

24CYS333 - IoT - MQTT Implementation

Praneesh R V
CB.SC.U4CYS23036

Go to the AWS IoT Section and And go to MQTT Test client

The screenshot shows the AWS IoT console with the 'MQTT test client' page selected. On the left, there's a sidebar with various navigation options like 'Monitor', 'Connect', 'Test', 'Manage', and 'Logs'. The main area is titled 'MQTT test client' and contains instructions for establishing a connection. It has fields for 'Enter or generate a client ID' (containing 'iotconsole-9c5e3268-106d-4566-9cd5-925fd3f8339d') and a 'Generate new client ID' button. Below that, it shows the 'Endpoint' as 'a9631eakghbz1-ats.iot.ap-northeast-1.amazonaws.com' and the 'MQTT version' set to 'MQTT 5'. There are also checkboxes for 'Publish last will message' and 'Advanced configurations'. At the bottom right is a large orange 'Connect' button.

Then connect to the MQTT Test Client

MQTT test client Info

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

▼ Connection details Connected
[Disconnect](#)

To disconnect from the MQTT test client, choose Disconnect. To re-establish a connection, you can update the details and choose Connect.

| | | | |
|--|-----------------------------------|---|-----------------------------------|
| Client ID iotconsole-9c5e3268-106d-4566-9cd5-925fd 3f8339d | MQTT version 5 | Clean start Continue an existing session | Keep alive duration 30 seconds |
| Endpoint a9631eakgh6z1-ats.iot.ap-northeast-1.amazonaws.com | Will retain - | Will topic - | Will QoS - |
| Will payload - | Session expiry - | Receive maximum - | Maximum packet size 149504 |
| Topic alias maximum - | Request response information - | Request problem information - | |

User properties (0)

| Key | Value |
|--------------------------------------|-------|
| This message has no user properties. | |

AWS IoT > **MQTT test client**

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

▶ **Connection details** Connected
[Disconnect](#)

To disconnect from the MQTT test client, choose Disconnect. To re-establish a connection, you can update the details and choose Connect.

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

Topic filter Info
The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

Additional configuration

[Subscribe](#)

Subscriptions

You have no topic subscriptions.

Topic

Subscribe or select a topic to view incoming messages.

Now subscribe to the required publishers

Subscriptions

| | |
|-----------------------|-----|
| home/bed/ir | ♥ X |
| home/bed/us | ♥ X |
| home/hygine/dustbin | ♥ X |
| home/hygine/laundry | ♥ X |
| home/env/dust | ♥ X |
| home/env/co2 | ♥ X |
| room/env/co2 | ♥ X |
| room/bottle/hydration | ♥ X |
| room/bottle/level | ♥ X |
| room/# | ♥ X |
| home/# | ♥ X |
| home/env/+ | ♥ X |
| +/env/co2 | ♥ X |

Now publish messages

Subscribe to a topic **Publish to a topic**

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

X

Message payload

```
{
  "message": "person still in bed"
}
```

► Additional configuration

Publish

Subscriptions

| | |
|-----------------------|-----|
| home/bed/ir | ✓ X |
| home/bed/us | ✓ X |
| home/hygine/dustbin | ✓ X |
| home/hygine/laundry | ✓ X |
| home/env/dust | ✓ X |
| home/env/co2 | ✓ X |
| room/env/co2 | ✓ X |
| room/bottle/hydration | ✓ X |
| room/bottle/level | ✓ X |
| room/# | ✓ X |
| home/# | ✓ X |

home/bed/ir Pause Clear Export Edit

Message payload

```
{
  "message": "Hello from AWS IoT console"
}
```

► Additional configuration

Publish

Properties

February 12, 2026, 13:10:35 (UTC+0530)

```
{
  "message": "person still in bed"
}
```

Subscribe to a topic **Publish to a topic**

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

X

Message payload

```
{
  "message": "co2 levels high"
}
```

► Additional configuration

Publish

Subscriptions

| | |
|-----------------------|-----|
| home/bed/ir | ✓ X |
| home/bed/us | ✓ X |
| home/hygine/dustbin | ✓ X |
| home/hygine/laundry | ✓ X |
| home/env/dust | ✓ X |
| home/env/co2 | ✓ X |
| room/env/co2 | ✓ X |
| room/bottle/hydration | ✓ X |
| room/bottle/level | ✓ X |
| room/# | ✓ X |
| home/# | ✓ X |
| home/env/+ | ✓ X |
| +/env/co2 | ✓ X |

room/env/co2 Pause Clear Export Edit

Message payload

```
{
  "message": "Hello from AWS IoT console"
}
```

► Additional configuration

Publish

Properties

February 12, 2026, 13:11:44 (UTC+0530)

```
{
  "message": "co2 levels high"
}
```

Properties

February 12, 2026, 13:11:44 (UTC+0530)

```
{
  "message": "co2 levels high"
}
```

The screenshot shows the AWS IoT console interface. At the top, there are two tabs: "Subscribe to a topic" and "Publish to a topic", with "Publish to a topic" being the active tab. Below the tabs, there is a section for "Topic name" with a text input field containing "home/hygine/dustbin". A message payload editor shows the JSON message: {"message": "dustbin nearly full"}. There is a "Message payload" section with a "Publish" button. On the left, a sidebar titled "Subscriptions" lists various topics with heart and X icons. The "home/hygine/dustbin" topic is selected, and its details are shown in a central panel. This panel includes a "Message payload" section with the message content, a "Pause", "Clear", "Export", and "Edit" button, and an "Additional configuration" section with a "Publish" button. Below this, a detailed view of the subscription shows the message content and a timestamp: "February 12, 2026, 13:13:22 (UTC+0530)".

Using wildcards
I used the following wildcards

| | |
|------------|-----|
| room/# | ♡ X |
| home/# | ♡ X |
| home/env/+ | ♡ X |
| +/env/co2 | ♡ X |

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

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

[X](#)

Message payload

```
{
  "message": "dust levels in the air is low"
}
```

[► Additional configuration](#)

[Publish](#)

Subscriptions

| | |
|-----------------------|-----|
| home/bed/ir | ♡ X |
| home/bed/us | ♡ X |
| home/hygiene/dustbin | ♡ X |
| home/hygiene/laundry | ♡ X |
| home/env/dust | ♡ X |
| home/env/co2 | ♡ X |
| room/env/co2 | ♡ X |
| room/bottle/hydration | ♡ X |
| room/bottle/level | ♡ X |
| room/# | ♡ X |
| home/# | ♡ X |
| home/env/+ | ♡ X |
| +/env/co2 | ♡ X |

home/#

[Pause](#) [Clear](#) [Export](#) [Edit](#)

home/env/dust February 12, 2026, 13:14:30 (UTC+0530)

```
{
  "message": "dust levels in the air is low"
}
```

[► Properties](#)

home/env/dust February 12, 2026, 13:14:30 (UTC+0530)

```
{
  "message": "dust levels in the air is low"
}
```

[► Properties](#)

Sending to home/env/dust sends it to home/# and home/env/+ as well

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

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

[X](#)

Message payload

```
{  
    "message": "70% water remaining"  
}
```

[▶ Additional configuration](#)

[Publish](#)

Subscriptions

| | |
|--|-----|
| home/bed/ir | ♡ X |
| home/bed/us | ♡ X |
| home/hygine/dustbin | ♡ X |
| home/hygine/laundry | ♡ X |
| home/env/dust | ♡ X |
| home/env/co2 | ♡ X |
| room/env/co2 | ♡ X |
| room/bottle/hydration | ♡ X |
| <input type="checkbox"/> room/bottle/level | ♡ X |
| <input checked="" type="checkbox"/> room/# | ♡ X |
| home/# | ♡ X |
| <input type="checkbox"/> home/env/+ | ♡ X |
| +/env/co2 | ♡ X |

room/#

[Pause](#) [Clear](#) [Export](#) [Edit](#)

room/bottle/level February 12, 2026, 13:15:56 (UTC+0530)

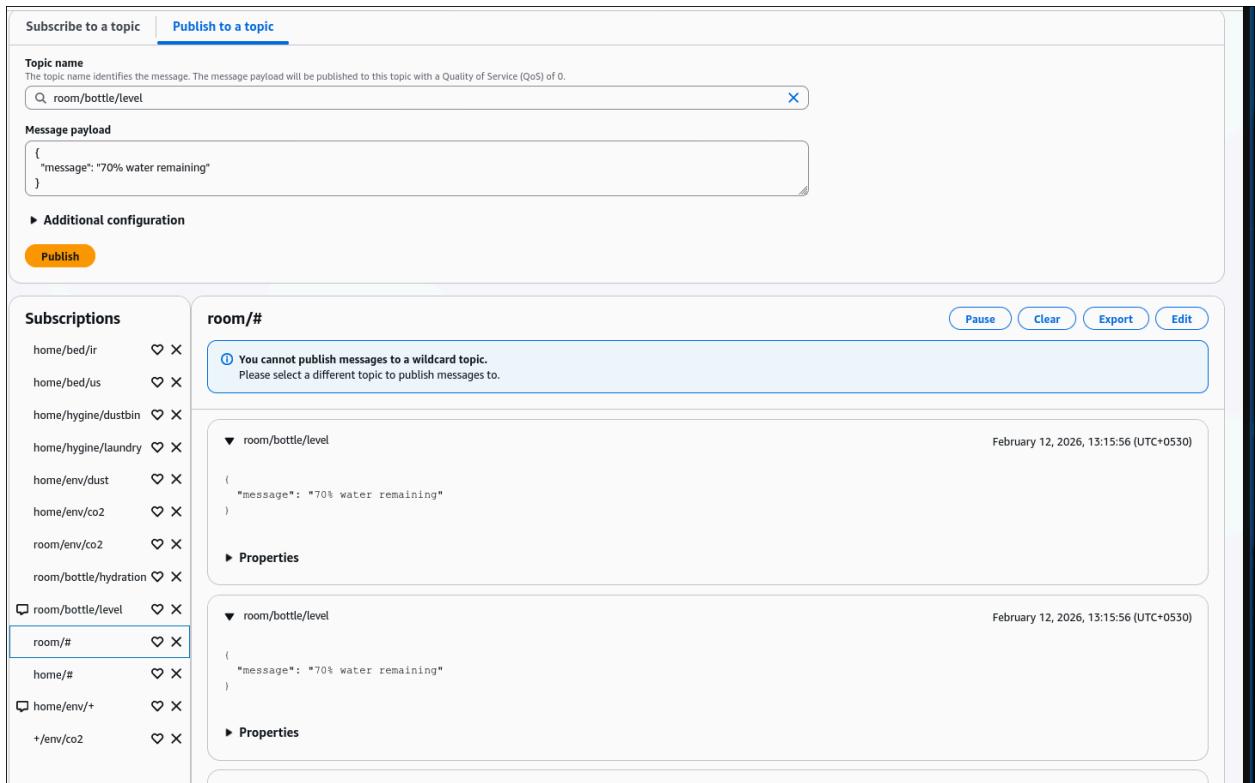
```
{  
    "message": "70% water remaining"  
}
```

[▶ Properties](#)

room/bottle/level February 12, 2026, 13:15:56 (UTC+0530)

```
{  
    "message": "70% water remaining"  
}
```

[▶ Properties](#)



Sending to /room/bottle/level also sends to room/#