IoT Real Time Analytics Assessment | NCI

Explore the docs » Report Bug

Building a Cloud-Based IoT System with PI Sense Hat SIM, Node-RED and AWS DynamoDB

This guide will walk you through the steps to build a cloud-based IoT system using Node-RED and AWS DynamoDB. By following these steps, you will be able to access sensor values, publish them to the cloud, and retrieve and graph the findings in a web application.

To complete the steps mentioned in the below message, you will need to install the following Node-RED nodes:

1. `node-red-dashboard:` This node provides a set of pre-built UI components for creating a web dashboard to visualize and control IoT data.
2. `node-red-contrib-aws:` This node provides a set of nodes for connecting Node-RED to AWS services, such as AWS IoT Core, AWS S3, AWS DynamoDB, and more.
3. `node-red-contrib-sense-hat:` This node provides a set of nodes for working with the Sense HAT hardware on a Raspberry Pi, such as reading temperature, humidity, pressure, and orientation data, displaying messages on the LED matrix, and more.

To install these nodes, follow these steps:

Open the Node-RED editor.

1. Click on the hamburger menu on the top-right corner of the editor and select "Manage palette".
2. Click on the "Install" tab.
3. Search for each of the three nodes mentioned above.
4. Click the "Install" button next to each node.
5. Wait for the installation to complete.
6. Close the "Manage palette" window.

Once you have installed these nodes, you will be able to complete the steps mentioned in the below message.

Step by Step Guide

Step 1: Set Up AWS IoT Core

Create an AWS account if you don't have one already. Go to the AWS IoT Core console and create a new Thing. Download the Thing's certificate, public key, and private key. Create an IoT policy that grants the Thing permission to publish and subscribe to the desired topics. Attach the policy to the Thing.

Here are the step-by-step technical details for creating an AWS IoT Thing, downloading the Thing's certificate, public key, and private key, creating an IoT policy, and attaching the policy to the Thing:

1. Create an AWS account if you don't have one already by going to the AWS website and following the prompts to sign up.
2. Once you have an AWS account, go to the AWS IoT Core console by navigating to the "Services" menu and searching for "IoT Core" in the search bar.
3. Click on "IoT Core" to open the console.
4. In the IoT Core console, click on "Manage" in the left-hand menu, and then click on "Things." Click on the "Create" button to create a new Thing.
5. Enter a name for your Thing, and optionally add any attribute key-value pairs. Click on "Create Thing" to create the Thing.
6. Once the Thing has been created, click on its name to open its details page.
7. On the Thing's details page, click on "Security" in the left-hand menu.
8. Click on the "Create a certificate" button to create a new certificate for the Thing.
9. Click on the "Download" button for each of the following to download the Thing's certificate, public key, and private key:

- Certificate: This is the certificate that your Thing will use to authenticate with AWS IoT Core.
- Public key: This is the public key that your Thing will use to encrypt messages.
- Private key: This is the private key that your Thing will use to decrypt messages.

10. Click on "Activate" to activate the certificate.
11. Attach a policy to the Thing to grant it permission to publish and subscribe to desired topics.
12. Click on "Attach a policy" and enter a name for the policy.

13. In the policy editor, enter the necessary details to grant the Thing permission to publish and subscribe to desired topics. For example, you can use the following policy to allow the Thing to publish and subscribe to a topic named “my/topic”:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Subscribe",
                "iot:Connect",
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:<account_id>:topic/my/topic"
            ]
        }
    ]
}
```

Note: Replace “” with your AWS account ID.

Click on “Create” to create the policy and attach it to the Thing. That’s it! You have now created an AWS IoT Thing, downloaded its certificate, public key, and private key, created an IoT policy, and attached the policy to the Thing.

2: Set Up Node-RED

Install Node-RED on your computer or a Raspberry Pi. Open the Node-RED editor and create a new flow. Drag an “inject” node onto the workspace and configure it to send a message at a desired interval. Drag a “Sense HAT” node onto the workspace and configure it to read sensor data from the Sense HAT sim. Connect the output of the “Sense HAT” node to an “aws-iot” node. Drag an “aws-iot” node onto the workspace and double-click it to configure the AWS IoT Core settings. Enter the Thing’s certificate, public key, and private key, and select the desired topic to publish to. Connect the output of the “Sense HAT” node to the input of the “aws-iot” node. Deploy the Node-RED flow.

Here are the step-by-step technical details for setting up Node-RED to read sensor data from Sense HAT sim and publish it to AWS IoT Core:

1. Install Node-RED on your computer or Raspberry Pi. You can follow the instructions for your specific platform from the official Node-RED documentation: https://nodered.org/docs/getting-started

2. Open the Node-RED editor by going to http://localhost:1880 (if Node-RED is installed on your local computer) or http://:1880 (if Node-RED is installed on a Raspberry Pi).

3. Create a new flow by clicking on the "+" icon in the top-right corner of the Node-RED editor.

4. Drag an "inject" node from the left-hand palette onto the workspace.

5. Double-click the "inject" node to open its configuration dialog. Set the "Payload" to "timestamp" and set the "Repeat" interval to the desired value (e.g., 5 seconds).

6. Drag a "Sense HAT" node from the left-hand palette onto the workspace.

7. Double-click the "Sense HAT" node to open its configuration dialog. Select "Sense HAT Sim" from the "Device" dropdown menu.

8. Connect the "output" of the "Sense HAT" node to the "input" of the "aws-iot" node.

9. Drag an "aws-iot" node from the left-hand palette onto the workspace.

10. Double-click the "aws-iot" node to open its configuration dialog.

11. Click the pencil icon next to the "AWS IoT Broker" field to open the AWS IoT Core settings dialog.

12. In the AWS IoT Core settings dialog, enter the following information:

- AWS Region: select the region where you created your AWS IoT Core resources (e.g., "us-east-1").
- Thing Name: enter the name of the Thing that you created in Step 1.
- Certificate: click the "pencil" icon and paste in the contents of the certificate file that you downloaded in Step 1.
- Private Key: click the "pencil" icon and paste in the contents of the private key file that you downloaded in Step 1.
- Node ID: enter a unique identifier for this Node-RED instance (e.g., "node-red-01").
- Enable Keep Alive: check the box to enable keep-alive messages.

13. In the "Message Topic" field, enter the desired topic to publish sensor data to (e.g., "sensors/temp").

14. Connect the "output" of the "Sense HAT" node to the "input" of the "aws-iot" node.

15. Deploy the Node-RED flow by clicking the "Deploy" button in the top-right corner of the editor.

Now, Node-RED should be sending simulated sensor data from the Sense HAT sim to AWS IoT Core at the specified interval. You can verify that the data is being received by checking the AWS IoT Core console.

Step 3: Set Up AWS DynamoDB

Go to the AWS DynamoDB console and create a new table. Choose a unique table name and a primary key. Configure any desired settings, such as read and write capacity units. Save the table.

Here are the step-by-step technical details to set up AWS DynamoDB:

1. Go to the AWS Management Console and log in to your account.
2. In the console search bar, type “DynamoDB” and select the “DynamoDB” service.
3. Click on the “Create table” button.
4. Enter a unique table name in the “Table name” field.
5. Choose a primary key for your table. You can either select a partition key, a combination of partition key and sort key or use no sort key.
6. Configure any additional desired settings, such as read and write capacity units, provisioned throughput, global secondary indexes or tags.
7. Click the “Create” button to create the table.

Your DynamoDB table is now set up and ready to be used.

Step 4: Configure the AWS IoT Rule

Go to the AWS IoT Core console and create a new rule. Configure the rule to match the desired topic that the Node-RED flow is publishing to. Choose an action to take when a message matches the rule. Select “Send a message to a DynamoDB table” and choose the table that you created in step 3. Configure the mapping between the message payload and the DynamoDB item attributes. Save the rule.

Here are the step-by-step technical details for configuring the AWS IoT rule:

Step 1: Log in to the AWS IoT Core console using your AWS account credentials.

Step 2: Click on “Create a rule” on the left-hand side of the screen.

Step 3: Enter a name for your rule in the “Rule name” field.

Step 4: In the “Rule query statement” section, enter the topic that your Node-RED flow is publishing to. For example, if your flow is publishing to a topic called “sensors/temperature,” enter that topic here.

Step 5: In the “Set one or more actions” section, select “Send a message to a DynamoDB table” from the dropdown menu.

Step 6: Choose the table that you created in step 3 from the “Choose a target DynamoDB table” dropdown menu.

Step 7: In the “Configure payload” section, you can choose to use the entire message payload or select specific fields to be included in the DynamoDB item attributes. You can also set data types and transformations as needed.

Step 8: Click on “Add action” to save the rule.

Your AWS IoT rule is now configured to send messages from your Node-RED flow to the DynamoDB table you created in step 3.

Step 5: Verify Data in DynamoDB

Here are the step-by-step technical details for verifying data in DynamoDB:

1. Wait for the Node-RED flow to publish some simulated sensor data to AWS IoT Core.
2. Go to the AWS DynamoDB console and select the table that you created in step 3.
3. Click on the “Items” tab to see the list of items in the table.
4. Check if the simulated sensor data is being written to the table by verifying the values in the items.
5. You can also use the “Query” tab to search for specific items based on the item attributes.
6. If you see the simulated sensor data in the table, then the integration between Node-RED and AWS DynamoDB is working correctly.

Wait for the Node-RED flow to publish some sensor data to AWS IoT Core. Go to the AWS DynamoDB console and verify that the data is being written to the table.

Step 6: Visualize the Data in Node-RED

Create a new flow in Node-RED that retrieves the sensor data from the DynamoDB table and visualizes it in a graph. Drag a “dynamodb” node onto the workspace and configure it to read from the table you created in step 3. Choose the desired key and any optional filtering or sorting options. Connect the output of the “dynamodb” node to a function node that extracts the sensor data from the DynamoDB item and formats it as an array. Connect the output of the function node to a “ui_chart” node, which can be used to display the sensor data as a line chart. Configure the chart with the desired settings. Deploy the Node-RED flow and open the Node-RED dashboard to view the chart.

That’s it! You have now built a cloud-based IoT system that accesses sensor values from the Sense HAT sim, publishes them to the cloud, and retrieves and graphs the findings in a web application using Node-RED and AWS DynamoDB.

1. Open the Node-RED editor and create a new flow.

2. Drag a “dynamodb” node from the Node-RED palette onto the workspace.
3. Double-click the “dynamodb” node and configure it to read from the table you created in step 3. Enter the desired key and any optional filtering or sorting options.
4. Drag a “function” node from the Node-RED palette onto the workspace and connect it to the “dynamodb” node.
5. Write some JavaScript code in the “function” node that extracts the sensor data from the DynamoDB item and formats it as an array. For example, you can use the following code:

```
msg.payload = msg.payload.Items.map(item => {
  return {
    timestamp: new Date(item.timestamp.S).getTime(),
    value: parseFloat(item.value.N)
  };
});
return msg;
```

This code extracts the timestamp and value from each DynamoDB item, converts the timestamp to a Unix timestamp in milliseconds, and formats the data as an array of objects.

6. Drag a “ui_chart” node from the Node-RED dashboard palette onto the workspace and connect it to the output of the “function” node.
7. Double-click the “ui_chart” node and configure it with the desired settings for the chart, such as the chart type, title, and axis labels.
8. Deploy the Node-RED flow and open the Node-RED dashboard to view the chart.

Congratulations! You have now visualized the sensor data retrieved from AWS DynamoDB in a chart using Node-RED.