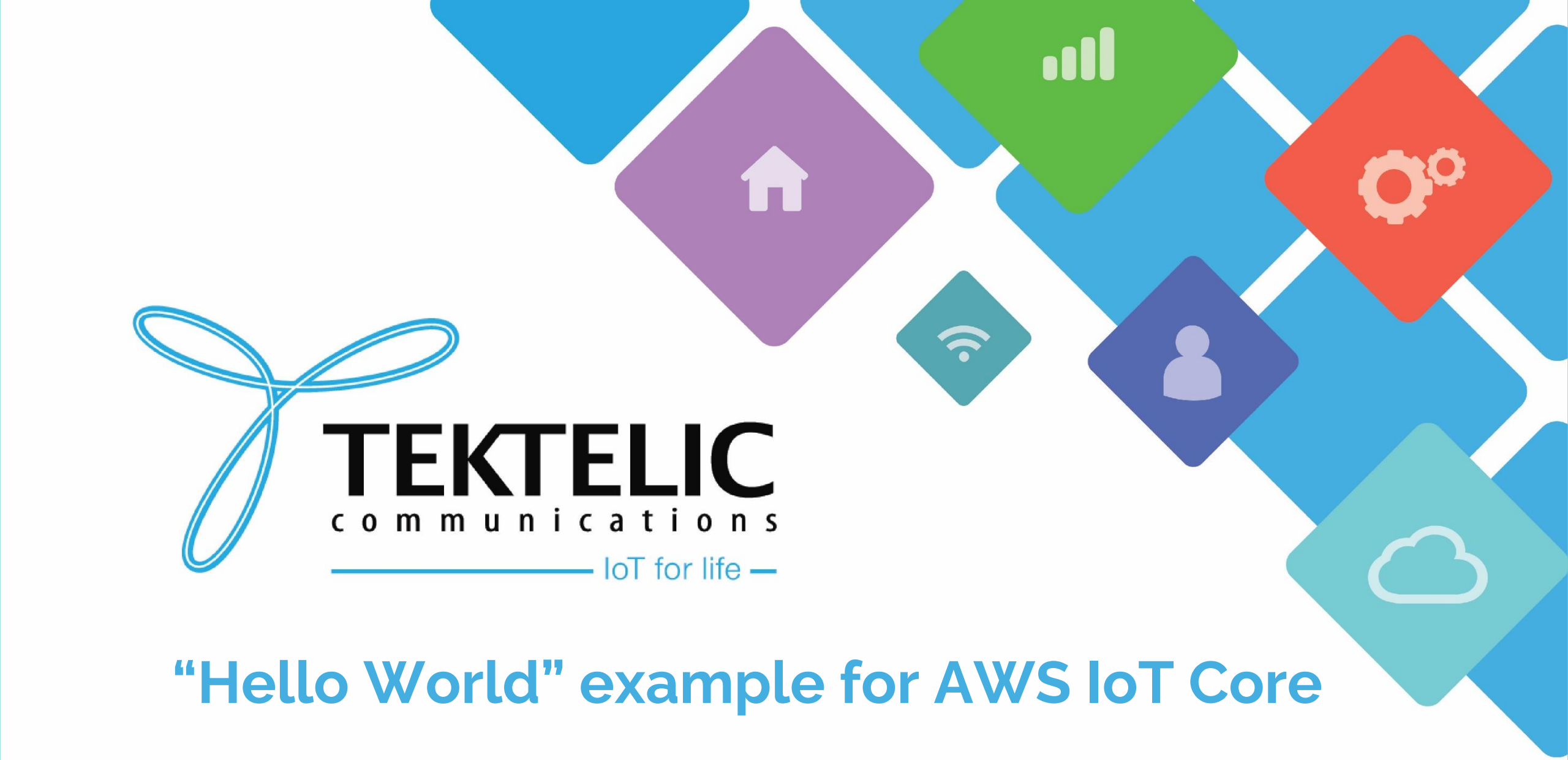

“Hello World” example for AWS IoT Core

Introduction

- Reference guide to verify operations
- List requirements.

- High-level procedure steps:

1. Create Lambda function for destination rule
2. Create the Destination rule
3. Configure SNS
4. Create and Configure IoT Analytics

Requirements

- Gateway connected to AWS
- Sensor connected to AWS
- AWS account

1. Create Lambda function for destination rule

Once the initial setup is complete, provisioned OTAA devices can join the network and start to send messages. Messages from OTAA devices can then be received by AWS IoT Core and then forwarded to the IoT Rules Engine.

1.1 Create Lambda function for destination rule

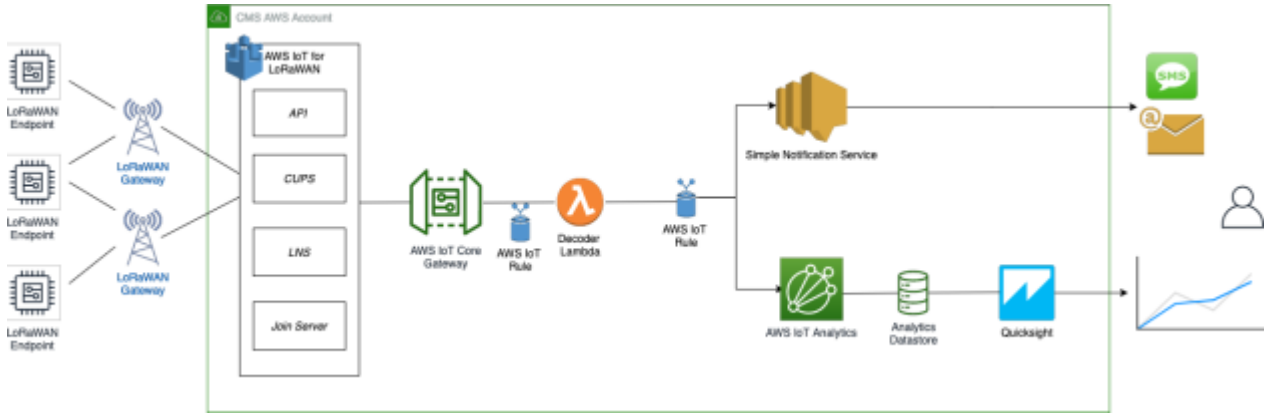

Figure-1 AWS Topology

1.1 Create Lambda function for destination rule

1. Go to AWS IoT Core.
2. In the left navigation pane, select Settings and note the Endpoint URL.
3. In a new browser tab, go to the AWS Lambda console.
4. In the left navigation pane, select Functions.
5. Select Create function.
6. In a new browser tab, click here and copy the code for the lambda function.
7. Under Function code, paste the code from Step 6 into the editor under the index.js tab.
8. At the top of the index.js text, change ***INSERT_ENDPOINT_URL_HERE*** with the endpoint URL from step 2.
9. Once the code has been pasted, select “Deploy” to deploy the lambda code.

1.2 Change Lambda Role Policy permission

10. Under the Configuration tab, select the Permissions tab of the lambda function.
11. Under Execution role, click on the hyperlink under Role name.
12. On the Permissions tab, find the policy name and click on it.
13. Select Edit policy, then select the JSON tab.
14. Append the following to the Statement section of the policy to allow publishing to AWS IoT:

```
{
    "Effect": "Allow",
    "Action": [
        "iot:Publish”
    ],
    "Resource": [
        "*“
    ]
}
```

15. Once appended, select Review policy, then select Save changes.

1.2 Change Lambda Role Policy permission

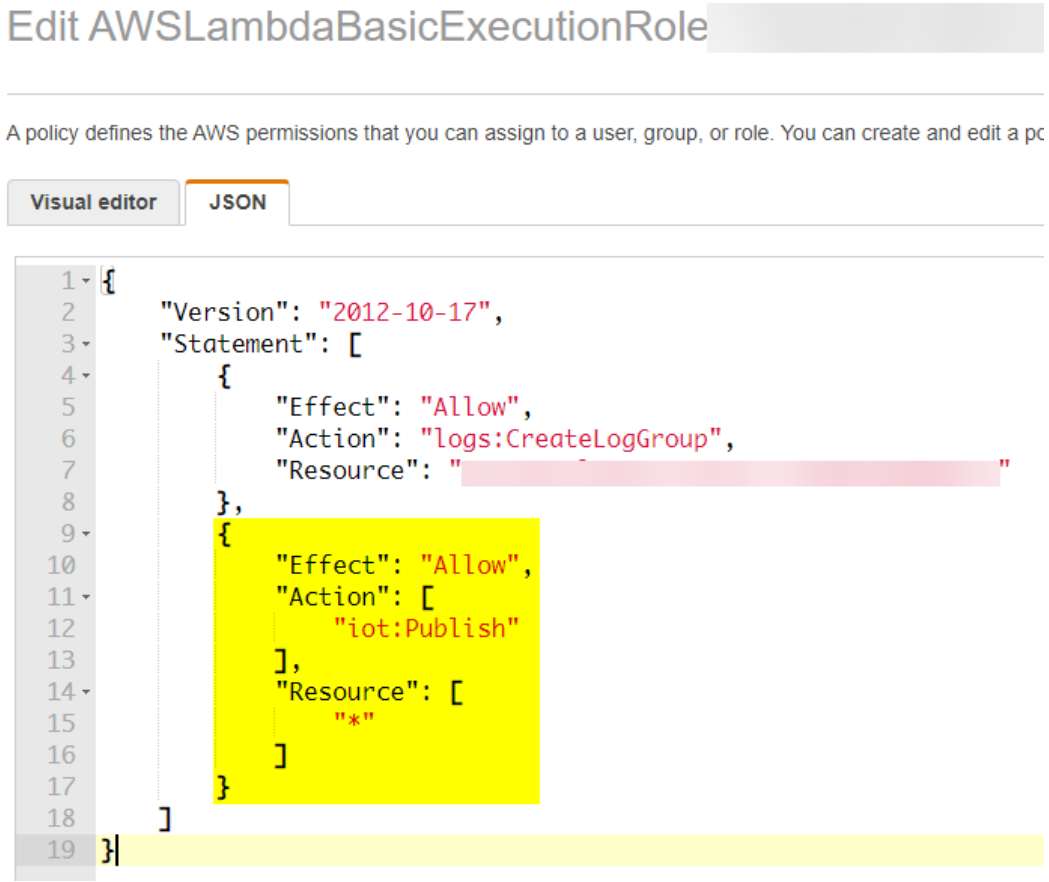

Figure-2 Appended Statement

1.3 Create test event for Lambda function

1. In the Code section of the Lambda function, select the drop-down menu for Test and select Configure test event.
2. Create a new event and enter a name under Event name.
3. Paste the following sample payload in the Event JSON section:

```
{
  "MessageId": "55d122ab-6355-2233-9874-ff47c5222108",
  "WirelessDeviceId": "65d128ab-90dd-4668-9556-fe47c589610b",
  "PayloadData": "AgAAA2cA3QRoLA==",
  "WirelessMetadata": {
    "LoRaWAN": {
      "DevEui": "647FDAXXXXXXXXX",
      "FPort": 10,
      "DataRate": 0,
      "Frequency": 904500000,
      "Gateways": [
        {
          "GatewayEui": "647FDAXXXXXXXXX",
          "Snr": 12.25,
          "Rssi": -47
        }
      ],
      "Timestamp": "2020-11-10T20:23:56Z"
    }
  }
}
```

4. Select Save to save event.

1.3 Create test event for Lambda function

5. Navigate to AWS IoT Core and on the navigation pane, under Test, select MQTT client.
6. Under Topic filter, enter “#” and then select Subscribe to listen for all events.
7. Under the Lambda function, click on Test to generate the test event you’ve just created.
8. Verify the published data in the MQTT Client. Output should look similar to the one found in the next slide.

1.3 Create test event for Lambda function

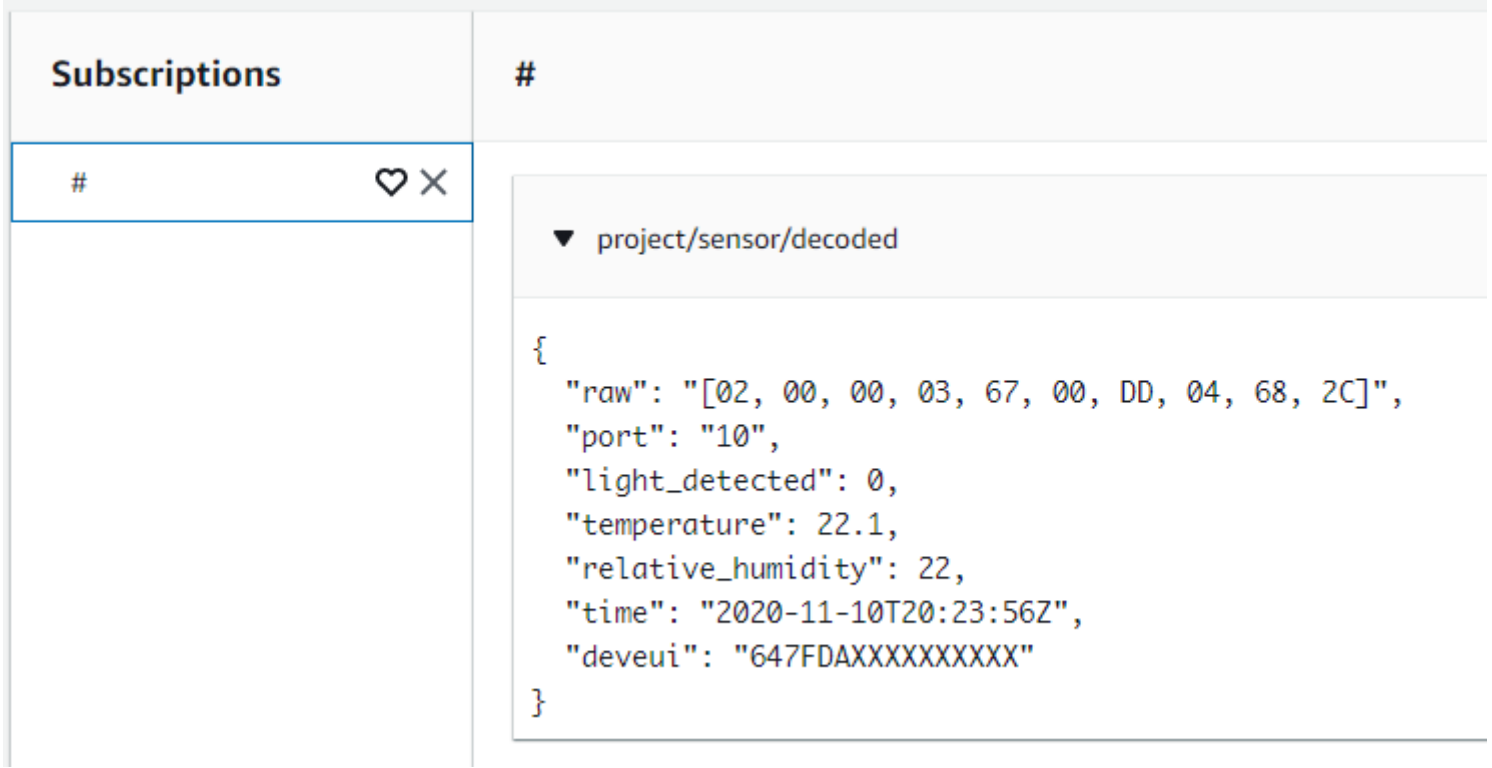

Figure-3 Test event output

2. Create Destination Rule

This section will create the IoT rule that forwards the device payload to your application. This rule is associated with the destination created earlier in section 4.3.

2. Create Destination Rule

1. Navigate to AWS IoT Core.
2. In the navigation pane, select Message Routing, then select Rules
3. On the Rules page, select Create rule.
4. For rule name, enter LoRaWANRouting. For Description, enter a description of your choice. Note the name of your rule. This information will be needed when you provision devices to run on AWS IoT Core. Select Next to continue.
5. In the SQL statement section, paste the following line:
   `SELECT * FROM 'iot/topic'`
6. Select Next to proceed.

2. Create Destination Rule

7. Under the Rule actions section, select Republish to AWS IoT topic for Action 1.
8. Under Topic, enter the following:
   `project/sensor/observed`
9. Under IAM Role, select Create new Role with a name of your choice.
10. Add another rule action and select Lambda.
11. For function, select the one you created earlier in section 1.
12. Finalize by selecting Next, then select Create.

2. Create Destination Rule

You can now check if decoded data is received and republished by AWS by triggering a condition or event on the device itself.

To see the data, please follow the instructions found in slide 10. For an example, please see slide 11.

3. Configuring SNS

In this section, you will configure Amazon's Simple Notification Service to send text messages when certain conditions are met.

3.1 Configuring SNS

1. Go to the AWS SNS console.
2. In the left navigation pane, select Text Messaging (SMS) and select Publish text message.
3. Under Message Type, select Promotional.
4. Under Destination phone number, enter your phone number.
5. Under Message, enter “Test message”.
6. If the phone number you entered is valid, you will receive a text message and your phone number will be confirmed.

3.2 Create SNS Topic

1. In the left navigation pane, select Topics.
2. Select Create topic.
3. Under Type, select Standard.
4. For name, enter a name of your choice. For example: “text_topic”.
5. Optional – configure Access policy.
6. Under Choose method, select Basic.
7. Under Define who can send messages to this topic, select Everyone.
8. Finalize by selecting Create topic.

3.3 Create SNS Subscription

1. In the page for your newly created SNS Topic, select the Subscriptions tab.
2. Then, select Create subscription.
3. Under Protocol, select SMS from the drop-down.
4. Under Endpoint, enter the previously validated phone number to receive SMS alerts.
5. Finalize by selecting Create subscription.

3.4 Add rule for SNS notification

1. Navigate to AWS IoT Core.
2. In the navigation pane, select Message Routing, then select Rules
3. On the Rules page, select Create rule.
4. For Name, enter text_alert and provide an appropriate Description.
5. Select Next. Under SQL statement, copy and paste the following:
```
SELECT deveui, "Temperature exceeded 20" as message, relative_humidity, temperature, time FROM 'project/sensor/decoded' where temperature > 20
```
6. Select Next. Under Rule actions, for Action 1 select Simple Notification Service (SNS).
7. For SNS topic, select the topic you created in slide 18.
8. For IAM role, choose or create a role to grant AWS IoT access to perform this action.
9. Finalize by selecting Create.

3.4 Add rule for SNS notification

Figure-4 Example Text Notification

4. IoT Analytics

We will use IoT Analytics to visually display data via graphs if there is a need in the future to do further analysis.

4.1 Create an IoT Analytics rule.

1. Navigate to AWS IoT Core.
2. In the navigation pane, select Message Routing, then select Rules
3. On the Rules page, select Create rule.
4. For rule name, enter Visualize. For Description, provide an appropriate description.
5. In the SQL statement section, paste the following line:
   `SELECT * FROM 'project/sensor/decoded'`
6. Under Rule Actions, select IoT Analytics for Action 1.
7. Create IoT Analytics channel:
   a. Enter a channel name of your choice.
   b. Choose a Storage type of your choice.
   c. Choose or create a S3 bucket of your choice.
   d. Choose or create an IAM role that has access to the bucket.
   e. Finalize by selecting Next, then Create channel.
8. Choose or create IAM role with a name of your choice.
9. Finalize by selecting Create.

4.2 Configure IoT Analytics

1. Navigate to the AWS IoT Analytics console
2. In the left navigation pane, select Datasets
3. Select the data set that was generated in the slide 22.
4. Under the details section, edit the SQL query with the following:
   ```
   select temperature, relative_humidity, deveui, time from [datastore you've created]
   ```
5. Under the Schedule section, select edit. Choose Every 1 minute for Frequency.
6. Finalize by selecting Save.

4.3 Configure QuickSight

1. Navigate to the AWS Management console.
2. From the management console, enter “QuickSight” in the “Search for services, features..” search box.
3. If you haven’t signed up for the service before, go ahead and sign up as there is a free trial period.
4. Select Standard Edition, then select Continue.
5. Enter a unique name in the field QuickSight account name.
6. Fill in the notification email address.
7. Review the other checkbox options and change them as necessary. The AWS IoT Analytics option must be selected.
8. Select Finish. You will see a confirmation message.

4.3 Configure QuickSight

9. Select Go to Amazon QuickSight.
10. Select Datasets.
11. Select New dataset.
12. Select AWS IoT Analytics.
13. Enter a name of your choice, then select the dataset created in slide 22.
14. Select Create data source, then finalize by selecting Visualize.
15. Select the dataset created, then select Refresh or Schedule Refresh for periodic refresh of the dataset.

4.4 Testing your “Hello World” Application

Using your device, we have created a condition to generate an event such as a high temperature condition. If the temperature is above the configured threshold, then you will receive a text alert on your phone. This alert will include key parameters about the alert.

4.4 Testing your “Hello World” Application

1. Go to the AWS IoT Analytics console.
2. Select Datasets.
3. Select the dataset created earlier.
4. Select Content an ensure that there are at least a few uplink entries available in the dataset.
5. Go to the QuickSight console.
6. Select New analysis.
7. Select the dataset created in slide 22.
8. Select time on the X-axis, Value as Temp (average) and Color as device_id to see a chart of your dataset.

4.4 Testing your “Hello World” Application

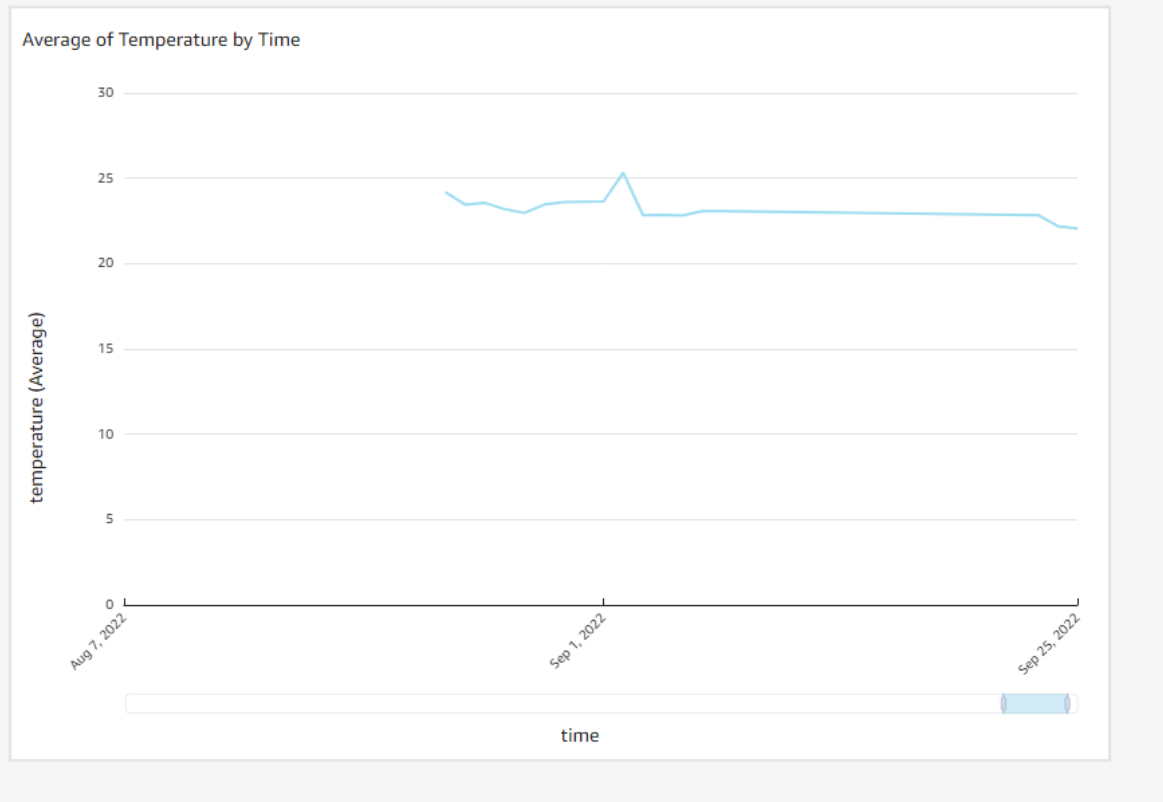

Figure-5 Example Visualization

Best-In-Class, Carrier Grade & Most Cost Effective Portfolio of Gateways, Network Server, Sensors & Applications