

# 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)

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

`/devices/AHU_01_u0SAYCUHda9VUZ5h5Z7nR/command`

1

You can change the command topic name in the connector configuration.

Commands Topic	Command Slot Topic
Devices Message Topic Format	/devices/\${deviceId}/events
Devices State Topic Format	/devices/\${deviceId}/state
Devices Service Topic Format	/devices/\${deviceId}/service
Devices Subscription Topic Format	/devices/\${deviceId}/command

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.

Commands Policy	Single Point Command
Message Type	{json('type')}
Command Set Object	{json('').escape}
Command Device Id	{json('deviceId')}
Command Point Id	{json('pointId')}
Command Action	{json('payload.action')}
Command Value	{json('payload.value')}
Command Duration	{json('payload.duration')}

3. On the left chose Cloud To Device Message.

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

/devices/AHU\_01\_u0SAYCUHda9VUZ5h5Z7nR/command

```

1 {
2   "type": "POINT_ACTION",
3   "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",
4   "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
5   "payload": {
6     "action": "SET",
7     "value": 1111
8   }
9 }

```

- Then hit **Send Message**.
- On the Niagara you should see the point value applied successfully.

My Network

- Bacnet Comm
- Monitor
- Tuning Policies
- AHU\_01
  - Alarm Source Info
  - Points
    - Setpoint**
      - Proxy Ext
      - InfoSource
      - NumericInterval
      - AwsloTPointExt
      - Return fan command

Property sheet

Setpoint (Numeric Writable)

Facets units=null,precision=1,min=-inf,max=+inf

Proxy Ext null

Out	1111.0 {ok} @ def
In1	{null}
In2	{null}
In3	{null}
In4	{null}
In5	{null}
In6	{null}
In7	{null}
In8	{null}
In9	{null}

- 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.

- Add a history. extension to the point.

Points

- Setpoint
  - Proxy Ext
  - InfoSource
  - NumericInterval**
  - AwsloTPointExt
  - Return fan command

activeBAS : Setpoint

Time Range 19-Oct-20 1:35 PM CEST to ?

Timestamp	Trend Flags	Status	Value
19-Oct-20 4:42:00 PM CEST	[ ]	{ok}	23.0
19-Oct-20 4:42:30 PM CEST	[ ]	{ok}	23.0
19-Oct-20 4:42:40 PM CEST	[ ]	{ok}	23.0
19-Oct-20 4:42:50 PM CEST	[ ]	{ok}	23.0
19-Oct-20 4:43:00 PM CEST	[ ]	{ok}	23.0
19-Oct-20 4:43:10 PM CEST	[ ]	{ok}	23.0
19-Oct-20 4:43:20 PM CEST	[ ]	{ok}	23.0

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

Publish

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

/devices/AHU\_01\_u0SAYCUHda9VUZ5h5Z7nR/command Put

```

1 {
2   "type": "SEND_HISTORY",
3   "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",
4   "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
5   "payload": {
6     "startDate": "2020-10-19T00:40:00.000+02:00",
7     "endDate": "2020-10-19T23:50:00.000+02:00"
8   }
9 }

```

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.

```

{
  "type": "SEND_HISTORY",
  "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",
  "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
  "payload": {
    "startDate": "2020-10-19T00:40:00.000+02:00",
    "endDate": "2020-10-19T23:50:00.000+02:00"
  }
}

```

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

Commands Policy	Single Point Command	
Message Type	<code>{json('type')}</code>	?
Command Set Object	<code>{json('').escape}</code>	?
Command Device Id	<code>{json('deviceId')}</code>	?
Command Point Id	<code>{json('pointId')}</code>	?
Command Action	<code>{json('payload.action')}</code>	?
Command Value	<code>{json('payload.value')}</code>	?
Command Duration	<code>{json('payload.duration')}</code>	?
Start Date	<code>{json('payload.startDate')}</code>	?
End Date	<code>{json('payload.endDate')}</code>	?
Delta	<code>{json('payload.delta')}</code>	?
Roll Up	<code>{json('payload.rollup')}</code>	?

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,
  "min": null,
  "max": null,
  "avg": 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": "null",
  "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",

```

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

History Message Variables	\$(startTimestamp) \$(endTimestamp) \$(pointId) \$(deviceId) \$(status) \$(value) \$(trendsFlags) \$(count) \$(min) \$(max) \$(avg) \$(sum)
History Message Template	<pre>{   "startTimestamp": "\${startTimestamp}",   "endTimestamp": "\${endTimestamp}",   "deviceId": "\${deviceId}",   "pointId": "\${pointId}",   "trendsFlags": "\${trendsFlags}",   "status": "\${status}",   "value": \${value},   "count": \${count},   "min": \${min}. }</pre>

## 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.
  - This is the default command template.

```
{
  "type": "ACK_ALARM",
  "payload": {
    "uuid": "5cc78b20-e516-4f82-a824-1ce9dcfb5e75"
  }
}
```

- This is the default connector send history command configuration.

	"min": \${min},
Alarm Uuid	{json('payload.uuid')}

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

Alarm History						
Timestamp	Source	State	Ack State	Source	Alarm Class	Priority
20-Oct-20 9:32:49 AM CEST	Active Power A	Normal	Unacked	Active Power A	HVAC_Level1	255
20-Oct-20 9:35:21 AM CEST	Active Power A	Normal	Unacked	Active Power A	HVAC_Level1	255
20-Oct-20 9:43:19 AM CEST	Active Power A	Normal	Unacked	Active Power A	HVAC_Level1	255
20-Oct-20 9:53:50 AM CEST	Active Power A	Normal	Unacked	Active Power A	HVAC_Level1	255
20-Oct-20 9:59:19 AM CEST	Active Power A	Normal	Unacked	Active Power A	HVAC_Level1	255
20-Oct-20 10:01:51 AM CEST	Active Power A	Normal	Unacked	Active Power A	HVAC_Level1	255

Alarm Record

Timestamp	20-Oct-20 9:32:49 AM CEST
Uuid	5cc78b20-e516-4f82-a824-1ce9dcfb5e75
Source State	Normal

- Go to device and send the command.

Publish

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









/devices/alarms/command

Publish to topic

```
1 {
2   "type": "ACK_ALARM",
3   "payload": {
4     "uuid": "5cc78b20-e516-4f82-a824-1ce9dcfb5e75"
5   }
6 }
```

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

#### Alarm History

Timestamp	Source State	Ack State	Source	Alarm Class	Priority	Message Text
 20-Oct-20 9:32:49 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power Phase A
 20-Oct-20 9:35:21 AM CEST	Normal	Unacked	Active Power A	HVAC_Level1	255	To Normal Power Phase A
 20-Oct-20 9:43:...	 Alarm Record					
 20-Oct-20 9:53:...						
 20-Oct-20 9:59:...						
 20-Oct-20 10:00:...						
 20-Oct-20 10:00:...						

**Timestamp** 20-Oct-20 9:32:49 AM CEST

**Uuid** 5cc78b20-e516-4f82-a824-1ce9dcfb5e75

**Source State** Normal

**Ack State** Acked

**Ack Required** false