node-RED interface with a sidebar on the left containing input, output, and function nodes, and a right sidebar with information and help for the selected node.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node > Add new tls-config config node

Cancel Add

Properties

☐ Use key and certificates from local files

Certificate  7c72dc344a-certificate.pem.crt

Private Key

Passphrase

CA Certificate

☒ Verify server certificate

Server Name

Name

info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function


timestamp



msg.payload

mqtt


Properties

Use key and certificates from local files

Certificate  7c72dc344a-certificate.pem.crt

Private Key  

Passphrase private key passphrase (optional)

CA Certificate 

Verify server certificate

Server Name for use with SNI

Name

Click Here

info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

Pressing **enter** will edit the first node in the current selection

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface running in a browser. A 'File Upload' dialog box is open, displaying a list of files in the 'Certificates' directory. The files are:

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

The dialog also shows a sidebar with file system locations like Desktop, Documents, Downloads, Music, Pictures, Videos, efi, Filesystem..., home, and WinD. At the bottom of the dialog are buttons for 'Cancel', 'Open', and 'All Files'.

The background interface shows the Node-RED workspace with a 'Flow 1' tab. The left sidebar contains input nodes (inject, catch, status, link, mqtt, http, websocket, tcp, udp) and output nodes (debug, link, mqtt, http response, websocket, tcp, udp). The right sidebar shows the 'info' tab for the selected node, displaying its ID ('e7d4be42.7f1dc') and type ('tls-config').

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. A 'File Upload' dialog is open, displaying a list of files in the 'Certificates' directory. The file '7c72dc344a-private.pem.key' is selected. The background interface shows the 'input' and 'output' node palettes on the left and a 'config node' panel on the right.

**File Upload Dialog:**

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

**Node-RED Interface:**

- input:** inject, catch, status, link, mqtt, http, websocket, tcp, udp.
- output:** debug, link, mqtt, http response, websocket, tcp, udp.
- function:** (empty)
- config node:** info, Add, Information (Node: "e7d4be42.7f1dc", Type: tls-config), Description, Node Help (Configuration options for TLS connections).

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. The browser's address bar displays the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with 'input' and 'output' nodes, a central workspace, and a right sidebar with 'info' and 'help' panels. A 'File Upload' dialog box is open in the center, showing a file list with columns for Name, Size, and Modified. The file `7c72dc344a-private.pem.key` is selected. The dialog also shows a sidebar with file system locations like Desktop, Documents, and Downloads.

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node > Add new tls-config config node

Cancel Add

Properties

☐ Use key and certificates from local files

Certificate  7c72dc344a-certificate.pem.crt

Private Key  7c72dc344a-private.pem.key

Passphrase

CA Certificate

☒ Verify server certificate

Server Name

Name

Info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

Show the Info tab with  i o  
the Debug tab with  d



# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Properties

Use key and certificates from local files

Certificate Upload 7c72dc344a-certificate.pem.crt

Private Key Upload 7c72dc344a-private.pem.key

Passphrase private key passphrase (optional)

CA Certificate Upload

Verify server certificate

Server Name for use with SNI

Name

Click Here

info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

Hold down ctrl when you click on a node to add or remove it from the current selection

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. A 'File Upload' dialog is open, displaying a list of files in the 'Certificates' directory. The files are:

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

The dialog also shows a sidebar with file system locations (Recent, Home, Desktop, Documents, Downloads, Music, Pictures, Videos, efi, Filesystem..., home, WinD, Other Locations) and buttons for 'Cancel', 'Open', and 'All Files'.

The background interface shows the Node-RED workspace with a 'Flow 1' tab. The left sidebar contains input nodes (inject, catch, status, link, mqtt, http, websocket, tcp, udp) and output nodes (debug, link, mqtt, http response, websocket, tcp, udp). The right sidebar shows the 'info' tab for the selected node, displaying its ID ('e7d4be42.7f1dc') and type ('tls-config').

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. A 'File Upload' dialog is open, displaying a list of files in the 'Certificates' directory. The files are:

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

The 'root\_ca.pem' file is selected. The dialog also shows a sidebar with file system locations like Desktop, Documents, Downloads, etc. The background shows the Node-RED interface with various input and output nodes available for configuration.

# AWS Device Testing

## Node Red

The screenshot shows the Node-RED web interface in a browser. A 'File Upload' dialog is open, displaying a list of files in the 'Certificates' directory. The files are:

Name	Size	Modified
7c72dc344a-certificate.pem.crt	1.2 kB	Yesterday
7c72dc344a-private.pem.key	1.7 kB	Yesterday
7c72dc344a-public.pem.key	451 bytes	Yesterday
root_ca.pem	1.2 kB	Yesterday

The 'root\_ca.pem' file is selected. The dialog has 'Cancel' and 'Open' buttons at the bottom right. The background shows the Node-RED interface with a sidebar on the left containing input and output nodes, and a right sidebar with information and help for the selected 'tls-config' node.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node > Add new tls-config config node

Cancel Add

Properties

☐ Use key and certificates from local files

Certificate  7c72dc344a-certificate.pem.crt x

Private Key  7c72dc344a-private.pem.key x

Passphrase

CA Certificate  root\_ca.pem x

☒ Verify server certificate

Server Name

Name

info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

Enable or disable these tips from the option in the settings

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node > Add new mqtt-broker config node > Add new tls-config config node

Cancel Add

Properties

☐ Use key and certificates from local files

Certificate  7c72dc344a-certificate.pem.crt

Private Key  7c72dc344a-private.pem.key

Passphrase

CA Certificate  root\_ca.pem

☒ Verify server certificate

Server Name

Name

Click Here

info

Information

Node "e7d4be42.7f1dc"

Type tls-config

Description

Node Help

Configuration options for TLS connections.

Hold down [Shift] when you click on a node to also select all of its connected nodes

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The browser address bar shows the URL '127.0.0.1:1880/#flow/43f6e27.1be399c'. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The left sidebar contains a 'filter nodes' search bar and categories for 'input', 'output', and 'function' nodes. The right sidebar is open to the 'Edit mqtt out node' configuration panel, which includes tabs for 'Properties', 'Connection', 'Security', and 'Messages'. The 'Connection' tab is active, showing settings for 'Server' (al6lpc9okbq2a-ats.iot.ap-south-1.an), 'Port' (8883), and options for 'Enable secure (SSL/TLS) connection', 'Client ID', 'Keep alive time (s)', 'Use clean session', and 'Use legacy MQTT 3.1 support'. The 'Add' button is highlighted. The bottom right sidebar shows the 'info' panel with 'Node' information and a 'Description' section.

Node-RED

Flow 1

input

output

function

msg.payload

timestamp

mqtt

Edit mqtt out node > Add new mqtt-broker config node

Cancel Add

Properties

Name

Connection Security Messages

Server al6lpc9okbq2a-ats.iot.ap-south-1.an Port 8883

☒ Enable secure (SSL/TLS) connection

TLS Configuration TLS configuration

Client ID Leave blank for auto generated

☐ Keep alive time (s) 60 ☒ Use clean session

☒ Use legacy MQTT 3.1 support

info

Information

Node "69163c4f.344324"

Type mqtt-broker

Description

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to use a Clean Session connection. If a Client ID is set, it must be unique to the broker you are connecting to.

Birth Message

This is a message that will be published on the configured topic whenever the connection is established.

Close Message

This is a message that will be published on the configured topic before the connection is closed normally, either by re-deploying the flow or by manually closing the connection.

ctrl-space will toggle the view of this sidebar

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface. On the left, the 'input' and 'output' node palettes are visible. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The 'mqtt' node is selected, and the 'Edit mqtt out node' panel is open on the right. This panel contains fields for 'Server', 'Topic', 'QoS', 'Retain', and 'Name'. A yellow callout box with the text 'Click Here' points to the 'Done' button in the top right corner of the 'Edit mqtt out node' panel. The right sidebar shows the 'info' tab for the selected node, displaying its ID, type, and description.

Node-RED

Flow 1

input

output

function

timestamp

msg.payload

mqtt

Edit mqtt out node

Done

Click Here

info

Information

Node: "ed96660e.b7d83"

Type: mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload

topic

retain

Details

msg.payload is used as the payload of the published message. If it contains an Object it will be converted to a JSON string before



# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node

Delete Cancel Done

Properties

Server altpc9okbq2a-ats.iot.ap-south-1.amazo

Topic Topic

QoS QoS Retain Retain

Name Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Click Here

info

Information

Node "ed96660e.b7d83"

Type mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload string | buffer  
most users prefer simple text payloads, but binary buffers can also be published.

topic string  
the MQTT topic to publish to.

qos number  
0, fire and forget - 1, at least once - 2, once and once only. Default 0.

retain boolean  
set to true to retain the message on the broker. Default false.

Details

msg.payload is used as the payload of the published message. If it contains an Object will be converted to a JSON string before

Dragging a node onto a wire will splice it into the link

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The address bar shows the URL '127.0.0.1:1880/#flow/43f6e27.1be399c'. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The left sidebar contains a 'filter nodes' search bar and categories for 'input', 'output', and 'function' nodes. The right sidebar is divided into two panels: 'Edit mqtt out node' and 'info'. The 'Edit mqtt out node' panel shows the following properties:

- Delete**: Button
- Cancel**: Button
- Done**: Button
- Properties**: Section header
- Server**: Dropdown menu with value 'al6lpc9okbq2a-ats.iot.ap-south-1.amazo'
- Topic**: Text input field with value 'MyFirstDevice'
- QoS**: Dropdown menu
- Retain**: Dropdown menu
- Name**: Text input field with value 'Name'
- Tip**: Leave topic, qos or retain blank if you want to set them via msg properties.

The 'info' panel on the right shows the following information:

- info**: Section header
- Information**: Section header
- Node**: 'ed96660e.b7d83'
- Type**: 'mqtt out'
- Description**: Section header
- Node Help**: Section header
- Connects to a MQTT broker and publishes messages.**
- Inputs**: Section header
- payload**: 'string | buffer'
- topic**: 'string'
- qos**: 'number'
- retain**: 'boolean'
- Details**: Section header
- msg.payload**: 'is used as the payload of the published message. If it contains an Object will be converted to a JSON string before'

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node

Delete Cancel Done

Properties

Server al6lpc9okbq2a-ats.iot.ap-south-1.amazo

Topic MyFirstDevice

QoS Retain

Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Click Here

info

Information

Node "ed96660e.b7d83"

Type mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload string | buffer  
most users prefer simple text payloads, but binary buffers can also be published.

topic string  
the MQTT topic to publish to.

qos number  
0, fire and forget - 1, at least once - 2, once and once only. Default 0.

retain boolean  
set to true to retain the message on the broker. Default false.

Details

msg.payload is used as the payload of the published message. If it contains an Object will be converted to a JSON string before

Hold down ctrl when you click on a node to add or remove it from the current selection

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node

Delete Cancel Done

Properties

Server al6lpc9okbq2a-ats.iot.ap-south-1.amazo

Topic MyFirstDevice

QoS 0

Retain

Name

Tip: Leave topic properties.

info

Information

Node "ed96660e.b7d83"

Type mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload

topic

qos

retain

Details

msg.payload is used as the payload of the published message. If it contains an Object it will be converted to a JSON string before

Hold down when you click on node to also select all of its connected nodes

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

Flow 1

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

mqtt

Edit mqtt out node

Delete Cancel Done

Properties

Server al6lpc9okbq2a-ats.iot.ap-south-1.amazo

Topic MyFirstDevice

QoS 0 Retain

Name 0

Tip: Leave topic properties.

1

2

Info

Information

Node "ed96660e.b7d83"

Type mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload string | buffer

topic string

qos number

retain boolean

Details

msg.payload is used as the payload of the published message. If it contains an Object it will be converted to a JSON string before

You can manage your palette of nodes with ctrl+sp

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface. On the left, the 'input' and 'output' node palettes are visible. The main workspace shows a flow named 'Flow 1' with a 'timestamp' node connected to 'msg.payload' and an 'mqtt' node. The 'mqtt' node is selected, and its configuration panel is open on the right. The configuration panel includes fields for 'Server', 'Topic', 'QoS', and 'Name'. A yellow tip box states: 'Tip: Leave topic, qos or retain blank if you want to set them via msg properties.' A yellow box with the text 'Click Here' points to the 'Done' button in the configuration panel. The right sidebar shows the 'info' tab with details about the selected node, including its ID, type, and description.

Node-RED

Flow 1

input

output

function

mqtt out node

Properties

Server: al6lpc9okbq2a-ats.iot.ap-south-1.amazo

Topic: MyFirstDevice

QoS: 0

Name: Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Click Here

info

Information

Node: "ed96660e.b7d83"

Type: mqtt out

Description

Node Help

Connects to a MQTT broker and publishes messages.

Inputs

payload: string | buffer

topic: string

qos: number

retain: boolean

Details

msg.payload is used as the payload of the published message. If it contains an Object will be converted to a JSON string before

Pressing enter will edit the first node in the current selection

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. Below this, the browser address bar shows the URL '127.0.0.1:1880/#flow/43f6e27.1be399c'. The Node-RED interface features a left sidebar with a 'filter nodes' search bar and two main categories: 'input' and 'output'. The 'input' category includes nodes like inject, catch, status, link, mqtt, http, websocket, tcp, and udp. The 'output' category includes debug, link, mqtt, http response, websocket, tcp, and udp. The 'function' category is also visible at the bottom. The main workspace, titled 'Flow 1', contains a flow diagram. It starts with a 'timestamp' node (blue) which connects to two output nodes: 'msg.payload' (green) and 'MyFirstDevice' (orange). The 'MyFirstDevice' node is an MQTT output node. On the right side of the interface, there is an 'info' sidebar. It contains sections for 'Information', 'Description', 'Node Help', and 'Details'. The 'Information' section shows the node ID as 'ed96660e.b7d83' and the type as 'mqtt out'. The 'Description' section explains that the node connects to a MQTT broker and publishes messages. The 'Node Help' section lists inputs: 'payload' (string | buffer), 'topic' (string), 'qos' (number), and 'retain' (boolean). The 'Details' section notes that 'msg.payload' is used as the payload of the published message and will be converted to a JSON string if it is an object. A 'Deploy' button is located in the top right corner of the Node-RED interface.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

MyFirstDevice

Deploy

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

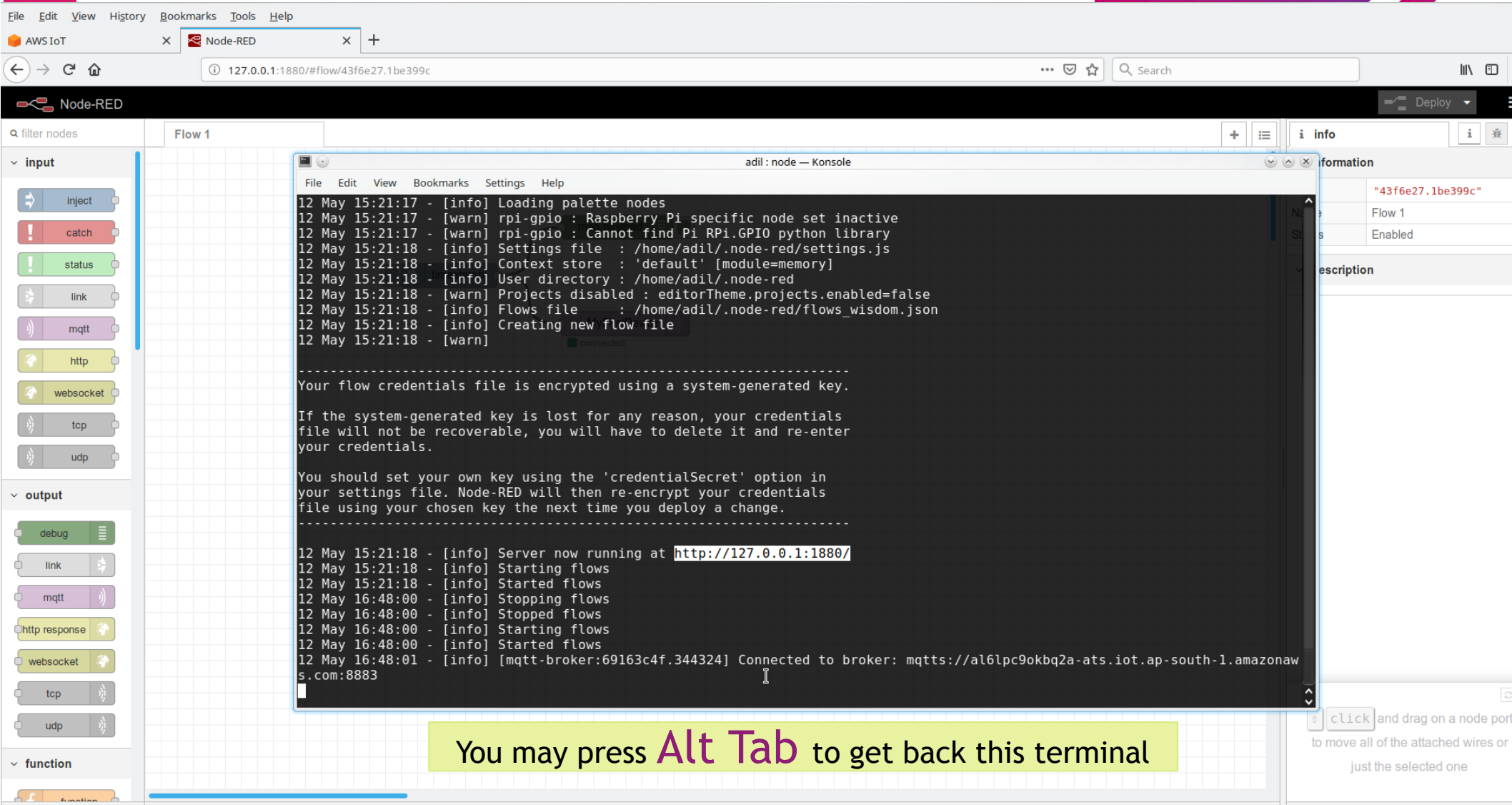
Click Here

click and drag on a node port to move all of the attached wires or just the selected one



# AWS Device Testing

## Node Red



The screenshot shows the Node-RED web interface in a browser. The address bar displays `127.0.0.1:1880/#flow/43f6e27.1be399c`. The interface includes a left sidebar with node palettes for input, output, and function. The main workspace shows a single flow named "Flow 1". A terminal window titled "adil: node — Konsole" is open, displaying the following logs:

```
12 May 15:21:17 - [info] Loading palette nodes
12 May 15:21:17 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
12 May 15:21:17 - [warn] rpi-gpio : Cannot find Pi RPi.GPIO python library
12 May 15:21:18 - [info] Settings file : /home/adil/.node-red/settings.js
12 May 15:21:18 - [info] Context store : 'default' [module=memory]
12 May 15:21:18 - [info] User directory : /home/adil/.node-red
12 May 15:21:18 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 May 15:21:18 - [info] Flows file : /home/adil/.node-red/flows_wisdom.json
12 May 15:21:18 - [info] Creating new flow file
12 May 15:21:18 - [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.
-----

12 May 15:21:18 - [info] Server now running at http://127.0.0.1:1880/
12 May 15:21:18 - [info] Starting flows
12 May 15:21:18 - [info] Started flows
12 May 16:48:00 - [info] Stopping flows
12 May 16:48:00 - [info] Stopped flows
12 May 16:48:00 - [info] Starting flows
12 May 16:48:00 - [info] Started flows
12 May 16:48:01 - [info] [mqtt-broker:69163c4f.344324] Connected to broker: mqtt://al6lpc9okbq2a-ats.iot.ap-south-1.amazonaws.com:8883
```

A yellow callout box at the bottom of the terminal window states: "You may press **Alt Tab** to get back this terminal".

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The browser address bar shows the URL '127.0.0.1:1880/#flow/43f6e27.1be399c'. The interface is divided into several sections:

- Left Panel:** Contains 'filter nodes' and two palettes: 'input' (with nodes like inject, catch, status, link, mqtt, http, websocket, tcp, udp) and 'output' (with nodes like debug, link, mqtt, http response, websocket, tcp, udp). A 'function' palette is partially visible at the bottom.
- Center Canvas:** Labeled 'Flow 1', it shows a grid for placing and connecting nodes.
- Right Panel:** Includes an 'info' tab showing details for 'Flow 1' (ID: '43f6e27.1be399c', Status: 'Enabled') and a 'description' tab.
- Console Window:** A terminal window titled 'adil: node — Konsole' is open, displaying the following logs:

```
12 May 15:21:17 - [info] Loading palette nodes
12 May 15:21:17 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
12 May 15:21:17 - [warn] rpi-gpio : Cannot find Pi RPi.GPIO python library
12 May 15:21:18 - [info] Settings file : /home/adil/.node-red/settings.js
12 May 15:21:18 - [info] Context store : 'default' [module=memory]
12 May 15:21:18 - [info] User directory : /home/adil/.node-red
12 May 15:21:18 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 May 15:21:18 - [info] Flows file : /home/adil/.node-red/flows_wisdom.json
12 May 15:21:18 - [info] Creating new flow file
12 May 15:21:18 - [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.
-----

12 May 15:21:18 - [info] Server now running at http://127.0.0.1:1880/
12 May 15:21:18 - [info] Starting flows
12 May 15:21:18 - [info] Started flows
12 May 16:48:00 - [info] Stopping flows
12 May 16:48:00 - [info] Stopped flows
12 May 16:48:00 - [info] Starting flows
12 May 16:48:00 - [info] Started flows
12 May 16:48:01 - [info] [mqtt-broker:69163c4f.344324] Connected to broker:[mqtt://al6lpc9okbq2a-ats.iot.ap-south-1.amazonaws.com:8883]
```

At the bottom right, a hint reads: 'Move the selected nodes using the [up/down] and [left/right] keys. Hold [shift] to nudge them further'.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

MyFirstDevice

connected

info

Information

Flow	"43f6e27.1be399c"
Name	Flow 1
Status	Enabled

Description

Click Here

Move the selected nodes using the ↑ ↓ ← → keys. Hold ⇧ to nudge them further

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with a 'filter nodes' search bar and three main categories: 'input', 'output', and 'function'. The 'input' category is expanded, showing nodes like 'inject', 'catch', 'status', 'link', 'mqtt', 'http', 'websocket', 'tcp', and 'udp'. The 'output' category is also expanded, showing nodes like 'debug', 'link', 'mqtt', 'http response', 'websocket', 'tcp', and 'udp'. The main workspace, titled 'Flow 1', contains a flow with three nodes: a 'timestamp' node (blue), a 'msg.payload' node (green), and a 'MyFirstDevice' node (purple). The 'timestamp' node is connected to both the 'msg.payload' and 'MyFirstDevice' nodes. The 'MyFirstDevice' node has a 'connected' status indicator. On the right side of the interface, there is a 'debug' tab and a 'Deploy' button.

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with a node palette categorized into 'input', 'output', and 'function'. The 'input' category is expanded, showing nodes like inject, catch, status, link, mqtt, http, websocket, tcp, and udp. The 'output' category shows debug, link, mqtt, http response, websocket, tcp, and udp. The 'function' category is partially visible. The main workspace, titled 'Flow 1', contains a flow diagram. A 'timestamp' node is circled with a dashed line. It is connected to two output nodes: 'msg.payload' and 'MyFirstDevice'. The 'MyFirstDevice' node has a green 'connected' indicator. A yellow callout box with the text 'Click Here' and an arrow points to the 'timestamp' node. On the right side of the interface, there is a 'debug' tab and a 'Deploy' button.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

MyFirstDevice

connected

debug

5/12/2019, 4:51:14 PM node: 5cb6fb32.695ae4

msg.payload : number

1557660074676

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with a 'filter nodes' search bar and three categories of nodes: 'input', 'output', and 'function'. The 'input' category is expanded, showing nodes like 'inject', 'catch', 'status', 'link', 'mqtt', 'http', 'websocket', 'tcp', and 'udp'. The 'output' category is also expanded, showing nodes like 'debug', 'link', 'mqtt', 'http response', 'websocket', 'tcp', and 'udp'. The 'function' category is partially visible at the bottom. The main workspace, titled 'Flow 1', contains a flow diagram. It starts with a 'timestamp' node (blue) which branches into two paths. The top path leads to a 'msg.payload' node (green), and the bottom path leads to a 'MyFirstDevice' node (purple). The 'MyFirstDevice' node has a 'connected' status indicator. A yellow callout box with the text 'Click Here' is positioned below the 'timestamp' node, with a dashed line pointing to the 'MyFirstDevice' node. On the right side of the interface, there is a 'debug' console showing a log entry: '5/12/2019, 4:51:14 PM node: 5cb6fb32.695ae4 msg.payload : number 1557660074676'. The browser's top bar shows tabs for 'AWS IoT' and 'Node-RED', and a search bar.

# AWS Device Testing

## Node Red

The screenshot displays the AWS IoT console's MQTT client interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar lists navigation options: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main content area is titled 'MQTT client' and features a 'Subscriptions' tab. Under 'Subscriptions', there are links for 'Subscribe to a topic' and 'Publish to a topic'. The 'Subscribe' section includes a description, a 'Subscription topic' input field, a 'Max message capture' dropdown set to 100, and 'Quality of Service' radio buttons (0 is selected). The 'MQTT payload display' section has three radio buttons: 'Auto-format JSON payloads (improves readability)' (selected), 'Display payloads as strings (more accurate)', and 'Display raw payloads (in hexadecimal)'. The 'Publish' section includes a description and a topic input field. A code editor at the bottom shows a JSON payload: 

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

. A 'Help' sidebar on the right is open, showing the 'Custom endpoint' section, which is 'Enabled' and provides instructions on how to use the custom endpoint for connecting to AWS IoT.



# AWS Device Testing

## Node Red

The screenshot shows the AWS IoT console interface for the MQTT client. The left sidebar contains navigation links: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main content area is titled 'MQTT client' and includes a 'Subscriptions' tab. Under 'Subscriptions', there are links to 'Subscribe to a topic' and 'Publish to a topic'. The 'Subscribe' section includes a 'Subscription topic' input field, a 'Max message capture' dropdown set to '100', and a 'Quality of Service' section with two radio button options. A yellow box labeled 'Click Here' points to the first option: '0 - This client will not acknowledge to the Device Gateway that messages are received'. A dashed line connects this option to the 'Custom endpoint' section on the right. The 'Custom endpoint' section is titled 'Enabled' and contains a text area with the endpoint address 'a161pc9okbq2a-ats.iot.ap-sou...'. The 'Publish' section at the bottom includes a 'Specify a topic and a message to publish with a QoS of 0.' input field and a code editor showing a JSON message: 

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as iotconsole-1557659605254-2

### MQTT client

Subscriptions

[Subscribe to a topic](#)

[Publish to a topic](#)

**Subscribe**

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**

**Subscribe to topic**

**Max message capture**

**Quality of Service**

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

**Publish**

Specify a topic and a message to publish with a QoS of 0.

**Publish to topic**

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as iotconsole-1557659605254-2

### MQTT client

**Subscriptions**

[Subscribe to a topic](#)  
[Publish to a topic](#)

**Subscribe**  
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

**Subscription topic**  
Specify a topic to subscribe to, e.g. myTopic/1 [Subscribe to topic](#)

**Max message capture** [?](#)  
100

**Quality of Service** [?](#)  
☒ 0 - This client will not acknowledge to the Device Gateway that messages are received  
☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**  
☒ Auto-format JSON payloads (improves readability)  
☐ Display payloads as strings (more accurate)  
☐ Display raw payloads (in hexadecimal)

**Publish**  
Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1 [Publish to topic](#)

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as iotconsole-1557659605254-2

### MQTT client

Subscriptions

[Subscribe to a topic](#)

[Publish to a topic](#)

**Subscribe**

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

[Subscribe to topic](#)

Max message capture [?](#)

Quality of Service [?](#)

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

**Publish**

Specify a topic and a message to publish with a QoS of 0.

[Publish to topic](#)

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Services Resource Groups

Connected as **iotconsole-1557659605254-2**

### MQTT client

**Subscriptions**

[Subscribe to a topic](#)

[Publish to a topic](#)

**Subscribe**

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

MyFirstDevice

Max message capture [?](#)

100

Quality of Service [?](#)

☒ 0 - This client will not acknowledge to the Device Gateway that messages are received

☐ 1 - This client will acknowledge to the Device Gateway that messages are received

**MQTT payload display**

☒ Auto-format JSON payloads (improves readability)

☐ Display payloads as strings (more accurate)

☐ Display raw payloads (in hexadecimal)

**Publish**

Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1

[Publish to topic](#)

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

aws Services Resource Groups

Connected as iotconsole-1557659605254-2

### MQTT client

Subscriptions

Subscribe to a topic

Publish to a topic

MyFirstDevice

MyFirstDevice

Export Clear Pause

Publish

Specify a topic and a message to publish with a QoS of 0.

MyFirstDevice

Publish to topic

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

# AWS Device Testing

## Node Red

The screenshot shows the AWS IoT console interface. On the left, there is a navigation menu with options: Monitor, Onboard, Manage, Secure, Act, Test, Software, Settings, and Learn. The main area is titled 'MQTT client' and shows a 'Subscriptions' list on the left with 'MyFirstDevice' selected. The right pane is titled 'MyFirstDevice' and contains a 'Publish' section with the instruction 'Specify a topic and a message to publish with a QoS of 0.' Below this is a text input field containing 'MyFirstDevice' and a 'Publish to topic' button. A code editor shows a JSON message: 

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

. A yellow box with the text 'Click Here' is positioned below the code editor, with a dashed line pointing from the 'Tools' menu in the Node-RED browser tab to the 'Publish to topic' button.

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with node categories: **input** (inject, catch, status, link, mqtt, http, websocket, tcp, udp) and **output** (debug, link, mqtt, http response, websocket, tcp, udp). The main workspace shows a flow named "Flow 1" with a "timestamp" node connected to "msg.payload" and "MyFirstDevice". The right sidebar shows a "debug" console with a log entry: "5/12/2019, 4:51:14 PM node: 5cb6fb32.695ae4 msg.payload : number 1557660074676".



# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with a 'filter nodes' search bar and three categories of nodes: 'input', 'output', and 'function'. The 'input' category is expanded, showing nodes like 'inject', 'catch', 'status', 'link', 'mqtt', 'http', 'websocket', 'tcp', and 'udp'. The 'output' category is also expanded, showing 'debug', 'link', 'mqtt', 'http response', 'websocket', 'tcp', and 'udp'. The 'function' category is partially visible at the bottom.

The main workspace, titled 'Flow 1', contains a flow diagram. It starts with a 'timestamp' node (blue) which is circled with a dashed line. A dashed arrow points from a yellow box labeled 'Click Here' to this 'timestamp' node. The 'timestamp' node is connected to two output nodes: 'msg.payload' (green) and 'MyFirstDevice' (purple). The 'MyFirstDevice' node has a green 'connected' indicator.

On the right side of the interface, there is a 'debug' console. It shows a log entry for the time '5/12/2019, 4:51:14 PM' with node ID '5cb6fb32.695ae4'. The log entry shows 'msg.payload : number' and the value '1557660074676'.

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT x Node-RED x +

127.0.0.1:1880/#flow/43f6e27.1be399c

Node-RED

filter nodes

Flow 1

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp

function

timestamp

msg.payload

MyFirstDevice

connected

debug

all nodes

5/12/2019, 4:51:14 PM node: 5cb6fb32.695ae4  
msg.payload : number  
1557660074676

5/12/2019, 5:00:20 PM node: 5cb6fb32.695ae4  
msg.payload : number  
1557660620384

# AWS Device Testing

## Node Red

The screenshot displays the Node-RED web interface in a browser. The browser's address bar shows the URL `127.0.0.1:1880/#flow/43f6e27.1be399c`. The Node-RED interface includes a left sidebar with node categories: **input** (inject, catch, status, link, mqtt, http, websocket, tcp, udp), **output** (debug, link, mqtt, http response, websocket, tcp, udp), and **function**. The main workspace, titled "Flow 1", contains a flow with the following nodes: a **timestamp** node, a **msg.payload** node, and a **MyFirstDevice** node (an MQTT client). The **MyFirstDevice** node is marked as "connected". A dashed line connects the **timestamp** node to the **MyFirstDevice** node. A yellow callout box with the text "Click Here" points to the **MyFirstDevice** node. On the right, the **debug** console shows two log entries:

```
5/12/2019, 4:51:14 PM node: 5cb6fb32.695ae4  
msg.payload : number  
1557660074676  
5/12/2019, 5:00:20 PM node: 5cb6fb32.695ae4  
msg.payload : number  
1557660620384
```

# AWS Device Testing

## Node Red

FileEditViewHistoryBookmarksToolsHelp

AWS IoTNode-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

Search

aws

ServicesResource Groups

Syed AdilMumbaiSupport

AWS IoT

MonitorOnboardManageSecureActTest

MQTT client

Connected as iotconsole-1557659605254-2

Subscriptions

Subscribe to a topic

Publish to a topic

MyFirstDevice

MyFirstDevice

ExportClearPause

Publish

Specify a topic and a message to publish with a QoS of 0.

MyFirstDevice

Publish to topic

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

MyFirstDevice

May 12, 2019 5:00:20 PM +0530

ExportHide

1557660620384

# AWS Device Testing

## Node Red

File Edit View History Bookmarks Tools Help

AWS IoT Node-RED

https://ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test

aws Services Resource Groups

Connected as iotconsole-1557659605254-2

### MQTT client

Subscriptions

Subscribe to a topic

Publish to a topic

MyFirstDevice

MyFirstDevice

Export Clear Pause

Publish

Specify a topic and a message to publish with a QoS of 0.

MyFirstDevice

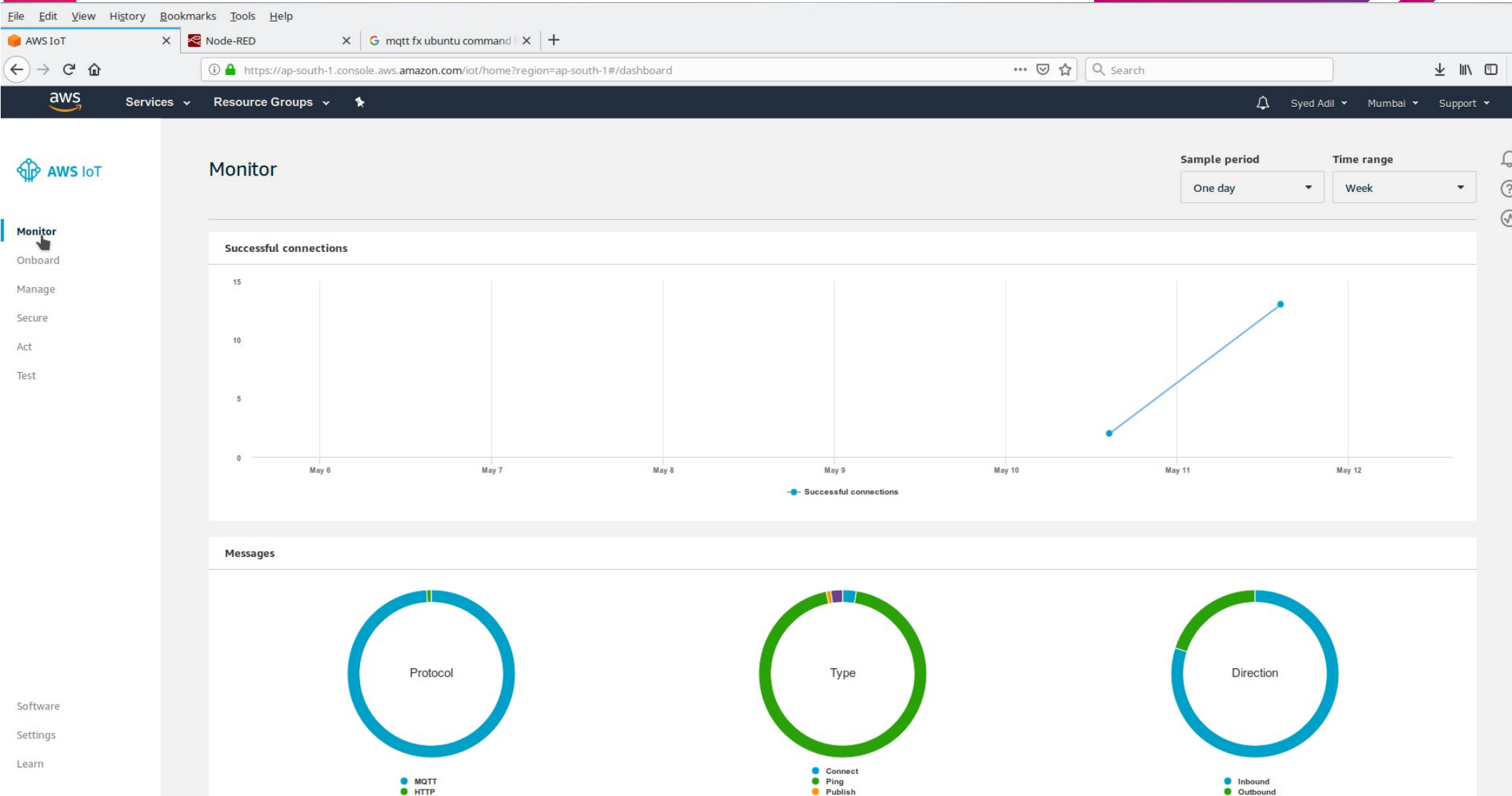
1 {  
2 "message": "Hello from AWS IoT console"  
3 }

MyFirstDevice May 12, 2019 5:00:20 PM +0530 Export Hide

1557660620384

# AWS Device Testing

## Node Red



Thank You