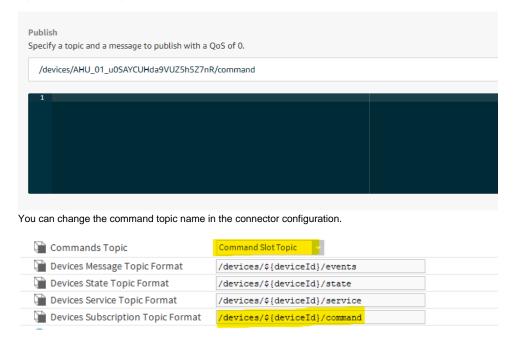
Step 5 Send commands from AWS to Niagara

AWS Connector supports cloud to niagara communication via mqtt messages to control points remotely.

Send a point action command

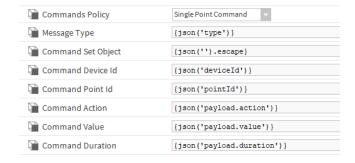
1. Go to the AWS IoT Core test application then put the topic name of the device you want to send commands to. (make sure to use the commands topic not the shadow topic)



- 2. By default we use this message template for **POINT_ACTION** command. by you can use any format that meat your needs. check the connector advanced settings.
 - a. This is the default command template.

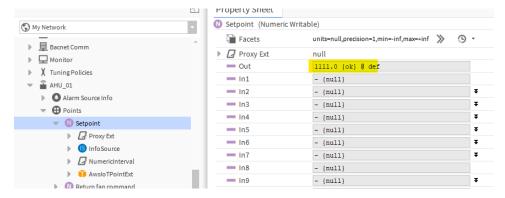
```
{
    "type": "POINT_ACTION",
    "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",
    "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
    "payload": {
        "action": "SET",
        "value": 1111
    }
}
```

b. This is the default connector point action command configuration.



3. On the left chose Cloud To Device Message.

- 4. Then hit Send Message.
- 5. On the niagara you should see the point value applied successfully.



6. And the new value sent to the cloud.

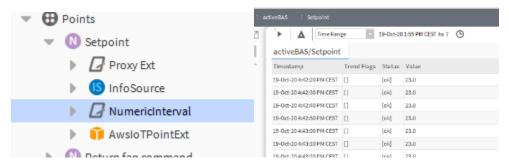
```
/devices/AHU_01_u0SAYCUHda9VUZ5h5Z7nR/... October 19, 2020, 18:00:39 (UTC+02

{
    "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
    "timestamp": "2020-10-19T18:00:39.827+02:00",
    "value": 1111,
    "status": "{ok} @ def"
}
```

Send a send history command

You can pull historical value for any point that has a history associated by sending a **SEND_HISTORY** command.

1. Add a history. extension to the point.



2. Go to the aws and send the command to the device.

- 3. By default we use this message template for send history command. by you can use any format that meat your needs. check the connector advanced settings.
 - a. This is the default command template.

b. This is the default connector send history command configuration.



4. And you should see the messages being sent.

```
/devices/AHU_01_u0SAYCUHda9VUZ5h5Z7nR/... October 20, 2020, 10:37:31 (UTC+0200)

{
    "startTimestamp": "2020-10-19T18:15:20.294+02:00",
    "endTimestamp": "null",
    "deviceId": "AHU 01_u0SAYCUHda9VUZ5h5Z7nR",
    "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
    "trendsFlags": "0",
    "status": "{ok}",
    "value": 1111,
    "count": null,
    "mar": null,
    "awg": null,
    "sum": null,
    "sum": null
}

/devices/AHU_01_u0SAYCUHda9VUZ5h5Z7nR/... October 20, 2020, 10:37:31 (UTC+0200)

{
    "startTimestamp": "2020-10-19T18:15:30.632+02:00",
    "endTimestamp": "AHU 01 u0SAYCUHda9VUZ5h5Z7nR".
```

To change the message format check the connector advanced setting then history message template

```
S(startTimestamp)
S(endTimestamp)
S(pointId)
S(deviceId)
S(status)
S(value)
S(trendsFlags)
S(count)
S(min)
S(max)
S(avg)
S(sum)

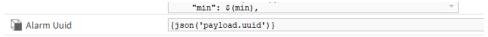
{
    "startTimestamp": "$(startTimestamp)",
    "endTimestamp": "$(endTimestamp)",
    "deviceId": "$(deviceId)",
    "pointId": "$(pointId)",
    "trendsFlags": "$(trendsFlags)",
    "status": $(value),
    "count": $(count),
    "min": $(min).
```

Send ack alarm command

You can ack alarms by sending an ACK_ALARM command to any alarm recipient device.

- By default we use this message template for ack alarm command. by you can use any format that meat your needs. check the connector advanced settings.
 - a. This is the default command template.

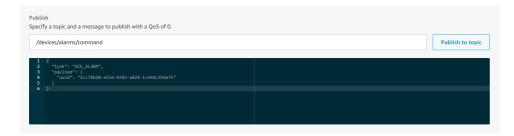
b. This is the default connector send history command configuration.



2. Go the alarms console and pick an unacked alarm id.



3. Go to device and send the command.



4. Go back to the console and you should see that the alarm has been acked.

