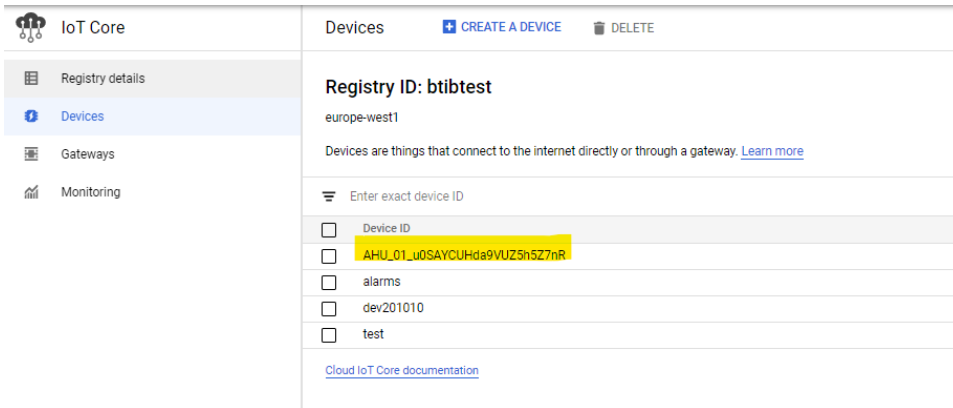


Step 5 Send commands from google cloud to Niagara

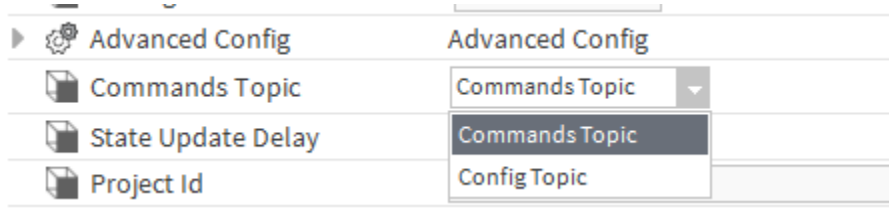
Google Connector supports cloud to niagara communication via mqtt messages to control points remotely.

Send a point action command

1. Go to the Google IoT Core view then choose the name of the device you want to send commands to. (make sure to use the commands topic not the config topic)



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



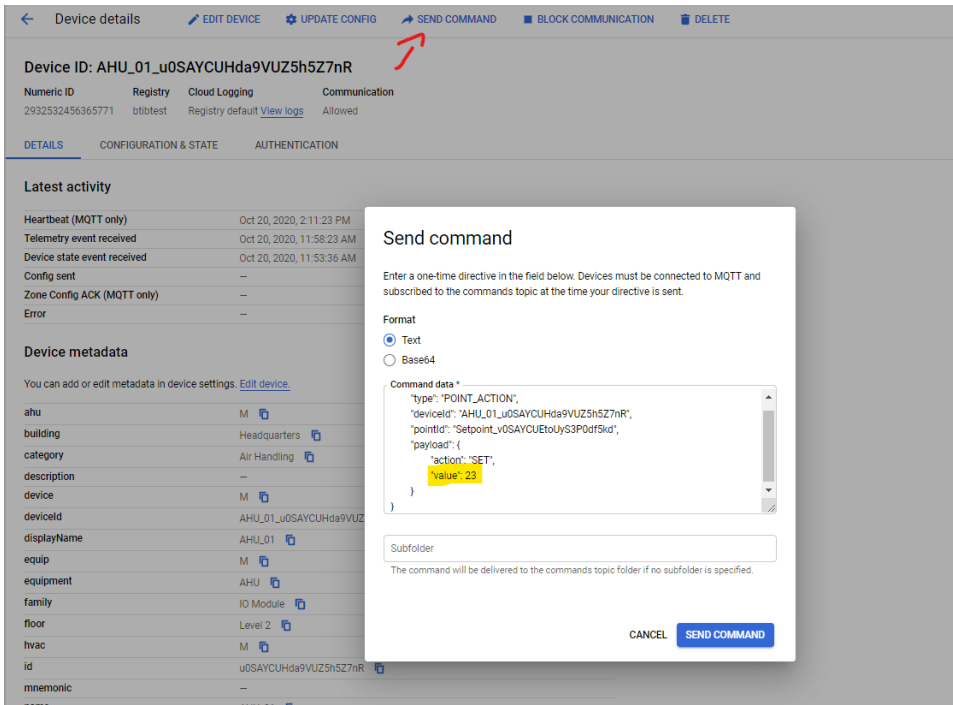
2. By default we use this message template for **POINT_ACTION** command. by you can use any format that meat your needs. check the connector advanced settings.
 - a. This is the default command template.

```
{
  "type": "POINT_ACTION",
  "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",
  "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
  "payload": {
    "action": "SET",
    "value": 23
  }
}
```

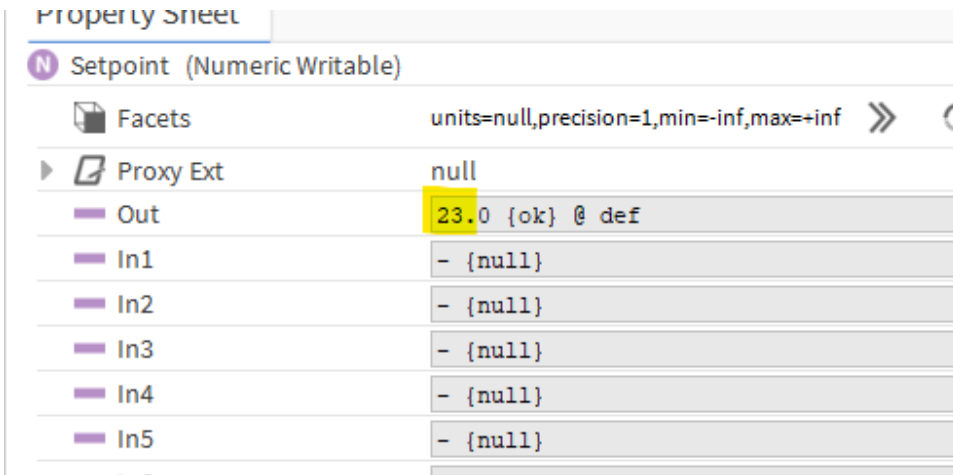
- b. This is the default connector point action command configuration.

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')}

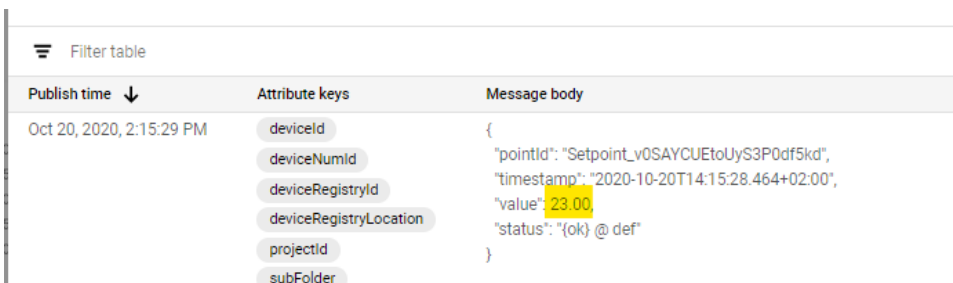
3. On the left chose send command.



- Then hit **Send command**.
- On the niagara you should see the point value applied successfully.



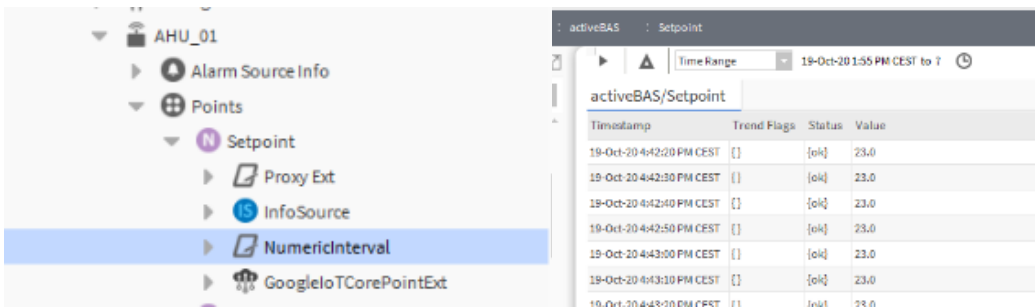
- And the new value sent to the cloud.



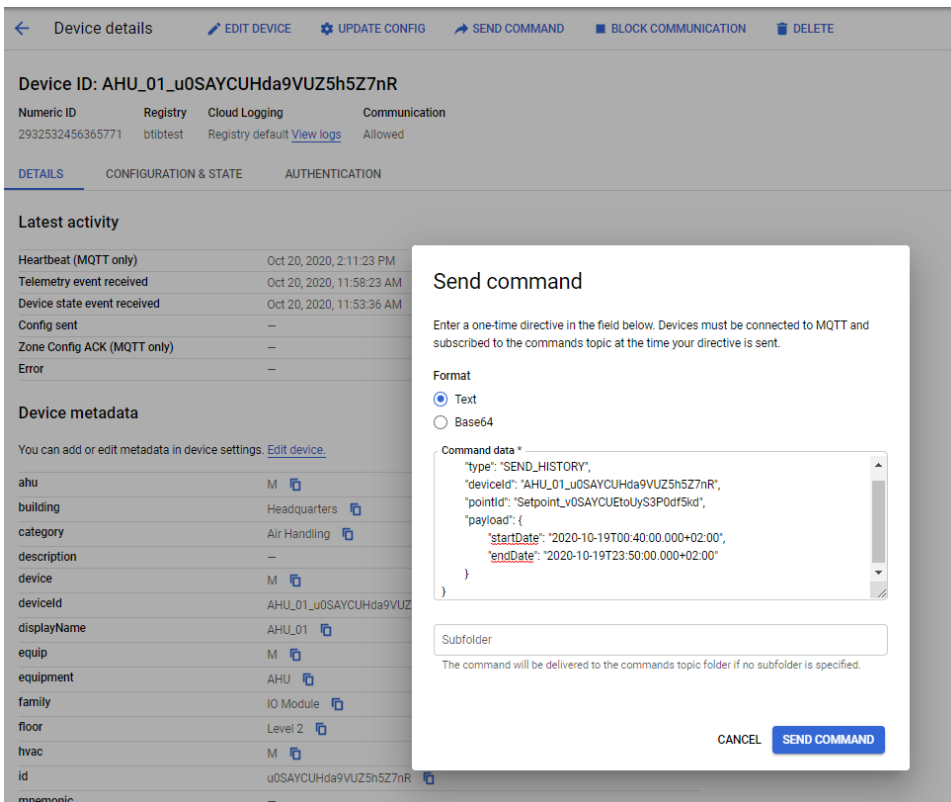
Send a send history command

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.



2. Go to the cloud console 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.

```
{
  "type": "SEND_HISTORY",
  "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR",
  "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd",
  "payload": {
    "startDate": "2020-10-20T00:40:00.000+02:00",
    "endDate": "2020-10-20T23:50:00.000+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.

PULL ☐ Enable ack messages

Filter table			
Publish time ↓	Attribute keys	Message body	Ack
Oct 20, 2020, 2:22:22 PM	<div>deviceId</div> <div>deviceNumId</div> <div>deviceRegistryId</div> <div>deviceRegistryLocation</div> <div>projectId</div> <div>subFolder</div>	<pre>{ "startTimestamp": "2020-10-20T13:01:30.479+02:00", "endTimestamp": "null", "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR", "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd", "trendsFlags": "0", "status": "(overridden)", "value": 10.0, "count": null, "min": null, "max": null, "avg": null, "sum": null }</pre>	Deadline exceeded
Oct 20, 2020, 2:22:22 PM	<div>deviceId</div> <div>deviceNumId</div> <div>deviceRegistryId</div> <div>deviceRegistryLocation</div> <div>projectId</div> <div>subFolder</div>	<pre>{ "startTimestamp": "2020-10-20T13:01:40.325+02:00", "endTimestamp": "null", "deviceId": "AHU_01_u0SAYCUHda9VUZ5h5Z7nR", "pointId": "Setpoint_v0SAYCUEtoUyS3P0df5kd", "trendsFlags": "0", "status": "(overridden)", "value": 10.0, "count": null, "min": null, "max": null, "avg": null, "sum": null }</pre>	Deadline exceeded
Oct 20, 2020, 2:22:22 PM	<div>deviceId</div>	<pre>{</pre>	Deadline exceeded

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": "f698beca-c625-44ce-a8d8-5af9b9f46b1b"
  }
}
```

- 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.

Today

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 PI
20-Oct-20 9:35:21 AM CEST	Normal	Unacked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 9:43:19 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 9:53:50 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 9:59:19 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 10:01:51 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 10:07:19 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 10:12:49 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI
20-Oct-20 10:15:21 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power PI

Alarm Record

Timestamp	20-Oct-20 9:35:21 AM CEST
Uuid	f698beca-c625-44ce-a8d8-5af9b9f46b1b
Source State	Normal
Ack State	Unacked
Ack Required	true
Source	Active Power A

- Go to the alarm device and send the command.

Device details EDIT DEVICE UPDATE CONFIG SEND COMMAND BLOCK COMMUNICATION DELETE

Device ID: **alarms**

Numeric ID 2857530869906684 Registry btbtest Cloud Logging Registry default View logs Communication Allowed

DETAILS CONFIGURATION & STATE AUTHENTICATION

Latest activity

Heartbeat (MQTT only)	Oct 20, 2020, 2:26:08 PM
Telemetry event received	Oct 20, 2020, 12:09:07 PM
Device state event received	—
Config sent	—
Zone Config ACK (MQTT only)	—
Error	—

Device metadata

You can add or edit metadata in device settings. [Edit device.](#)

Send command

Enter a one-time directive in the field below. Devices must be connected to MQTT and subscribed to the commands topic at the time your directive is sent.

Format

☒ Text ☐ Base64

Command data *

```
{
  "type": "ACK_ALARM",
  "payload": {
    "uuid": "f698beca-c625-44ce-a8d8-5af9b9f46b1b"
  }
}
```

Subfolder

The command will be delivered to the commands topic folder if no subfolder is specified.

CANCEL SEND COMMAND

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

Today

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
20-Oct-20 9:35:21 AM CEST	Normal	Acked	Active Power A	HVAC_Level1	255	To Normal Power
20-Oct-20 9:43:19						
20-Oct-20 9:53:50						
20-Oct-20 9:59:19						
20-Oct-20 10:01:5						
20-Oct-20 10:07:1						
20-Oct-20 10:12:4						

Alarm Record

Timestamp	20-Oct-20 9:35:21 AM CEST
Uuid	f698beca-c625-44ce-a8d8-5af9b9f46b1b
Source State	Normal
Ack State	Acked
Ack Required	false
Source	Active Power A

Next Step

Step 6 Consume Data from IoT Core to Google BigQuery