Edge Xpert V2 Demo:
Chemical Tank Control

Demo Overview

This end-to-end Edge Xpert demonstration performs the following key edge IoT tasks:

- Collecting and ingesting data from two simulated data sources - in this case Modbus and OPC-UA
- Delivering the data to an InfluxDB time-series database
- Visualization of the data via a user-configured Grafana dashboard
- Edge decision making, control and actuation via Node-RED
- Streaming data northbound to a Cloud/IT system - in this case represented by HiveMQ

The demonstration models a Modbus-based chemical tank producing temperature, pressure and level values, and an OPC-UA outlet valve with a setting to open and close the valve. Many other device connectors are supported but not demonstrated here. The control logic applied by Node-RED in this demo will open the OPC-UA outlet valve when any of the Modbus chemical tank values rise beyond a given level. When the levels drop the valve is closed again.

Demo Requirements

- Linux (this version tested on Ubuntu 20.04) with sudo permissions
- Edge Xpert requires Docker and Docker-Compose to be installed
- The Modbus and OPC-UA simulators require Java to be installed
    - The Modbus simulator also requires librxtx-java (e.g. sudo apt-get install librxtx-java)

Demo Preparation

1. **Install Edge Xpert**

   Edge Xpert V2 is still in the GA process and so is not yet available from the IOTech website.
   Please contact IOTech for the Edge Xpert V2 download link.

   Note that the installation instructions for Edge Xpert V1 still apply to V2.

   Ensure that Docker and Docker-Compose are installed and working.

2. **Install the Edge Xpert license file**

   Please contact IOTech if you don't have an Edge Xpert license file.

   Ensure the license file is readable by the "other" user:
   $> chmod 774 <EdgeXpertLicenseFile>.lic

   $> edgexpert license install <EdgeXpertLicenseFile>.lic
   $> edgexpert license check
   Edge Xpert Licensing: Signature valid
   Edge Xpert Licensing: License valid

3. **Install the Prosys OPC-UA Simulation Server**

   Download link: https://downloads.prosysopc.com/opc-ua-simulation-server-downloads.php
   install command similar to:
   $> sh prosys-opc-ua-simulation-server-linux-5.0.4-284.sh

Demo Operation Mode

Edge Xpert can be controlled and operated via both its REST APIs and via the Edge Xpert Manager UI.

This demonstration provides full instructions for interacting with the Edge Xpert CLI and Edge Xpert Management UI tools. This is recommended for a first time user since it helps in understanding of the specific steps involved and how the Edge Xpert microservices operate.

In addition, a **curl-commants.txt** file shows the REST commands that can achieve each step, while a complete **setup.sh** shell script can execute all of the steps below.

Running the Demo

1. Start the Edge Xpert using the Edge Xpert CLI tool

If any previous instances of Edge Xpert have been running, stop and delete these:
$> edgexpert clean

Start the required microservices and then verify they are running:
$> edgexpert up --secret device-modbus device-opc-ua xpert-manager mqtt-broker influxdb grafana nodered sys-mgmt

Note that the "--secret" and "sys-mgmt" options are required for the Edge Xpert Manager UI to create the application services in steps 6-9 below.

```
$> edgexpert up --secret device-modbus device-opc-ua xpert-manager mqtt-broker influxdb grafana nodered sys-mgmt
Creating network "edgexpert_edgex-network" with driver "bridge"
Creating network "edgexpert_default" with the default driver
Creating volume "edgexpert_db-data" with default driver
Creating volume "edgexpert_consul-data" with default driver
Creating volume "edgexpert_consul-config" with default driver
Creating volume "edgexpert_kuiper-data" with default driver
Creating volume "edgexpert_edgex-init" with default driver
Creating volume "edgexpert_consul-acl-token" with default driver
Creating volume "edgexpert_redis-config" with default driver
Creating volume "edgexpert_vault-config" with default driver
Creating volume "edgexpert_vault-file" with default driver
Creating volume "edgexpert_vault-logs" with default driver
Creating volume "edgexpert_postgres-config" with default driver
Creating volume "edgexpert_postgres-data" with default driver
Creating volume "edgexpert_kong" with default driver
Creating volume "edgexpert_kuiper-config" with default driver
Creating volume "edgexpert_asc-config" with default driver
Creating volume "edgexpert_grafana-data" with default driver
Creating volume "edgexpert_nodered-data" with default driver
Creating volume "edgexpert_portainer-data" with default driver
Creating volume "edgexpert_mosquitto-log" with default driver
Creating volume "edgexpert_mosquitto-data" with default driver
Creating volume "edgexpert_xpert-manager-data" with default driver
Creating volume "edgexpert_device-modbus-data" with default driver
Creating volume "edgexpert_device-bacnet-data" with default driver
Creating volume "edgexpert_device-opc-ua-data" with default driver
Creating consul                ... done
Creating influxdb              ... done
Creating redis                 ... done
Creating mqtt-broker           ... done
Creating secretstore-setup     ... done
Creating grafana               ... done
Creating security-bootstrapper ... done
Creating xpert-manager         ... done
Creating core-data             ... done
Creating nodered               ... done
Creating device-opc-ua         ... done
Creating vault                 ... done
Creating sys-mgmt              ... done
Creating core-metadata         ... done
Creating core-command          ... done
Creating device-modbus         ... done
$>
```

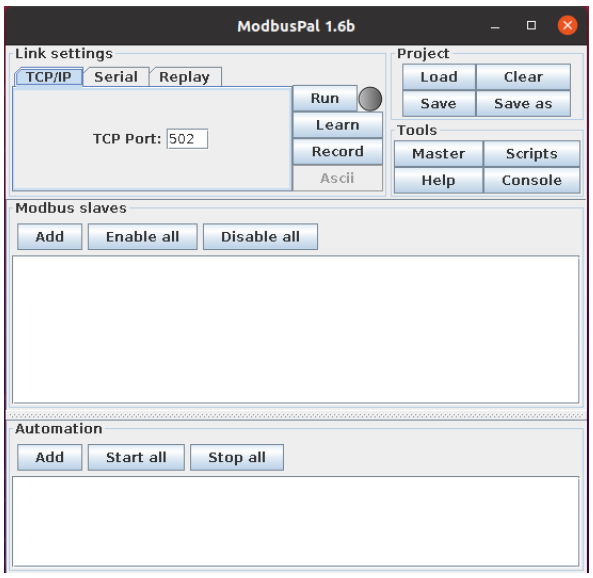
$> edgexpert status

```
$> edgexpert status
NAMES                   STATUS                          CREATED             PORTS
core-metadata           Up About a minute               About a minute ago  0.0.0.0:59881->59881/tcp
device-modbus           Up About a minute               About a minute ago  0.0.0.0:59901->59901/tcp
core-command            Up About a minute               About a minute ago  59882/tcp
vault                   Up About a minute               About a minute ago  8200/tcp
device-opc-ua           Up About a minute               About a minute ago  0.0.0.0:59953->59953/tcp
nodered                 Up About a minute (healthy)     About a minute ago  0.0.0.0:1880->1880/tcp
sys-mgmt                Up About a minute               About a minute ago  0.0.0.0:58890->58890/tcp
core-data               Up About a minute               About a minute ago  0.0.0.0:5563->5563/tcp, 0.0.0.0:
xpert-manager           Up About a minute               About a minute ago  0.0.0.0:9090->9090/tcp
grafana                 Up About a minute               About a minute ago  0.0.0.0:3000->3000/tcp
security-bootstrapper   Up About a minute               About a minute ago
mqtt-broker             Up About a minute               About a minute ago  0.0.0.0:1883->1883/tcp
secretstore-setup       Up About a minute               About a minute ago
redis                   Up About a minute               About a minute ago  6379/tcp
consul                  Up About a minute               About a minute ago  8300-8302/tcp, 8301-8302/udp, 86
influxdb                Up About a minute               About a minute ago  0.0.0.0:8086->8086/tcp
$>
```

2. Start the ModbusPal simulator

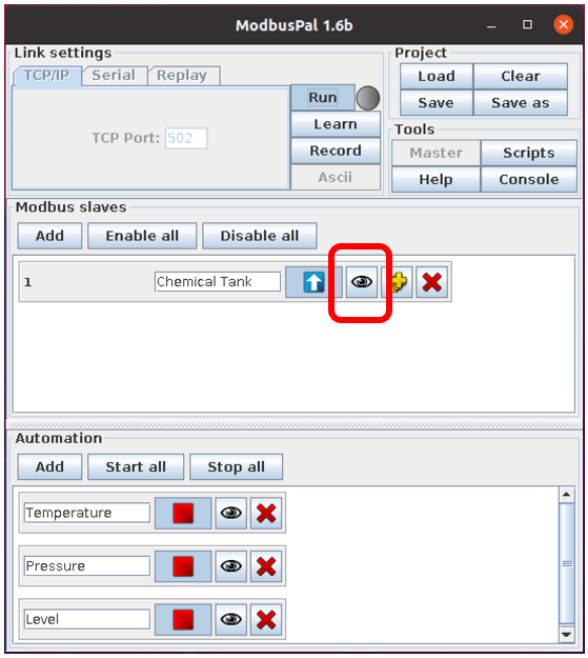
Note it is important to use the ModbusPal.jar provided in the project. Other versions of ModbusPal have been found to not work correctly with loaded simulation files.

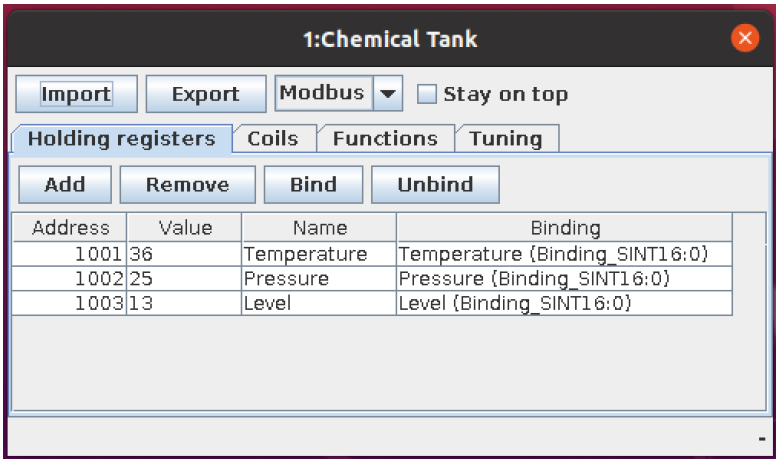
$> sudo java -jar ModbusPal.jar



Project -> Load -> ModbusSimulation.xmpp file
Automation -> Start all
Link settings -> Run

Clicking the “eye” icon shows the changing values in the simulator:





3. Start the OPC-UA Simulation Server

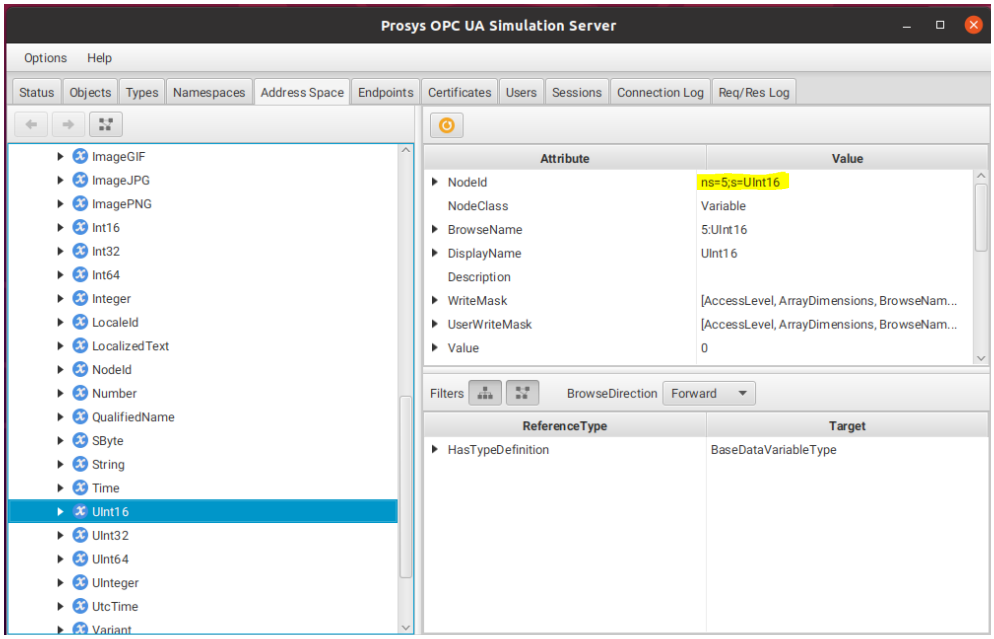
The actual command depends on where it was installed
$> ./prosys-opc-ua-simulation-server/UaSimulationServer

Switch to expert mode (Options -> Switch to Expert Mode)

Note that IPV6 may need to be temporarily disabled to help the Prosys simulator start (Ubuntu issue)

Verify that the following attribute exists (it is used in the OutletValve.yaml). If the NodeID is different, please edit the nodeID and nsIndex in OutletValve.yaml to match the settings in Prosys.

Address Space -> Objects -> Static Data -> Static Variables -> UInt16 (ns=5, s=UInt16)

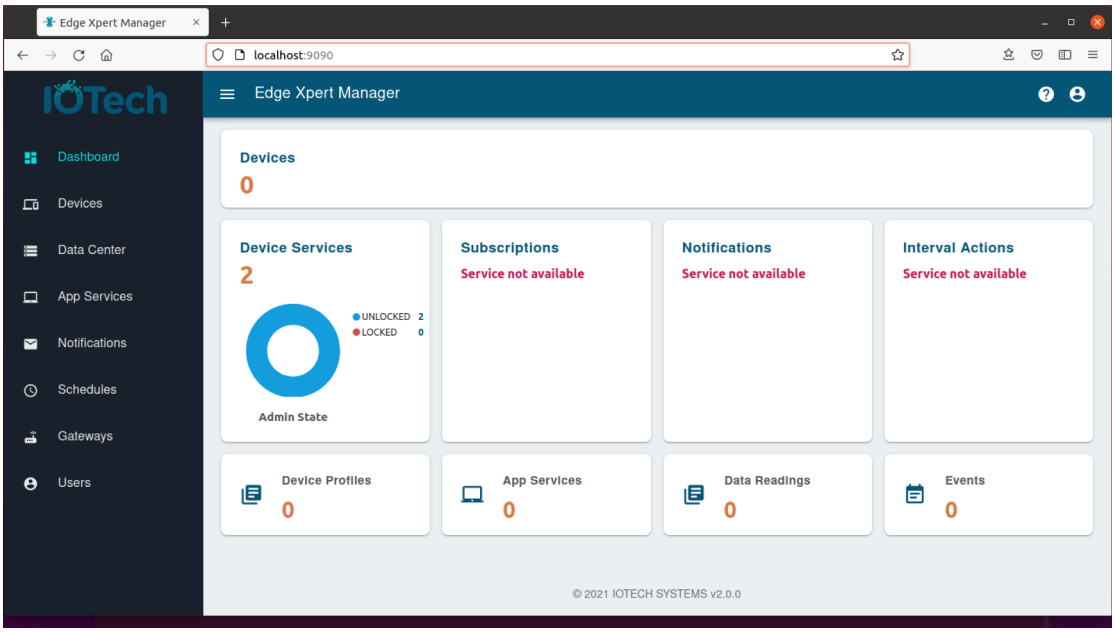


4. Onboard the Modbus device with the Edge Xpert Manager UI

Note also the Modbus device can be onboarded to Edge Xpert via curl REST commands. Refer to curl-commands.txt for details.

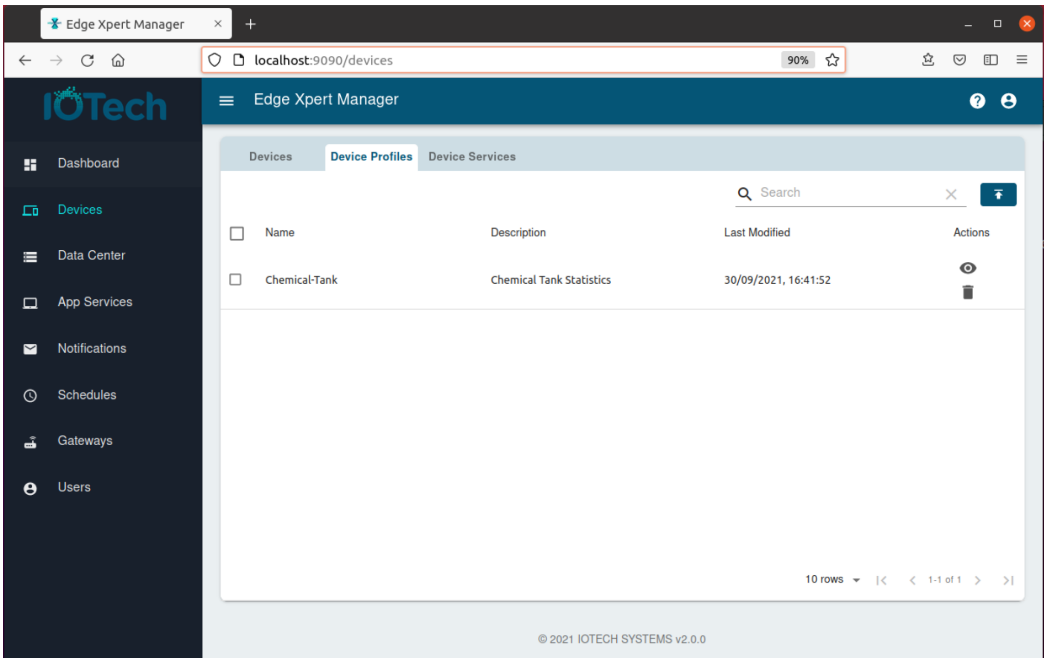
Use the Edge Xpert Manager UI. In a browser navigate to http://localhost:9090

Initial login is admin/admin



Use the device onboarding wizard to upload the ChemicalTank profile and add the Modbus device:

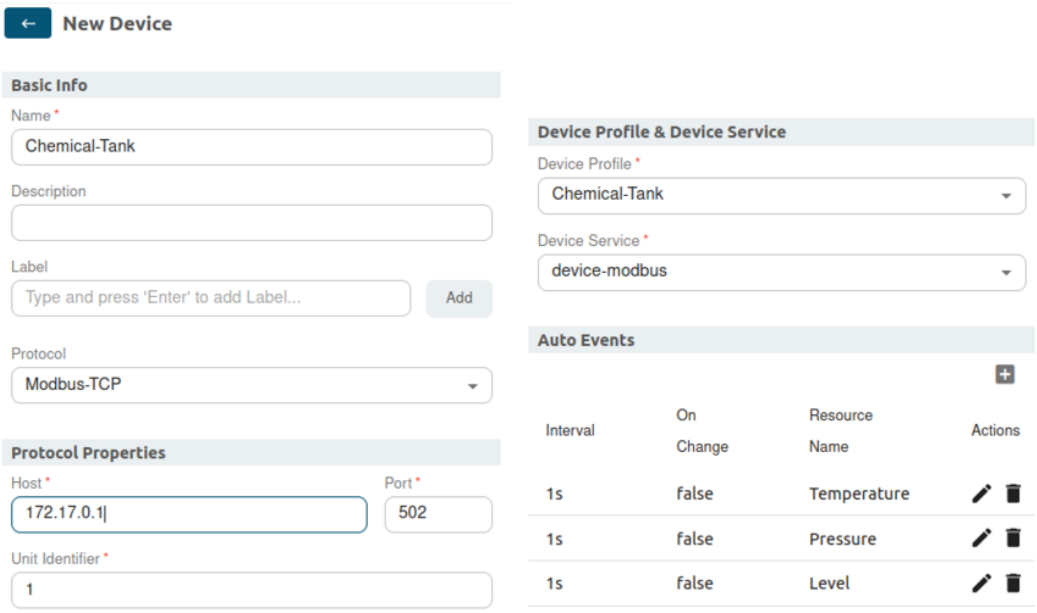
Devices -> Device Profiles -> Upload Profile -> ChemicalTank.yaml

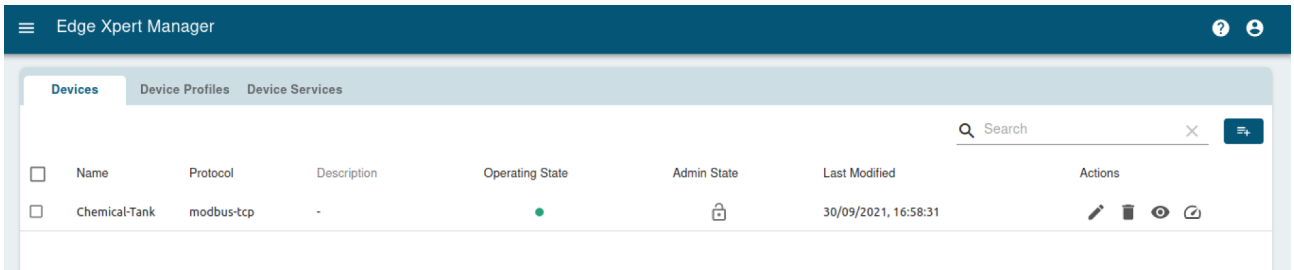


Devices -> Devices -> Add Device ("+" icon)
Name: Chemical-Tank
Protocol: Modbus-TCP
Host: 172.17.0.1
Port: 502
Unit Identifier: 1
Device Profile: Chemical-Tank.yaml
Device Service: device-modbus
AutoEvents

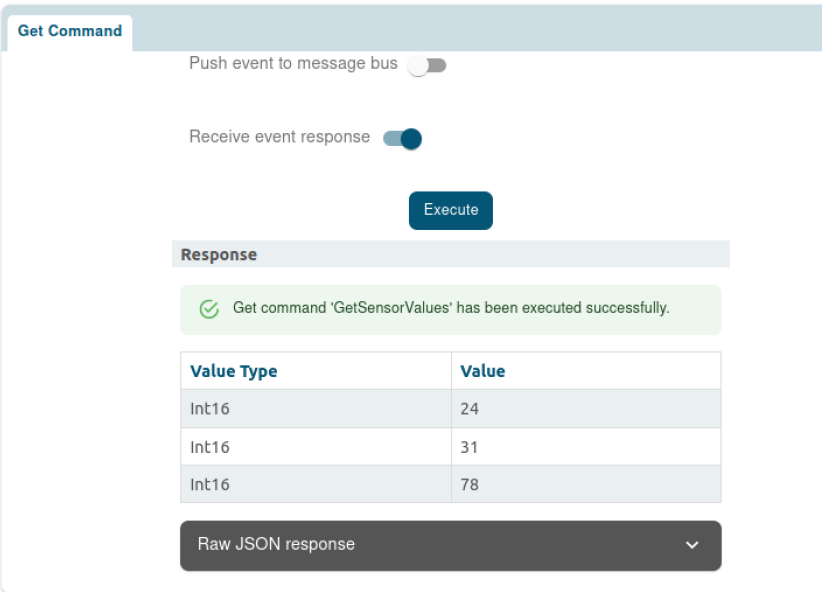
Interval: 1s, onChange: unchecked/false, Resource Name: Temperature
Interval: 1s, onChange: unchecked/false, Resource Name: Pressure
Interval: 1s, onChange: unchecked/false, Resource Name: Level

All other settings can be left as the default settings.
Click “Save” and the device should be onboarded.





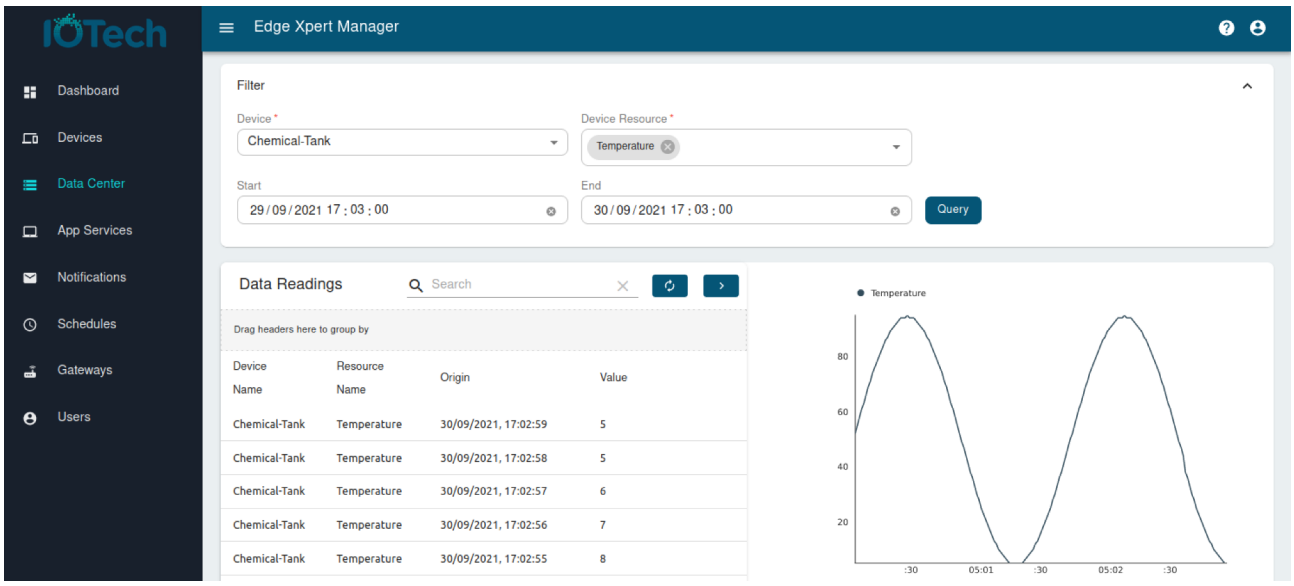
Click the “Control” button to see the current values.
Devices -> Chemical-Tank -> Control Device -> GetSensorValues -> Get Command -> execute



These values returned should track the values shown on the Modbus simulator.

Alternatively, the Edge Xpert Manager also provides a Data Center feature that lets you view the data in table and chart form:

Data Center -> Device -> Device Resource -> Query



5. Onboard the OPC-UA device with the Edge Xpert Manager UI

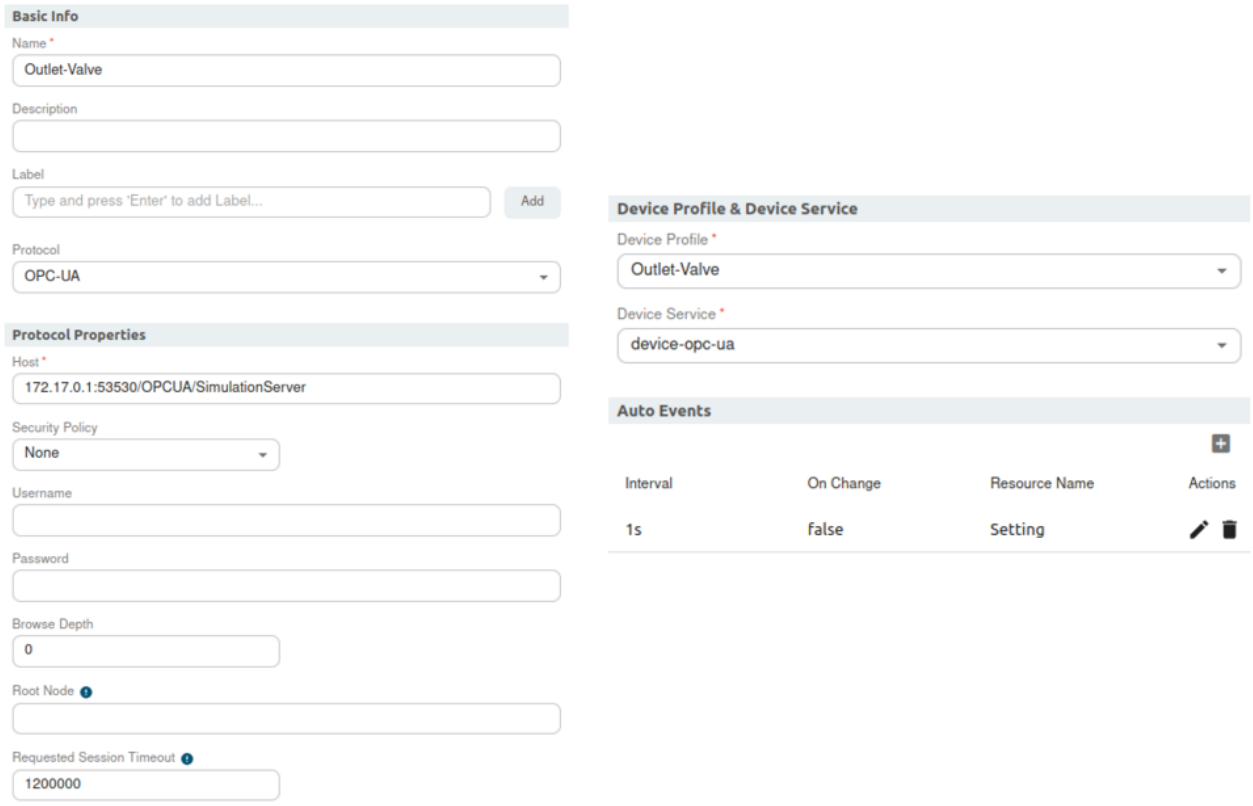
Note also the OPC-UA device can be onboarded to Edge Xpert via curl REST commands. Refer to curl-commands.txt for details.

Follow the same process to onboard the OPC-UA device, again using the Edge Xpert Manager device onboarding wizard:

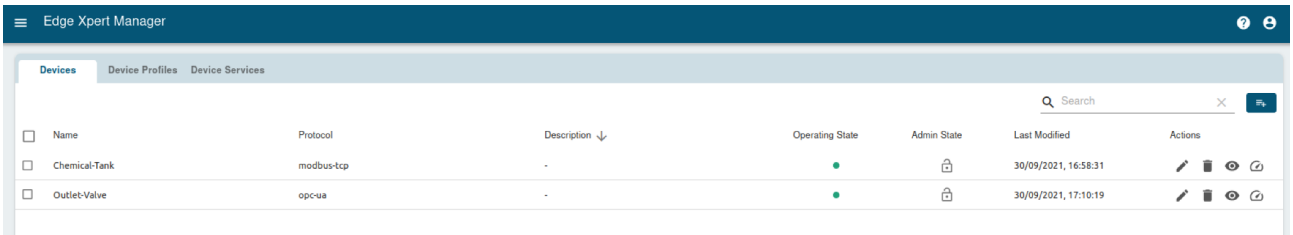
Devices -> Device Profiles -> Upload Profile -> OutletValve.yaml

Devices -> Devices -> Add Device ("+" icon)
Name: Outlet-Valve
Protocol: OPC-UA
Host: 172.17.0.1:53530/OPCUA/SimulationServer
Device Profile: Outlet-Valve.yaml
Device Service: device-opc-ua
AutoEvents
Interval: 1s, onChange: unchecked/false, Resource Name: Setting

All other settings can be left as the default settings



Click “Save” and now both devices should be connected:



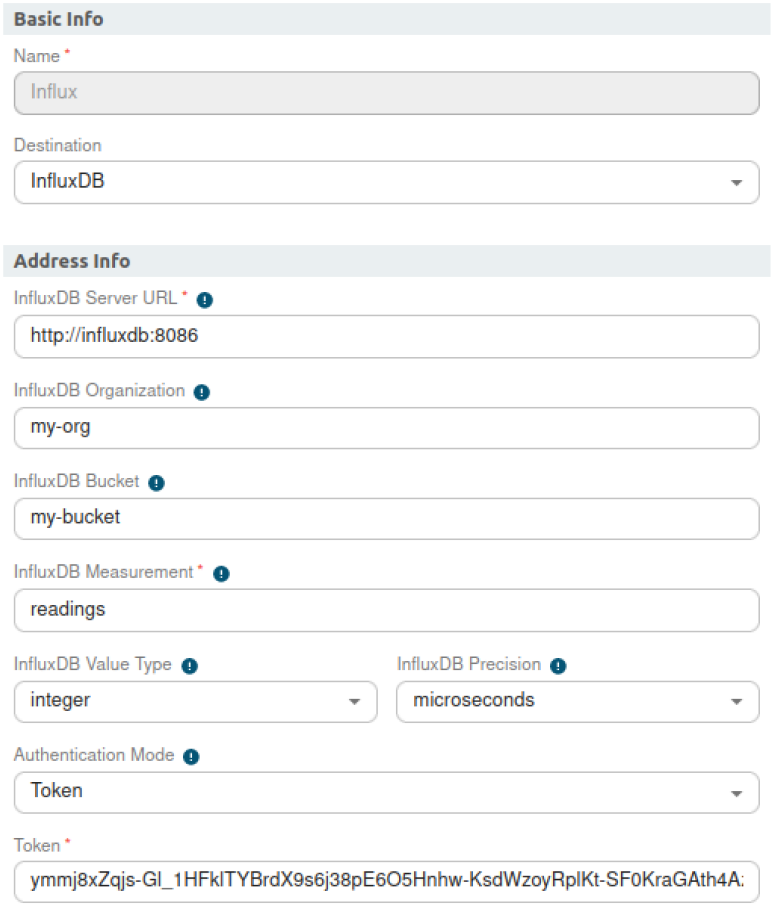
6. Export to the InfluxDB Time-Series database with the Edge Xpert Manager UI

Note also the Influx export can be created via an Edge Xpert app service on the command line. Refer to notes.txt and setup.sh for details.

```
# Get the InfluxDB authentication token. Note 'jq' must be installed, e.g. sudo apt install jq
$> docker exec influxdb influx auth list --user admin --json | jq .[] | jq '.token'
   "ymmj8xZqjs-Gl_1HFklTYBrdX9s6j38pE6O5Hnhw-KsdWzoyRplKt-SF0KraGAth4Azpg9fS_y3q2DiB2NTKhA=="
```

Use the application service onboarding wizard to create the export to InfluxDB:

```
# App Services -> Add App Service ("+" icon)
# Name: Influx
# Destination: InfluxDB
# InfluxDBServerURL: http://influxdb:8086
# InfluxDBOrganization: my-org
# InfluxDBBucket: my-bucket
# InfluxDBValueType: integer
# Authentication Mode: Token
# Token: <paste influx token obtained from above, no quotes>
All other values left as default, click Save
```



Optional: You can see if the data is arriving in influx by executing into the influxdb container:

$> docker exec -it influxdb /bin/sh
influx query 'from(bucket: "my-bucket") |> range(start: -3s) |> filter(fn: (r) => r["_measurement"] == "readings" and r["resourceName"] == "Temperature")'

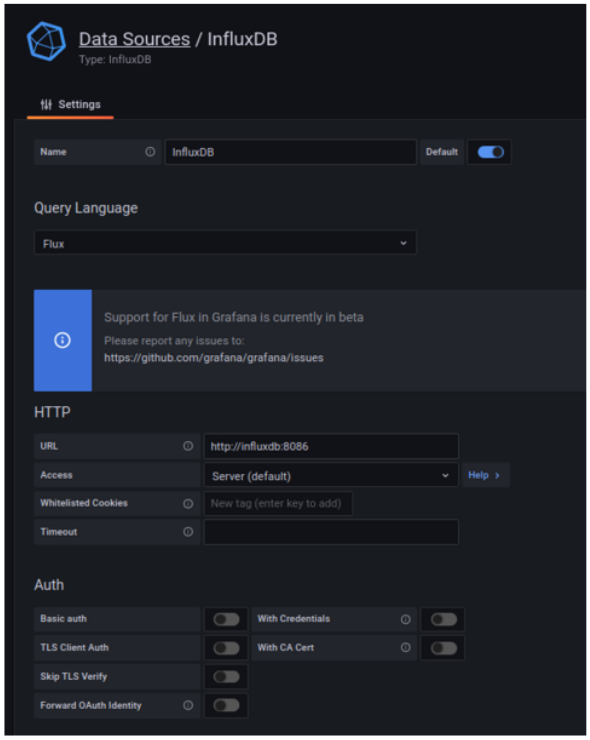
```
$> docker exec -it influxdb /bin/sh
# influx query 'from(bucket: "my-bucket") |> range(start: -3s) |> filter(fn: (r) => r["_measurement"] == "readings" and r["resourceName
"] == "Temperature")'
Result: _result
Table: keys: [_start, _stop, _field, _measurement, deviceName, event_id, resourceName]
                    _start:time                      _stop:time           _field:string     _measurement:string       deviceName:string
                       event_id:string     resourceName:string                  _value:int                      _time:time
------------------------------  ------------------------------  ----------------------  ----------------------  ----------------------
 -----------------------------------  ----------------------  -------------------------  -----------------------------
2021-09-30T16:19:09.016177163Z  2021-09-30T16:19:12.016177163Z                    value                readings           Chemical-Tank
 3c035414-985c-45bf-9fbc-5e2219610b7d             Temperature                       74  2021-09-30T16:19:09.922733000Z
Table: keys: [_start, _stop, _field, _measurement, deviceName, event_id, resourceName]
                    _start:time                      _stop:time           _field:string     _measurement:string       deviceName:string
                       event_id:string     resourceName:string                  _value:int                      _time:time
------------------------------  ------------------------------  ----------------------  ----------------------  ----------------------
 -----------------------------------  ----------------------  -------------------------  -----------------------------
2021-09-30T16:19:09.016177163Z  2021-09-30T16:19:12.016177163Z                    value                readings           Chemical-Tank
 745f8914-89c1-4dd2-b196-493973a38fa7             Temperature                       69  2021-09-30T16:19:11.926949000Z
Table: keys: [_start, _stop, _field, _measurement, deviceName, event_id, resourceName]
                    _start:time                      _stop:time           _field:string     _measurement:string       deviceName:string
                       event_id:string     resourceName:string                  _value:int                      _time:time
------------------------------  ------------------------------  ----------------------  ----------------------  ----------------------
 -----------------------------------  ----------------------  -------------------------  -----------------------------
2021-09-30T16:19:09.016177163Z  2021-09-30T16:19:12.016177163Z                    value                readings           Chemical-Tank
 9cda9678-1f88-4c4e-8e2f-72c5bc7a99de             Temperature                       71  2021-09-30T16:19:10.924787000Z
#
```

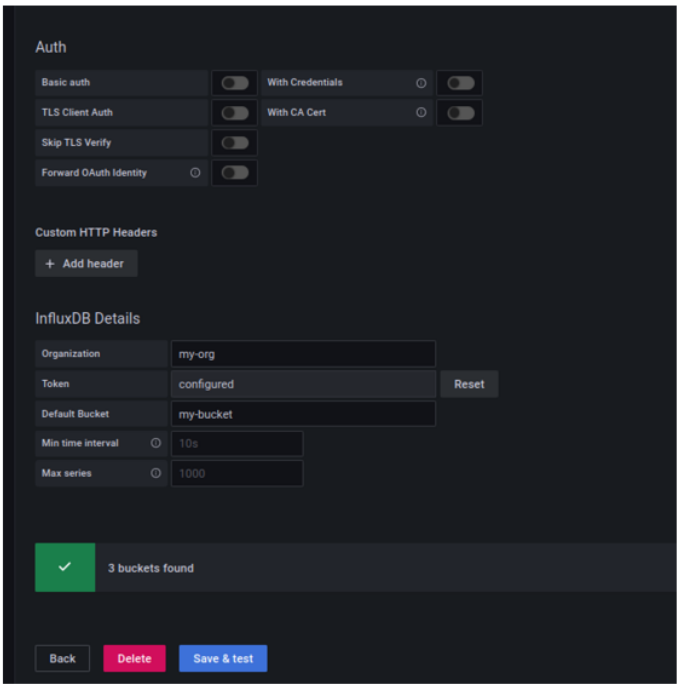
7. The Grafana dashboard

Note also the Grafana configuration and dashboard can be uploaded with a curl REST command. Refer to curl-commands.txt and setup.sh for details.

Use a browser and navigate to localhost:3000. Initial login is admin/admin

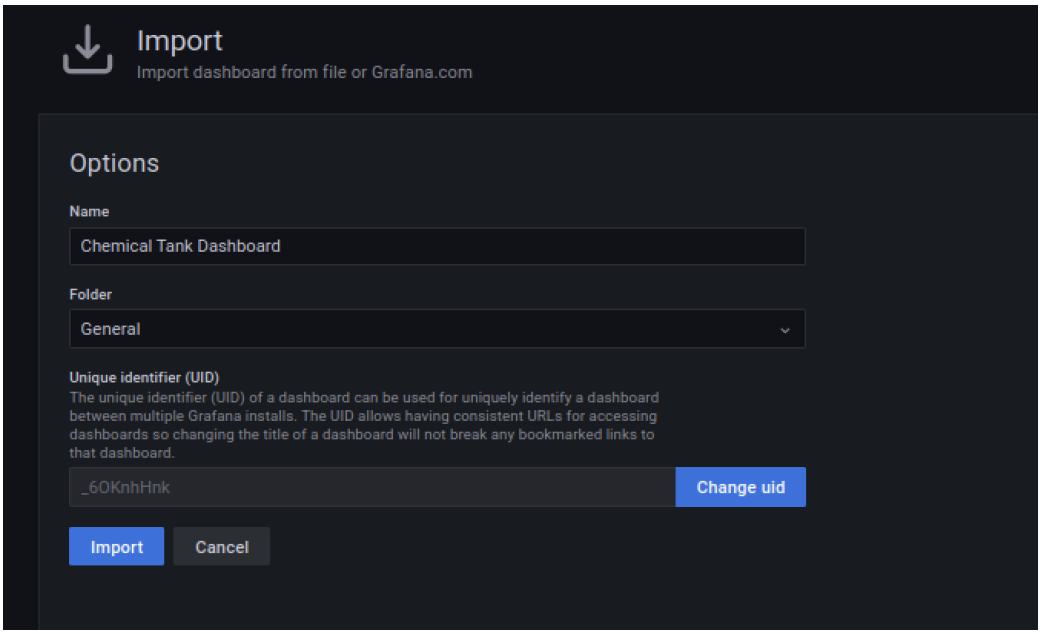
Configuration -> Data Sources -> Add: InfluxDB
Name: InfluxDB (default)
Query Language: Flux
HTTP URL: http://influxdb:8086
Disable Basic auth
InfluxDB Details:
  Organization: my-org
  Token: <paste influx token obtained from above, no quotes>, click enter
  Default Bucket: my-bucket
Click “Save & Test”. It should say "buckets found"





Now view the Grafana dashboard as follows:

Go to Create ("+" icon) -> Import -> Upload .json file -> select Grafana.json -> Import



The dashboard should look similar to as follows:

8. Edge rules with Node-RED

Note also the Node-RED flow can be uploaded and started with a curl REST command. Refer to curl-commands.txt and setup.sh for details.

Use a browser and navigate to localhost:1880
Upload the flows rule:
Options -> Import -> Clipboard -> Select a file to import -> flows.json
Once loaded, can deploy the flows: Deploy
Should show successfully deployed

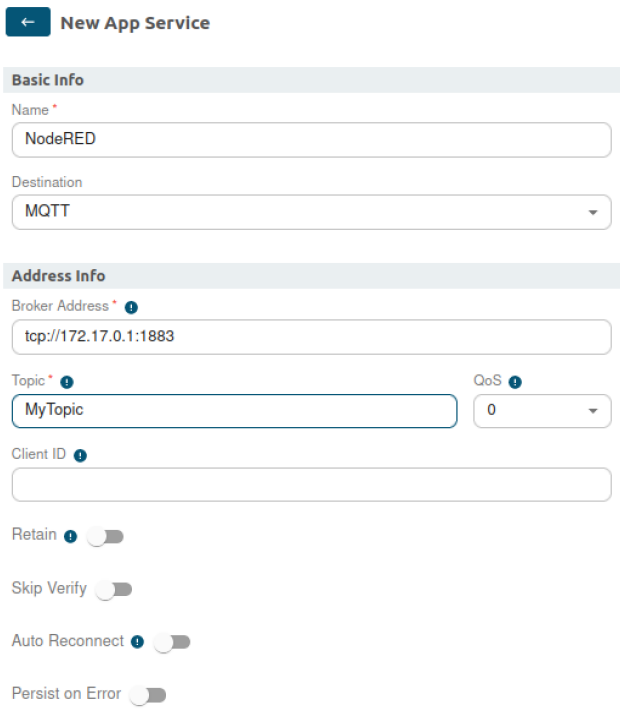
The Node-RED flow uses MQTT to receive the data from Edge Xpert. The “Decision Making” function executes some simple JavaScript code to perform the control logic. Export the data to the built in MQTT-broker so that Node-RED can receive the data and the logic flow starts.

Note also the MQTT export can be created via an Edge Xpert app service on the command line. Refer to notes.txt and setup.sh for details.

Use the application service onboarding wizard to create the export to MQTT:

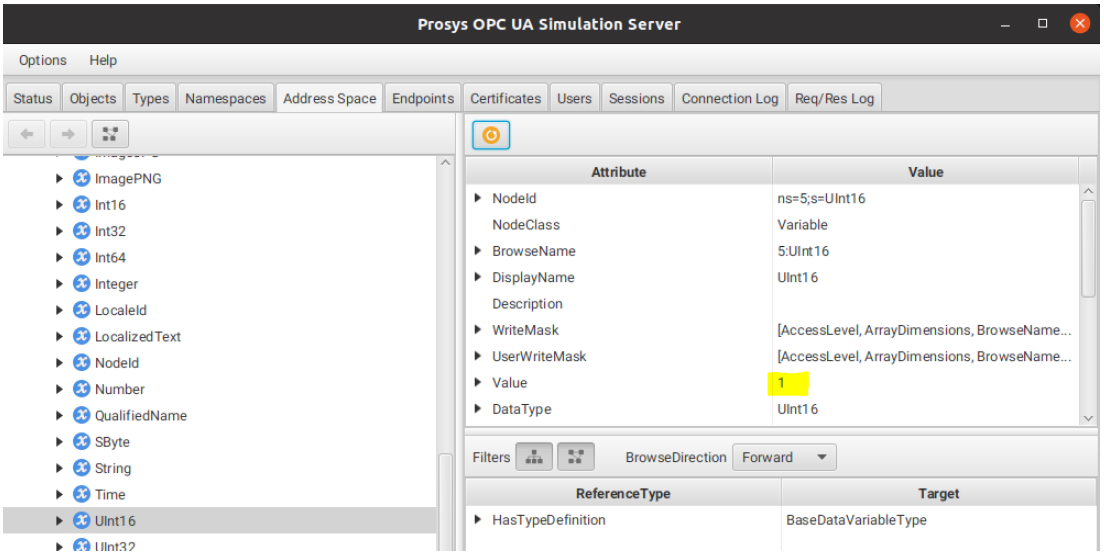
App Services -> Add App Service ("+" icon)
Name: NodeRED
Broker Address: tcp://172.17.0.1:1883
Topic: MyTopic

All other values left as default, click Save



Back on the Grafana dashboard, you should now see that the Outlet Valve is opened (i.e. set to 1) when any of the Modbus values go above 80 (as per the rule in the Node-RED flows.json):

Checking on the OPC-UA server will show that the Outlet-Valve Setting parameter has been opened (i.e. set to 1) when any of the Modbus values are greater than 80. You may need to refresh the view in order to see data updates.



9. Export to the Cloud with the Edge Xpert Manager UI

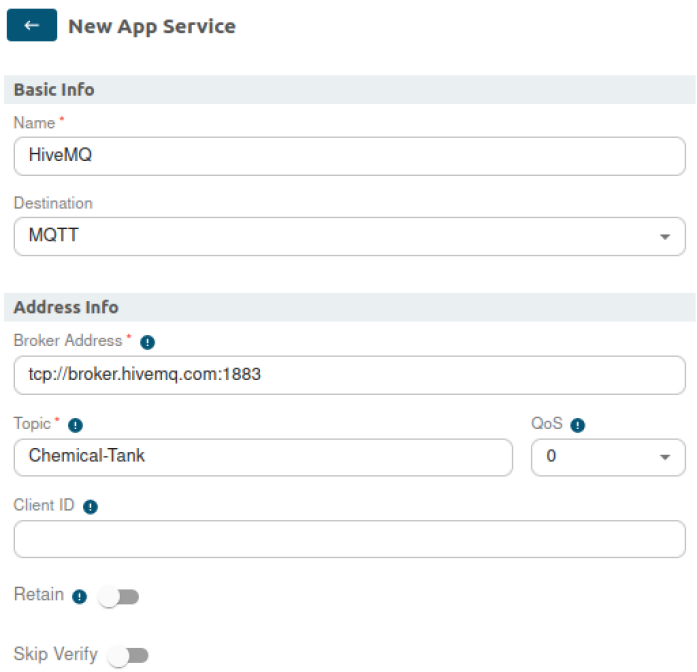
Note also that Cloud exports can be created via an Edge Xpert app service on the command line. Refer to notes.txt and setup.sh for details.

Use the Edge Xpert Application Services to stream the data to Cloud endpoint, in this case HiveMQ which is easy and convenient to use.

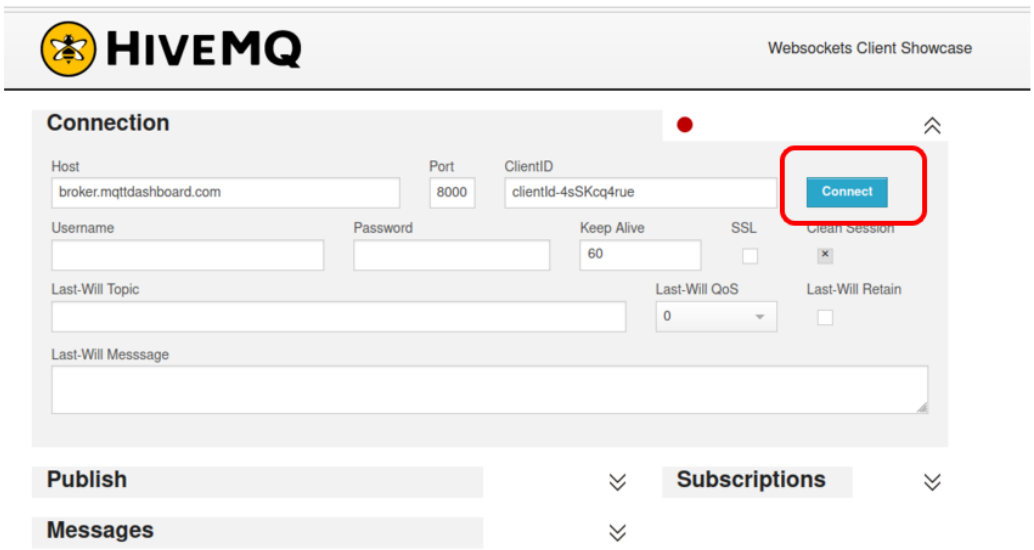
Use the application service onboarding wizard to create the export to MQTT:

App Services -> Add App Service ("+" icon)
Name: HiveMQ
Broker Address: tcp://broker.hivemq.com:1883
Topic: Chemical-Tank

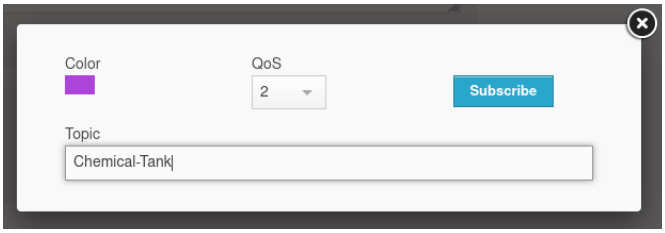
All other values left as default, click Save



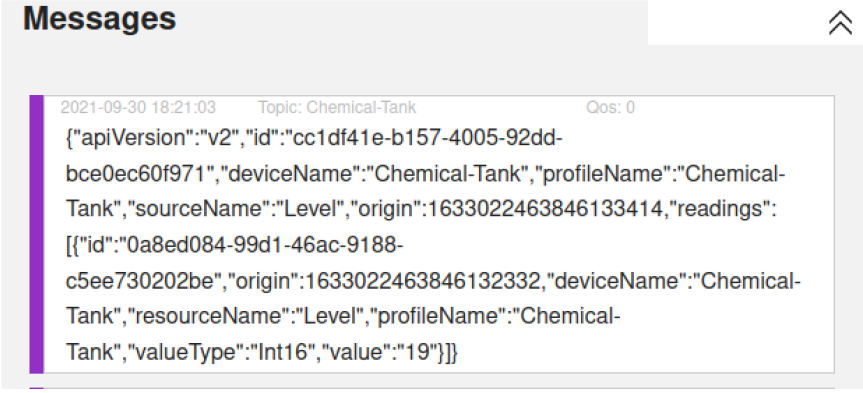
Open a web browser, navigate to http://www.hivemq.com/demos/websocket-client/
Click “Connect” with the default credentials



Once connected, click “Add New Topic Subscription” and set Topic to Chemical-Tank



You should see the data arriving in the cloud:



10. Stop Edge Xpert

The stop command stops all the running Edge Xpert microservices but leaves the containers (and therefore associated state) for later restart.

$> edgexpert down

This means that the Edge Xpert microservices and saved state such as the configured devices, exports and dashboard configurations will be maintained, so that an edge xpert up command will cause the microservices to pick up where they were stopped. No reconfiguration is required.

Warning: The edgexpert clean command stops all the running Edge Xpert microservices, deletes all the containers and deletes any volumes or networks used by Edge Xpert. So all Edge Xpert state is lost when this operation is conducted. The next time you run an 'edgexpert up', all of the configuration defined above will need to be recreated

$> edgexpert clean