

- [Распределенные системы и сети](#)
 - [Урок 3. Примеры распределённых технологий](#)
 - [Домашнее задание](#)
 - [Решение](#)
 - [Учетные данные](#)
 - [Список сервисов и их адресов](#)
 - [1. Mosquitto](#)
 - [2. IndluxDB2](#)
 - [3. Grafana](#)
 - [4. NodeRed](#)
 - [Barviha](#)
 - [Nginx](#)

Распределенные системы и сети

Урок 3. Примеры распределённых технологий

Домашнее задание

Развертывание системы проксирования через NGINX, согласно занятию.

Предоставление реквизитов доступа в виде доменных имен к сервисам Grafana, InfluxDB, Node-Red каждым студентом.

(Например: grafana-gvi.gb-iot-1234.ru, influxdb-gvi.gb-iot-1234.ru, nodered-gvi.gb-iot-1234.ru).

Решение

Учетные данные

Ко всем службам используются одни учетные данные:

- логин: admin
- пароль: students

Список сервисов и их адресов

1. **Mosquitto**: wss://mosquitto-rrg-5471.gb-iot.ru
2. **InfluxDB2**: https://influxdb-rrg-5471.gb-iot.ru
3. **Grafana**: https://grafana-rrg-5471.gb-iot.ru
4. **NodeRed**: https://nodered-rrg-5471.gb-iot.ru

1. Mosquitto

В качестве клиента рекомендую [MQTTX](#), т.к. стандартный консольный клиент не умеет работать с вебсокетом, который еще и обернут в TLS от nginx.

В схеме обратного прокси с поддоменами, достаточно одного открытого порта на шлюзе 443 для nginx, в отличие от схемы с подпапками, где пришлось открывать дополнительный порт под вебсокет.

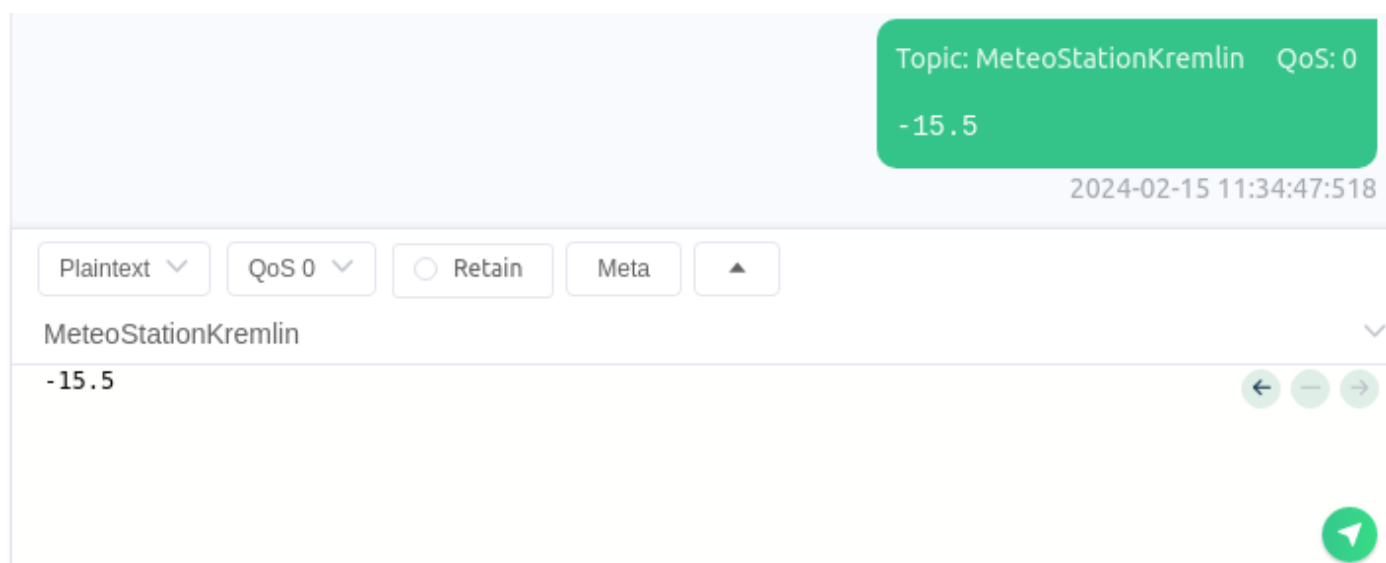
В MQTTX через кнопку (+) плюс создаем новое соединение с шифрованным вебсокетом WSS:

The screenshot shows the MQTTX application interface. On the left is a dark sidebar with a menu containing a close button, a list icon, a plus button, a code icon, a document icon, a settings icon, and a signal icon. The main area is titled 'Connections' and shows a list of connections, including 'iot@192.168.1.8:8081'. The 'New' connection configuration screen is displayed, featuring a 'General' tab. The fields are as follows:

- Name**: mosquitto-rrg-5471.gb-iot.ru
- Client ID**: mqttx_c056d262
- Host**: wss:// mosquitto-rrg-5471.gb-iot.ru
- Port**: 443
- Path**: /IoT
- Username**: admin
- Password**: [masked]
- SSL/TLS**: [checked]
- SSL Secure**: [checked]
- ALPN**: [empty]
- Certificate**: CA signed server certificate (selected)

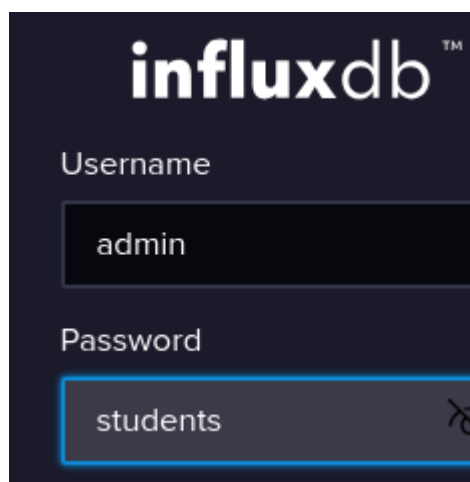
Нажав кнопку Connect соединяемся через nginx с Mosquitto. Таким образом мы можем эмулировать работу оконечного оборудования.

Предположим мы имеем, несколько метеостанций в различных районах и информация поступает по защищенному протоколу на наш сервер по сети интернет. Внизу программы с помощью окошка отправки сообщения, в его заголовке напечатаем MeteoStationKremlin - это topic, и выбрав формат сообщения PlainText в теле сообщения отправим температуру -15.5

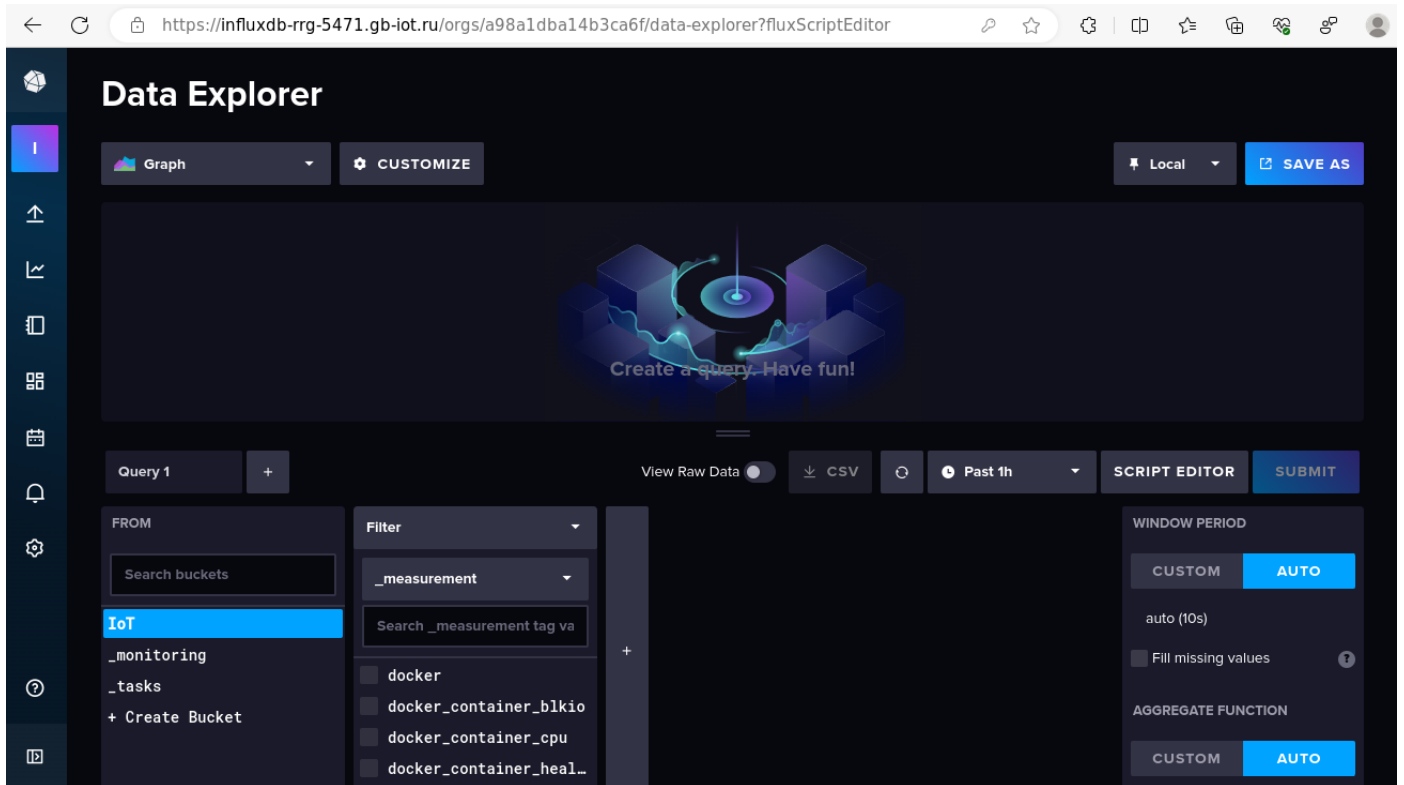


2. IndluxDB2

С помощью браузера откроем адрес <https://influxdb-rrg-5471.gb-iot.ru>



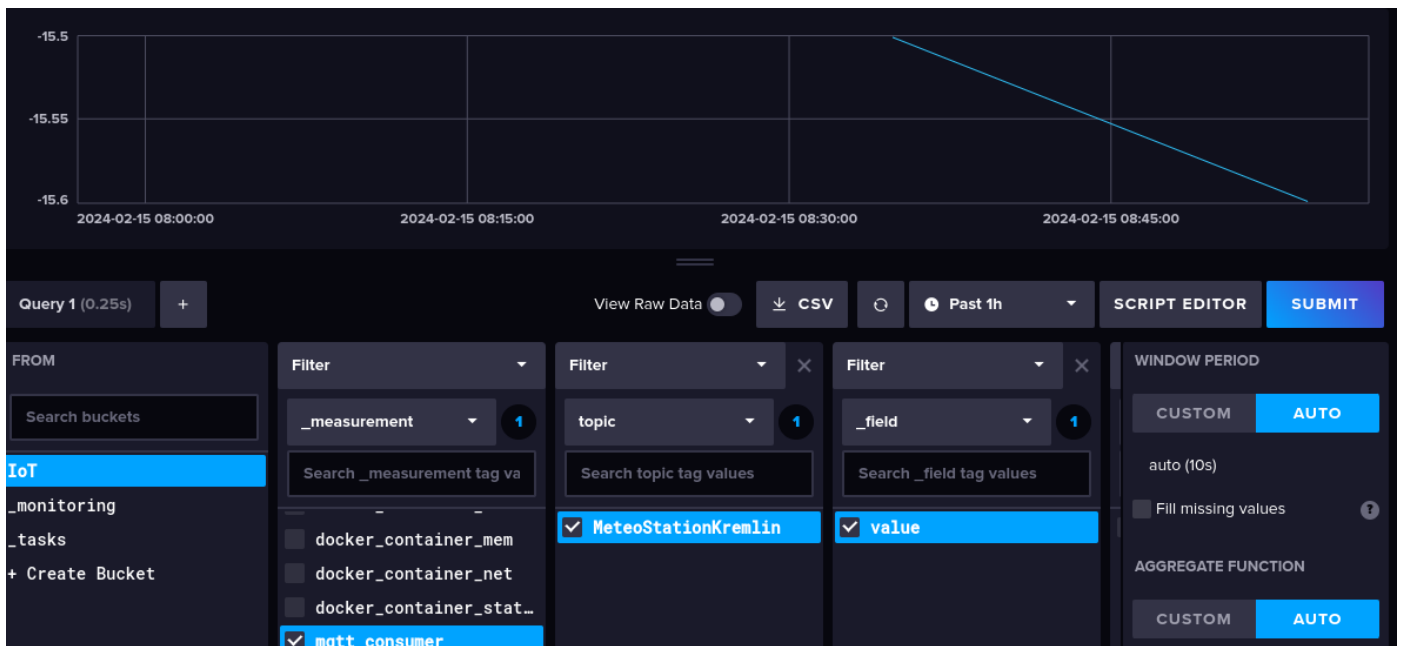
После авторизации попадаем в основное окно, нажав слева в главной вертикальной панели инструментов, кнопку Data Explorer, попадем в проводник данных:



Раздел Data Explorer состоит из двух основных областей: главной области графиков и снизу таблица запросов, состоящая из последовательно расположенных команд запроса в виде столбцов с выпадающими списками.

В первом столбце запросов FROM видим наш бакет IOT, поставив галку во втором столбце в самом низу списка на mqtt_consumer, откроется третий столбец формирования запроса, выберем в нем из выпадающего списка topic и увидим все наши сообщения разбитые по темам относящимся к нашему окончному оборудованию, тут мы наблюдаем только основную метео станцию MeteoStationKremlin, поставим на ней галку.

В появившемся четвертом столбце в списке _field поставим галку на value. Осталось нажать на кнопку SUBMIT и наш сформированный запрос методом нокод сформирует график временных рядов с показаниями температуры с нашей "метеостанции", которую сэмулировали с помощью MQTTX.



Нажав рядом с кнопкой SUBMIT другую SCRIPT EDITOR увидим реальный код запроса, а активировав переключатель View Raw Data наш график отобразит детальную информацию отправленную с метеостанции:

table	_measurement	_field	_value	_start	_stop	_time	host
mean	group	string	double	group	no group	dateTime:RFC3339	group
				dateTime:RFC3339	dateTime:RFC3339	dateTime:RFC3339	string
0	mqtt_consumer	value	-15.5	2024-02-15T07:56:54.094Z	2024-02-15T08:56:54.094Z	2024-02-15T08:34:50.000Z	10e865b2623

1 2

Query 1 (0.25s) View Raw Data CSV Past 1h QUERY BUILDER SUBMIT

```
1 from(bucket: "IoT")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "mqtt_consumer")
4   |> filter(fn: (r) => r["topic"] == "MeteoStationKremlin")
5   |> filter(fn: (r) => r["_field"] == "value")
6   |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
7   |> yield(name: "mean")
```

Filter Functions...

Transformations

aggregate.rate

chandeMomentumOscillator

columns

cov

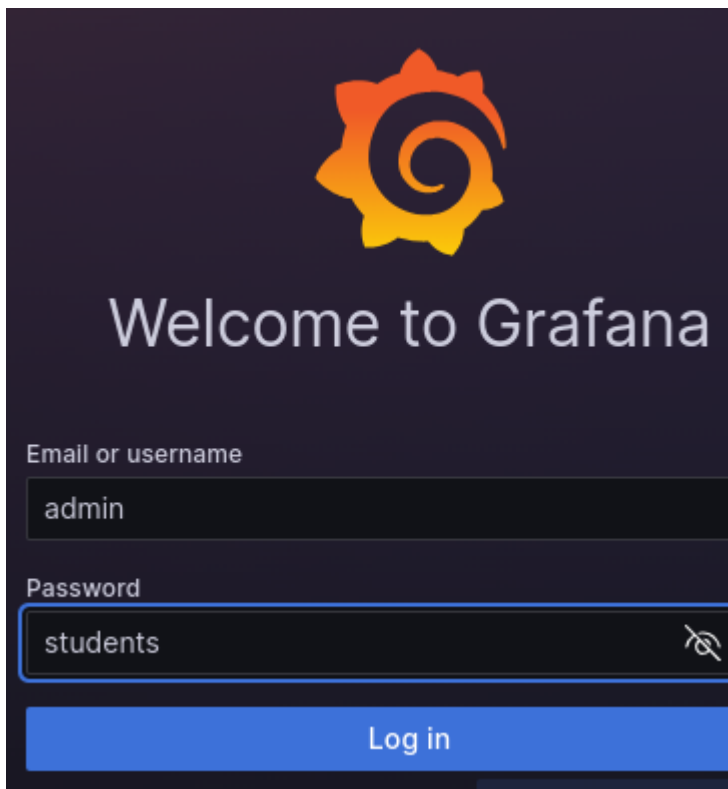
Functions

Variables

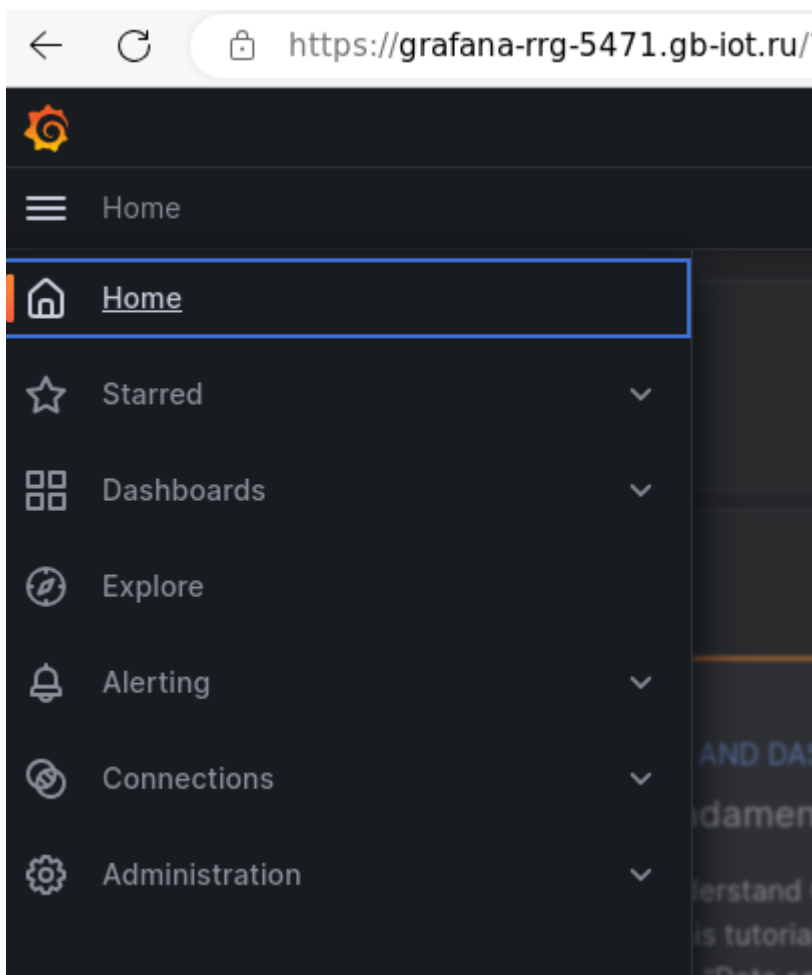
Код запроса можно использовать для построения графиков и спидометров в дашбордах Grafana.

3. Grafana

В браузере откроем адрес <https://grafana-rrg-5471.gb-iot.ru>

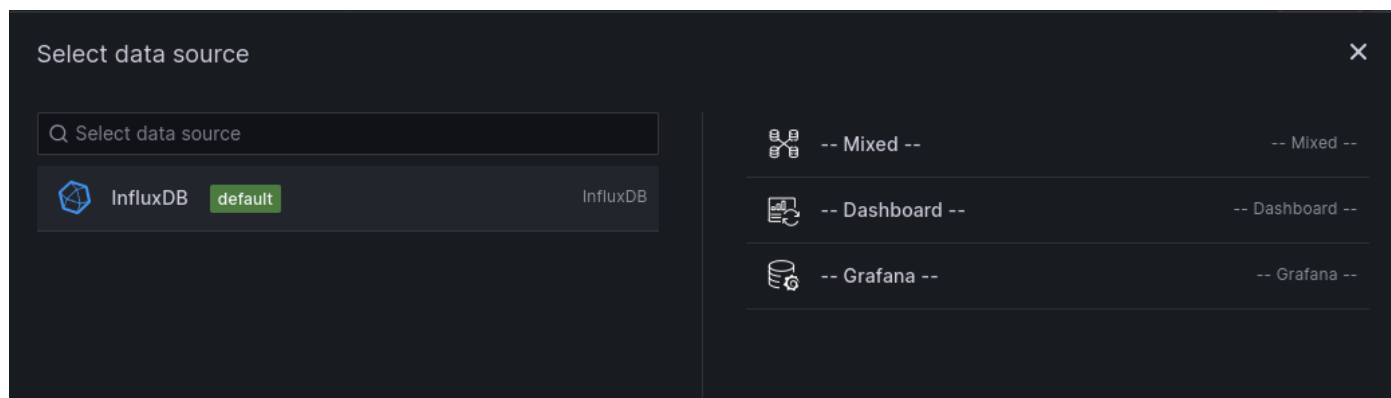


После авторизации через главное меню слева экрана войдем в раздел Дашбордов

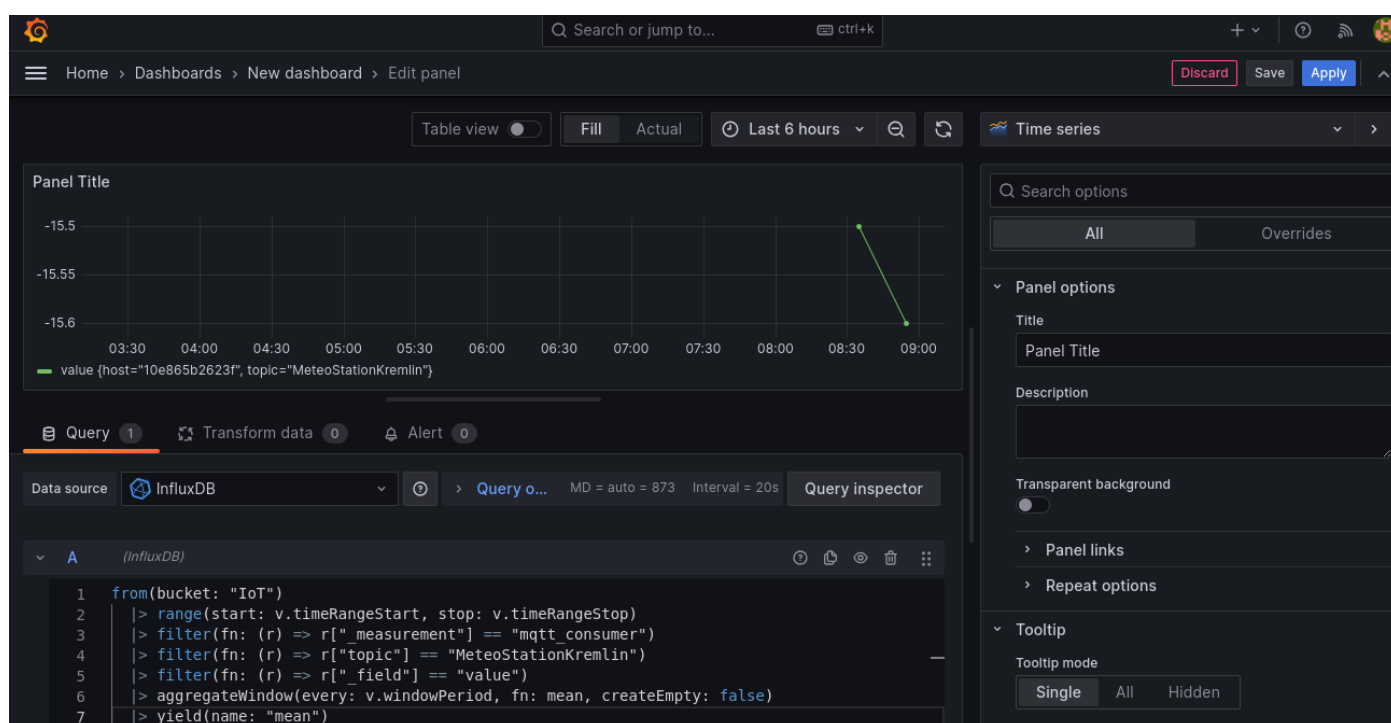


Прежде всего дашборд нужно создать нажав на кнопку +Create Dashboard, в следующем окне нажимаем кнопку +Add visualization.

В окне Select data source выберем нашу базу InfluxDB, кликнув по ней один раз.



После этого откроется главное окно нашего дашборда, внизу которого расположено окно запросов, туда мы вставим наш код из второго пункта InfluxDB2



Тут имеется много настроек, главный смысл которых, организация удобного представления данных для аналитики, можно настроить автообновления и тогда будем наблюдать картинку происходящего в режиме реального времени. После завершения настроек нажав Apply свехру справа. Останутся только графики, либо спидометры с гистограммами, которые можно уменьшать перемещать и делать заметки.

4. NodeRed

Откроем в браузере <https://nodered-rrg-5471.gb-iot.ru>



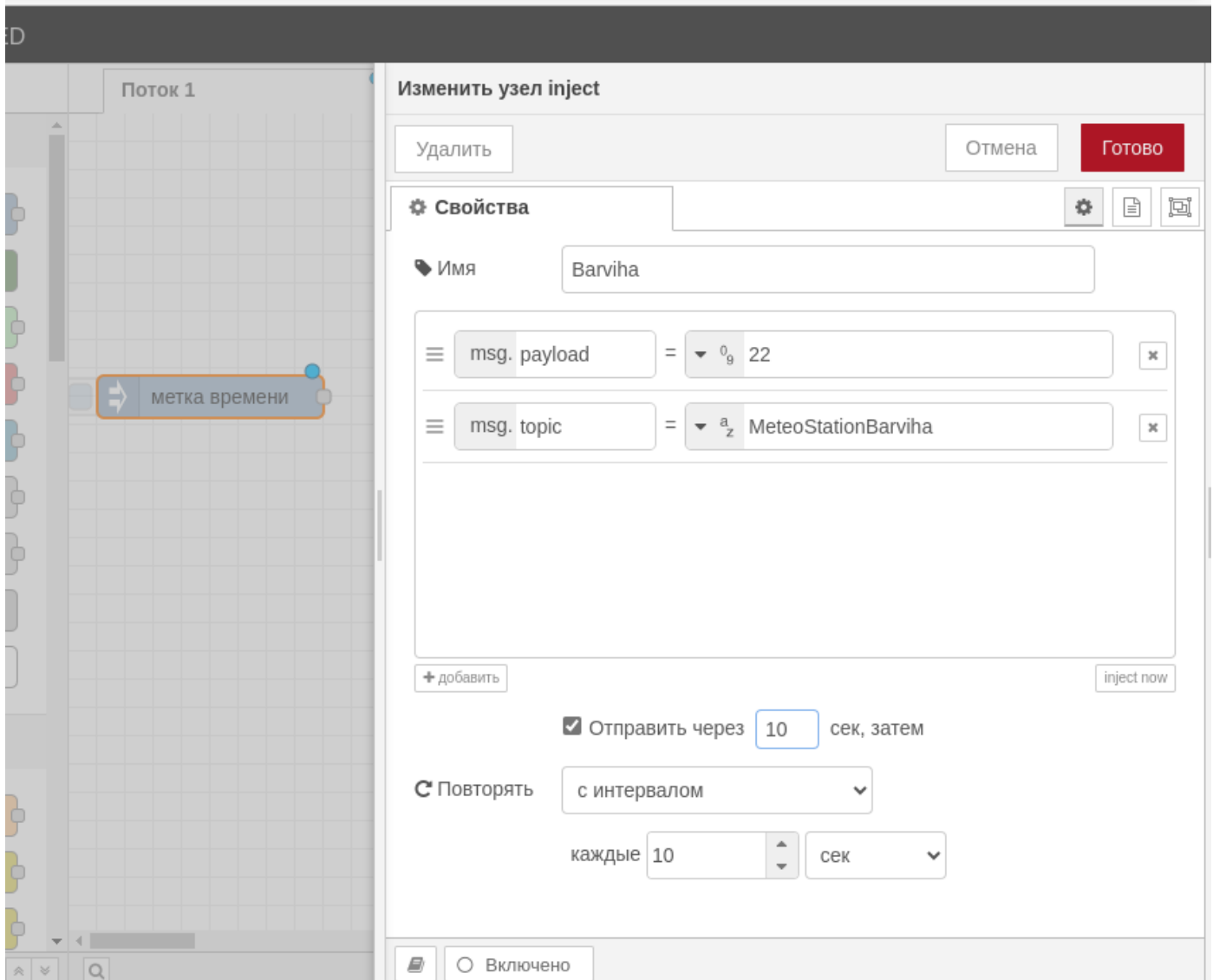
Имя
пользователя:

Пароль:

После авторизации, попадаем в мощнейший инструмент автоматизации, эмуляции, фильтрации, отладки, подключения и парсинга данных для IoT.

Сэмулируем еще одну метеостанцию MeteoStationBarviha:

1. Для этого из общих элементов перетащим на основное поле потока 1 метку времени (inject), двойным щелчком по метка времени откроем ее параметры и зададим их:



The screenshot shows the Node-RED web interface. On the left, a flow canvas labeled 'Поток 1' contains a 'метка времени' (Time Marker) node. On the right, the 'Изменить узел inject' (Edit inject node) configuration panel is open. The panel has a title bar with 'Удалить' (Delete), 'Отмена' (Cancel), and 'Готово' (Done) buttons. Below the title bar is a 'Свойства' (Properties) section with a gear icon and three sub-panels: 'Имя' (Name), 'msg. payload', and 'msg. topic'. The 'Имя' field contains 'Barviha'. The 'msg. payload' field has a dropdown set to '0₉' and the value '22'. The 'msg. topic' field has a dropdown set to 'a_z' and the value 'MeteoStationBarviha'. Below these fields are '+ добавить' (Add) and 'inject now' buttons. At the bottom of the panel, there is a checkbox 'Отправить через' (Send after) which is checked, followed by a text input '10' and the unit 'сек, затем' (seconds, then). Below this is a 'Повторять' (Repeat) section with a dropdown set to 'с интервалом' (with interval), and a 'каждые' (every) section with a text input '10' and a unit dropdown set to 'сек' (seconds). At the very bottom of the panel is a 'Включено' (Enabled) checkbox.

2. Выберем из раздела сеть элемент mqtt out и перетащим его на основное поле после метки времени Barviha, двойным кликом перейдем к настройкам mqtt, прежде всего нужно настроить сервер mqtt куда будут посылаться сгенерированные данные от NodeRed:

Поток 1

Изменить узел mqtt out > **Добавить новый конфиг узел mqtt-broker**

Отмена **Добавить**

Свойства

Имя

Соединение | Безопасность | Сообщения

Сервер Порт

☒ Connect automatically

☐ TLS

Protocol

ID клиента

Keep-alive время (сек)

Session ☒ Use clean start

Session Expiry (secs)

User Properties

☐ Включено ☐ 0 На всех потоках

Не забываем во вкладке безопасность задать пароль и юзера от Mosquitto

Изменить узел mqtt out > **Добавить новый конфиг узел mqtt-broker**

Отмена **Добавить**

Свойства

Имя

Соединение | **Безопасность** | Сообщения

Имя польз.

Пароль

После настроек нажимаем красную кнопку Добавить -> В теме не забываем указать MeteoStationBarviha и нажимаем красную кнопку Готово

Изменить узел mqtt out

УдалитьОтменаГотово

Свойства

СерверMeteoStation

ТемаMeteoStationBarviha

3. Для отладки можем добавить третий элемент на поле из Общие debug, далее необходимо соединить метку времени Barviha с debug1 и с MeteoStationBarviha. После нажимаем основную красную кнопку Deploy (Развернуть). И нажав на жучка в правый панели info (инфо) будем наблюдать свежесгенерированные данные

nodered-rrg-5471.gb-iot.ru/#flow/6003a684c341bcc9

Развернуть

Поток 1

Barviha

debug 1

MeteoStationBarviha

подключен

отладка

все узлыall

MeteoStationBarviha : msg.payload : number
22

15.02.2024, 09:38:14 node: debug 1
MeteoStationBarviha : msg.payload : number
22

15.02.2024, 09:38:24 node: debug 1
MeteoStationBarviha : msg.payload : number
22

15.02.2024, 09:38:34 node: debug 1
MeteoStationBarviha : msg.payload : number
22

15.02.2024, 09:38:44 node: debug 1
MeteoStationBarviha : msg.payload : number
22

Barviha

Вернемся в InfluxDB и обновим страницу Data Explorer

table	_measurement	_field	_value	_start	_stop	_time	host
mean	group	group	no group	group	group	no group	group
string	string	string	double	dateTime:RFC3339	dateTime:RFC3339	dateTime:RFC3339	string
0	mqtt_consumer	value	22	2024-02-15T08:41:52.270Z	2024-02-15T09:41:52.270Z	2024-02-15T09:37:40.000Z	10e865b2623

1 2 3 4 5 ... 26

Query 1 (0.25s) View Raw Data CSV Past 1h SCRIPT EDITOR SUBMIT

FROM Search buckets

IoT

_monitoring

_tasks

+ Create Bucket

Filter

_measurement 1

Search _measurement tag values

docker_container_mem

docker_container_net

docker_container_stat_

mqtt_consumer

Filter

topic 1

Search topic tag values

MeteoStationBarviha

MeteoStationKremlin

Filter

_field 1

Search _field tag values

value

WINDOW PERIOD

CUSTOM AUTO

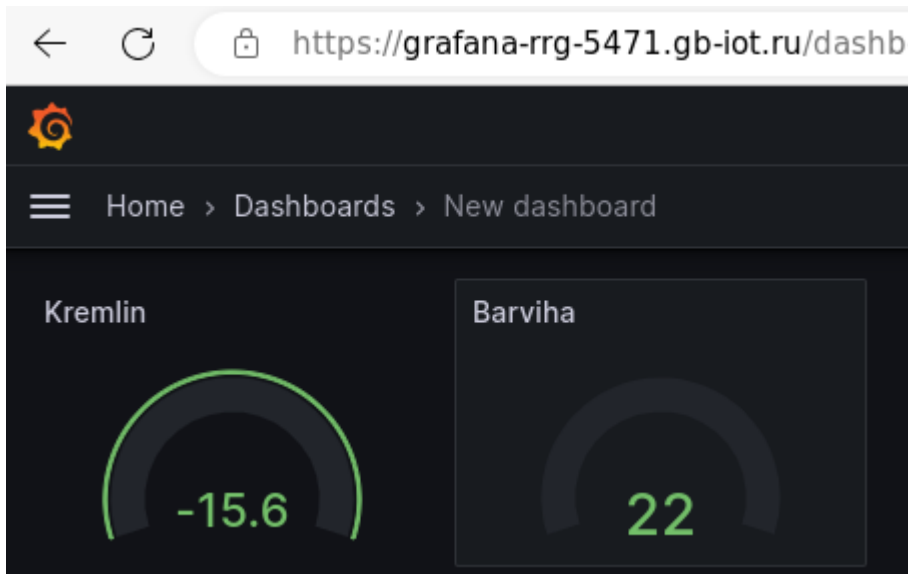
auto (10s)

Fill missing values

AGGREGATE FUNCTION

CUSTOM AUTO

Наблюдаем вновь появившиеся данные с дачи в Борвихе, скопируем код запроса в Графану и добавим визуализацию +Add:



Nginx

Благодаря конфигурации Nginx в качестве обратного прокси, мы можем использовать всего лишь один открытый порт 443 и обернуть передаваемые данные и трафик с систем визуализации в защищенное соединения, при этом используя только один общий сертификат безопасности.

/etc/nginx/conf.d/iot.conf

```
server {
    listen 443 ssl;
    server_name mosquitto-rrg-5471.gb-iot.ru;
```

```

location / {
    proxy_http_version 1.1;
    proxy_pass http://192.168.1.8:8081;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
}

ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
include /etc/letsencrypt/options-ssl-nginx.conf;
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}

server {
    listen 443 ssl;
    server_name influxdb-rrg-5471.gb-iot.ru;
    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://192.168.1.8:8086;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

server {
    listen 443 ssl;
    server_name grafana-rrg-5471.gb-iot.ru;
    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://192.168.1.8:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

server {
    listen 443 ssl;
    server_name nodered-rrg-5471.gb-iot.ru;
    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_pass http://192.168.1.8:1880;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

```
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
}
```