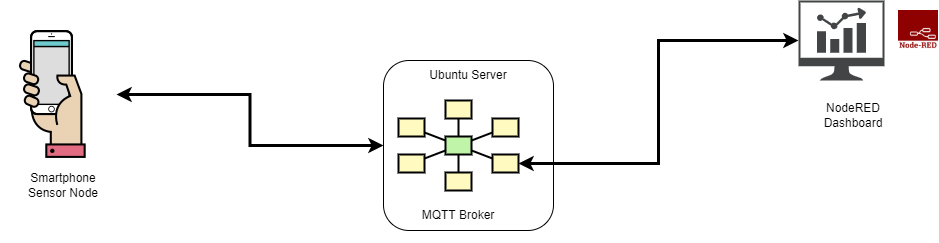
|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ ПО РАБОТЕ №2** | | |
|  | | |
| по дисциплине «Программирование сетей и инфокоммуникаций  вычислительных комплексов» | | |
| Выполнила | | |
| студентка гр. 3540904/10202 |  | Шалгуева С.Л. |
| Руководитель | | |
|  |  | Медведев Б.М. |
|  | | |

**Цель работы**

Цель выполнения данной работы создать простую систему Интернета вещей, которая может использовать легкодоступные компоненты, такие как облачные службы и смартфон. Схема такой системы представлена на рис. 1



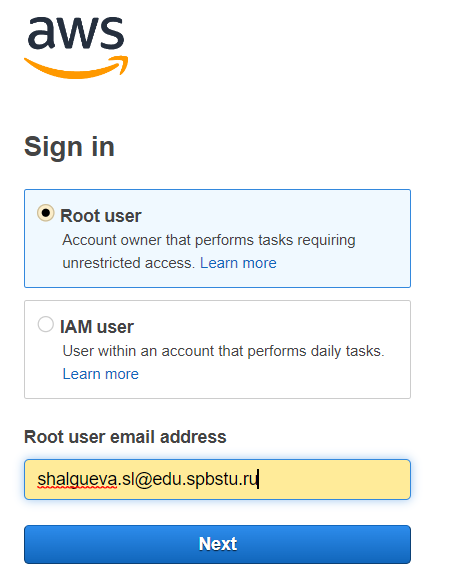
В этой системе все устройства могут взаимодействовать друг с другом, проходя через брокера MQTT. Задача брокера MQTT состоит в том, чтобы гарантировать, что сообщения доставляются нужным клиентам, в то время как клиенты также могут отправлять сообщения в режиме «один ко многим». Мы используем Node-RED, инструмент визуального программирования с открытым исходным кодом, для создания информационной панели для визуализации данных акселерометра в реальном времени, отправляемых со смартфона.

В качестве виртуального сервера выбрана платформа Amazon AWS, так как у меня уже есть аккаунт для работы с этой платформой.

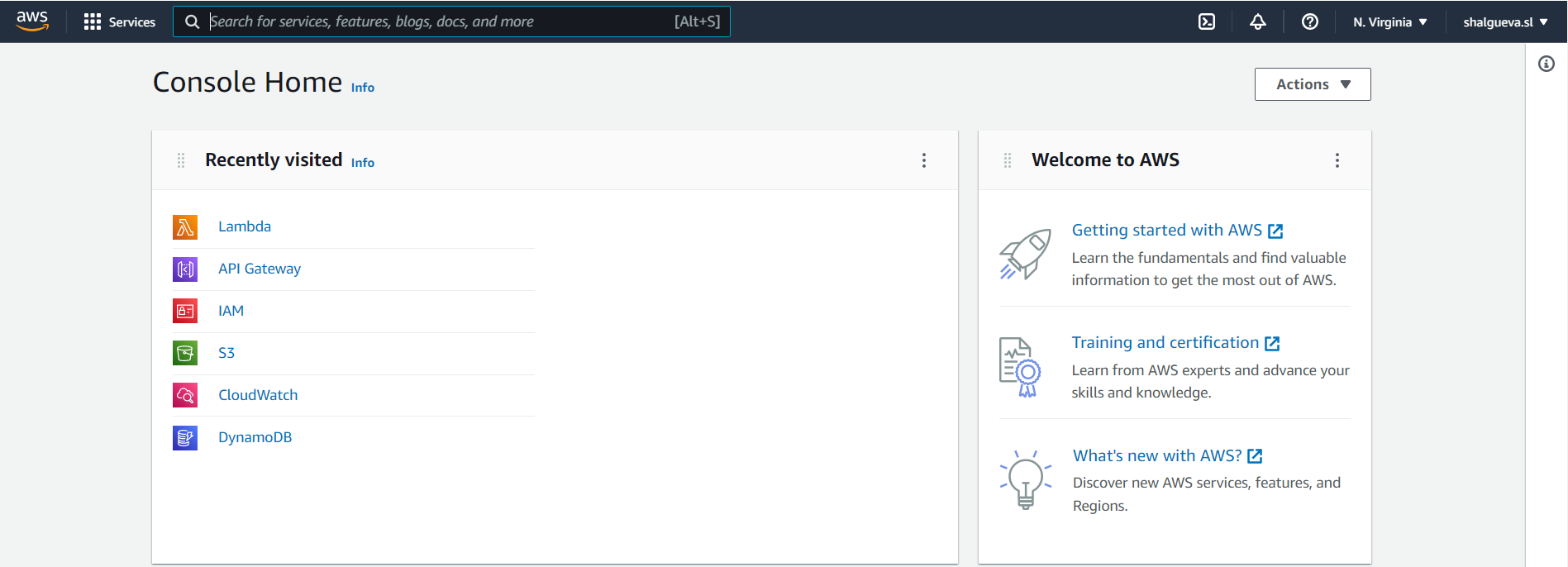
**Ход работы**

1. Установка MQTT брокера на виртуальный сервер.

Входим в аккаунт Amazon AWS.



И попадаем на «домашнюю» страницу

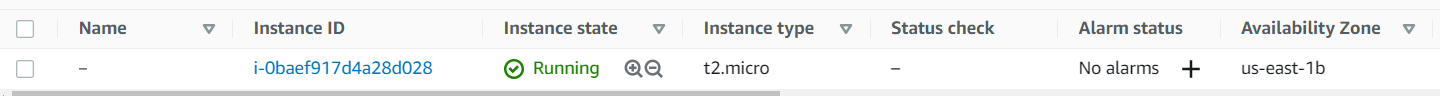


Перейдём к странице сервиса EC2 - который предоставляет вычислительные мощности путем предоставления виртуальных/физических машин с заранее созданными образами операционных систем.

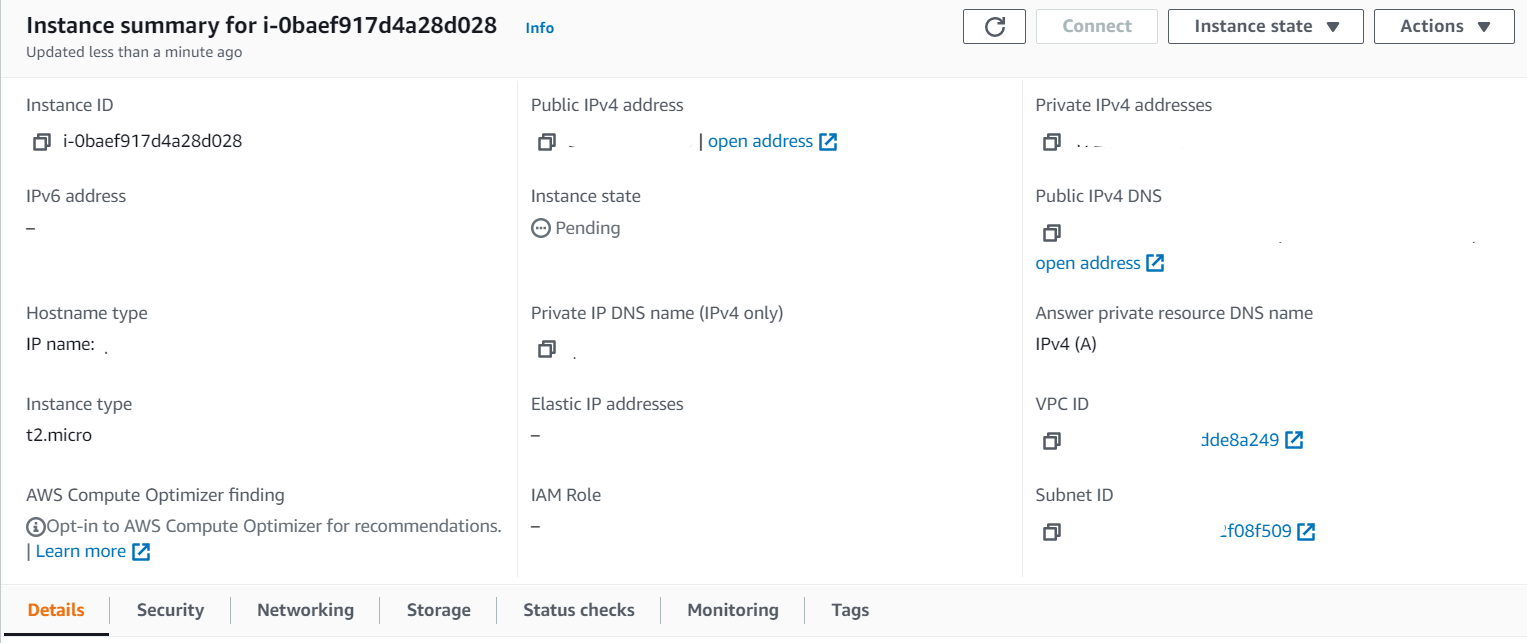
Изображение выглядит как текст

Автоматически созданное описание

Выбрали опцию LaunchInstance и установили сервер Ubuntu 20.04. После установки он появился в списке сервисов.



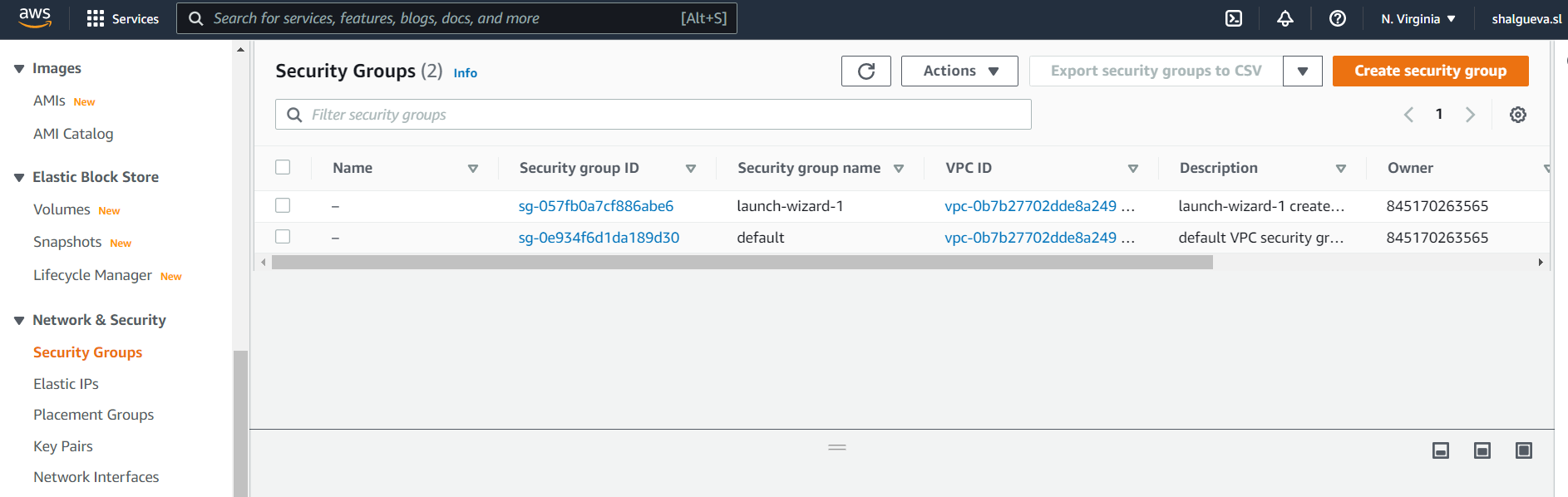
Можно также посмотреть информацию о деталях (некоторая информация скрыта в целях безопасности)



Изображение выглядит как текст

Автоматически созданное описание

Для настройки безопасности перейдём к вкладке Network&Security.



И добавим новое правило, которое откроет порт для доступа по протоколу TCP и через порт 1833.

Изображение выглядит как текст, снимок экрана, внутренний

Автоматически созданное описание

Можно соединяться с сервером. Тут же указан публичный IP адрес. По котором потом можно получить доступ ко всем запущенным службам.

Изображение выглядит как текст

Автоматически созданное описание

Открылось окно терминала, представляющего из себя обычный терминал Ubuntu.

Изображение выглядит как текст

Автоматически созданное описание

Можно переходить к установке Mosquitto. Вводим в терминал предложенные команды для обновления установленных пакетов, установки mosquito, конфигурирования конфигурационного файла /mosquitto.conf и настроить файервол.

|  |
| --- |
| sudo apt update  sudo apt upgrade  sudo apt install mosquitto -y  sudo apt-get install mosquitto-clients  sudo apt clean  sudo service mosquitto stop  sudo nano /etc/mosquitto/mosquitto.conf  sudo apt install ufw  sudo ufw allow 1883  sudo ufw allow 22  sudo ufw enable |

В конфигурационный файл необходимо дописать

|  |
| --- |
| listener 1883  allow\_anonymous true |

После этого остаётся запустить сервис mosquito.

Изображение выглядит как текст

Автоматически созданное описание

1. Установка и настройка Node-RED на AWS EC2

На вкладке Network&Security необходимо добавить ещё одно правило для порта 1880. Теперь набор правил выглядит вот так:

Изображение выглядит как текст, снимок экрана, внутренний, ноутбук

Автоматически созданное описание

После настройки правил необходимо снова подключиться к терминалу и установить необходимые зависимости и сервисы – node.js и Node-RED.

|  |
| --- |
| curl -sL https://deb.nodesource.com/setup\_12.x | sudo -E bash –  sudo apt-get install -y nodejs build-essential  sudo npm install -g --unsafe-perm node-red |

Запуск сервиса осуществляется с помощью команды node-red.

Изображение выглядит как текст

Автоматически созданное описание

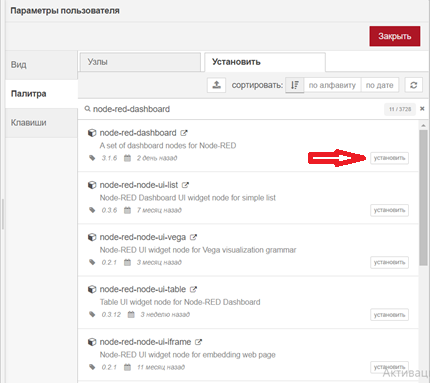
Теперь можно из терминала отправить curl запрос на 127.0.0.1:1880. Или воспользоваться открытым IPv4 адресом нашего сервера и получить такую страницу в браузере.

Изображение выглядит как стол

Автоматически созданное описание

Установка Node-RED успешно прошла.

Осталось настроить конфигурацию, загрузив дополнительную палитру.



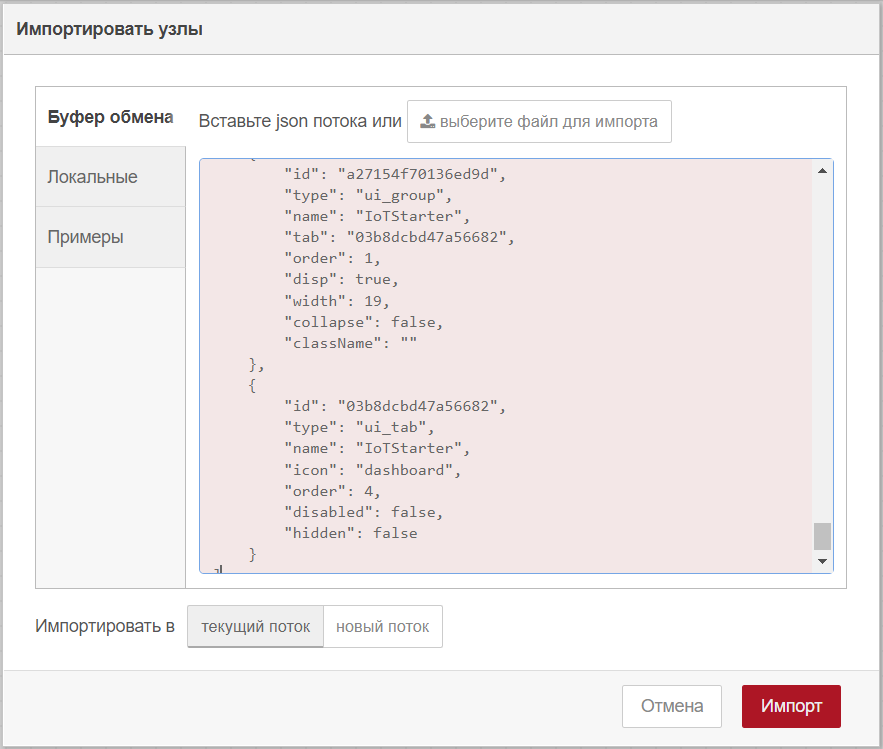
Установлено.

Изображение выглядит как текст

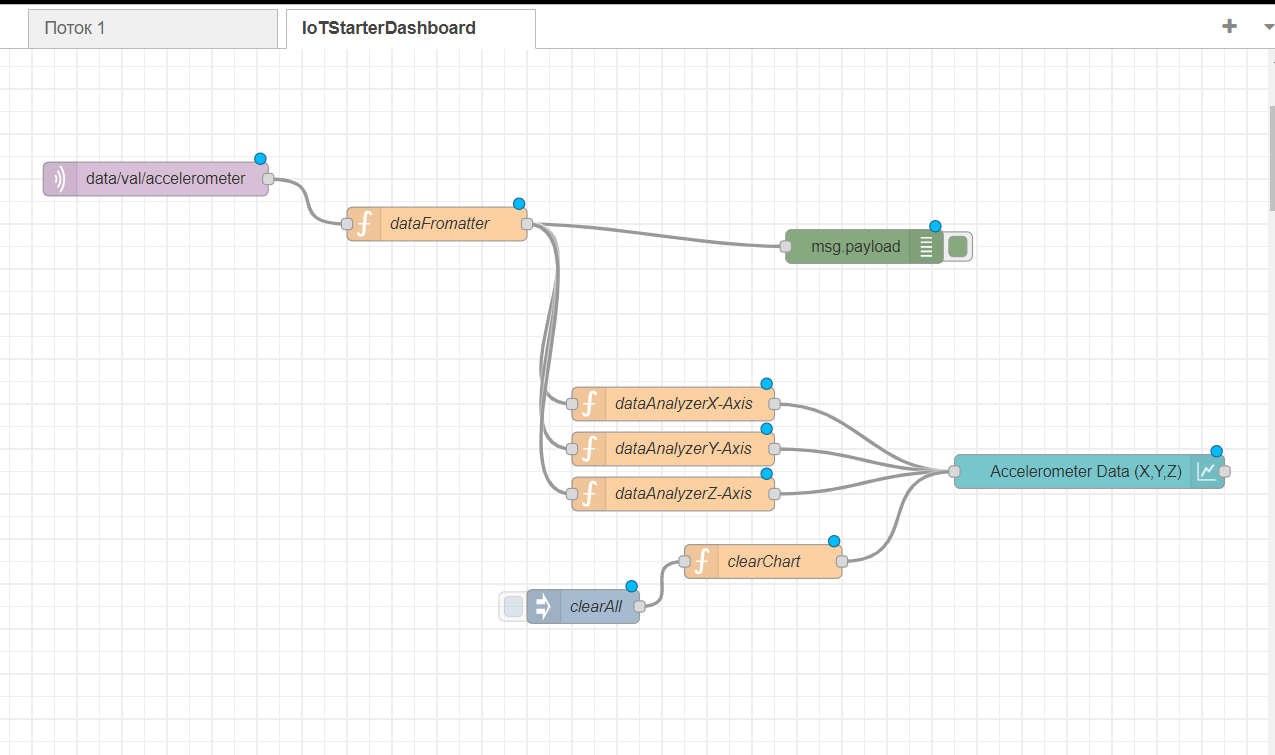
Автоматически созданное описание

1. Создание информационной панели Node-RED

Для создания панели воспользуемся готовой конфигурацией. Перейдём по ссылке из методический рекомендация и скопируем json файл. Полученный текст вставим в окошко импорта.



И увидим вот такую схему



Фиолетовый элемент – узел для связи с сервером mosquito. Сейчас в нём заданы дефолтные настройки.

Изображение выглядит как текст

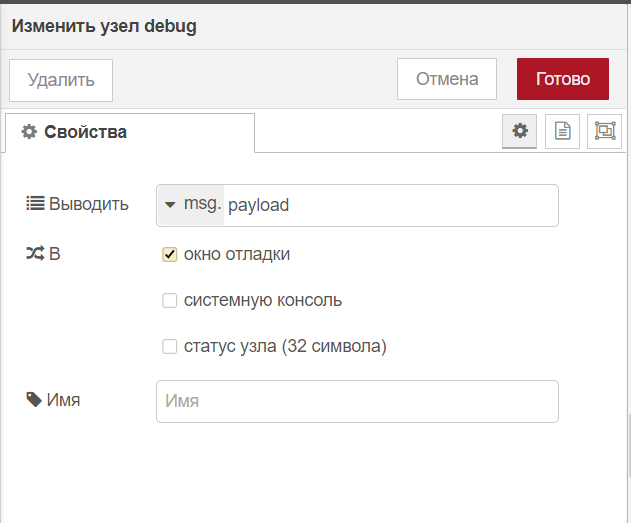
Автоматически созданное описание

Оранжевые – узлы функций, которые содержат код для управления данными и их обработки.

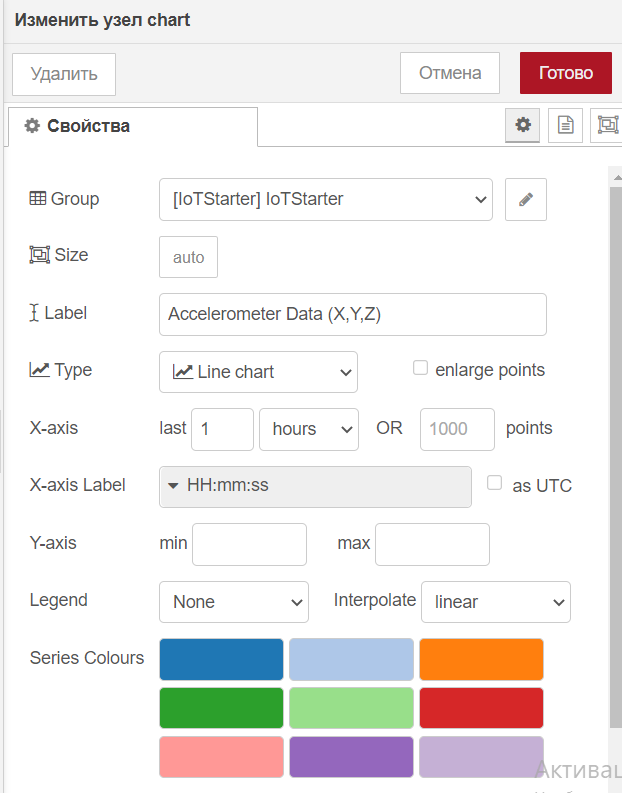
Изображение выглядит как текст

Автоматически созданное описание

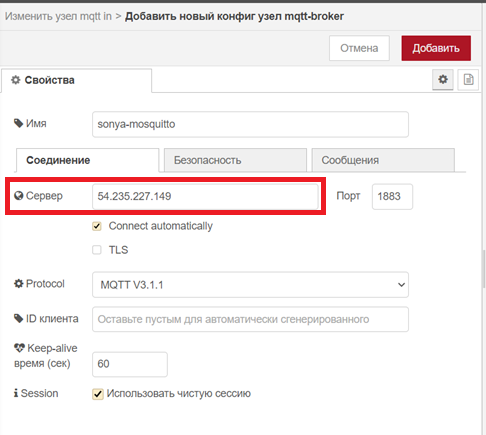
Зелёный – вспомогательный узел для отладки.



Синий – для отображения диаграммы.



Необходимо настроить фиолетовый mqtt узел для работы с mosquito, который запущен на нашем сервере Ubuntu. Для этого снова зайдём в настройки узла и укажем публичный IP адрес сервера.



После настройки самое время развернуть приложение Node-RED. Для этого нажимаем на кнопку Развернуть (Deploy) и следим за статусом развертывания.

Изображение выглядит как стол

Автоматически созданное описание

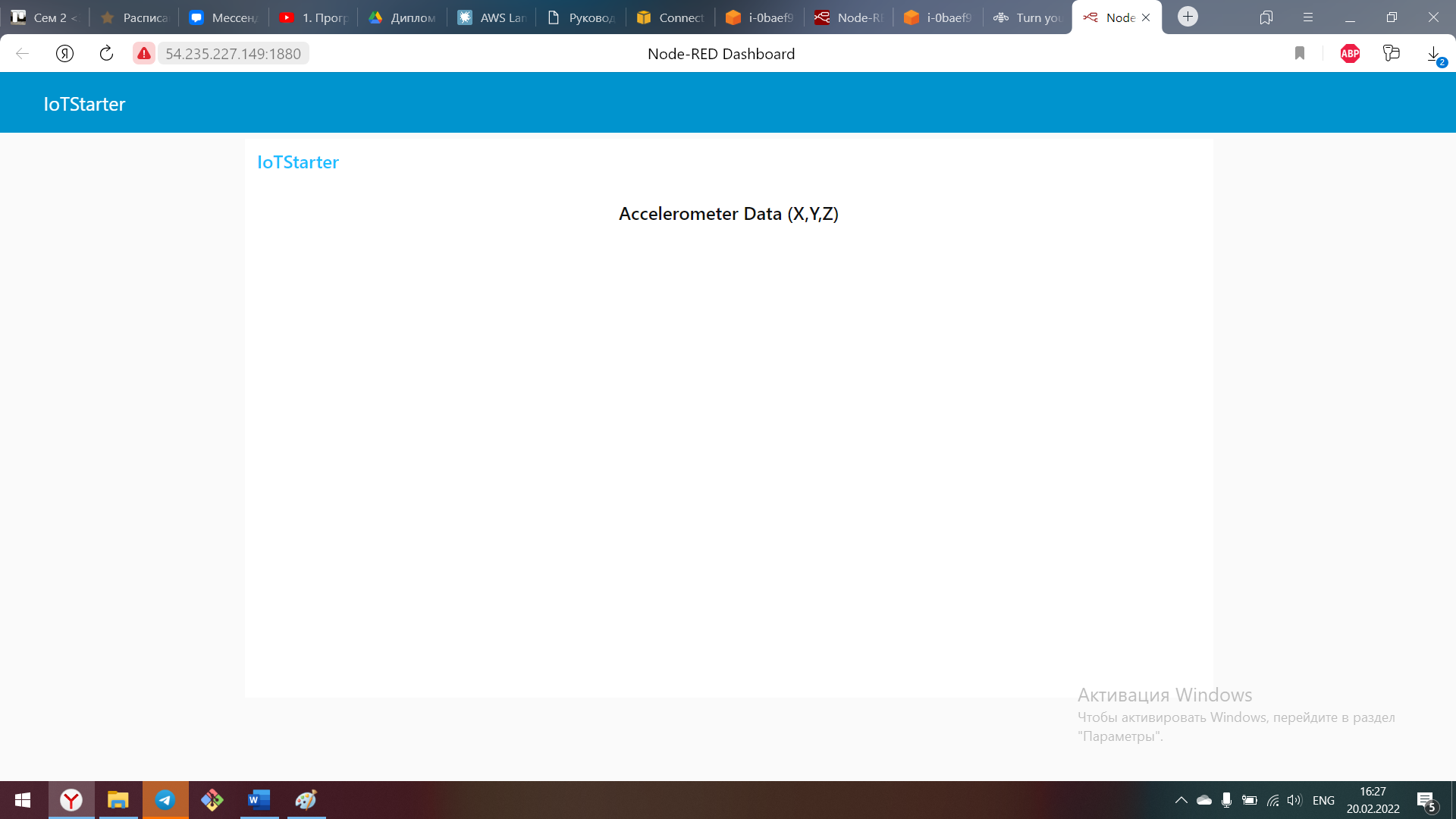
На данном этапе обнаружилось, что я допустила ошибку при настройке Network&Security и вместо порта 1883 для MQTT указала порт 1833. Поэтому подключение долго не происходило. После указания правильного порта картинка поменялась на:

Изображение выглядит как текст, окно, седзи, внутренний

Автоматически созданное описание

Перейдём к интерфесу пользователя, который располагается по адресу

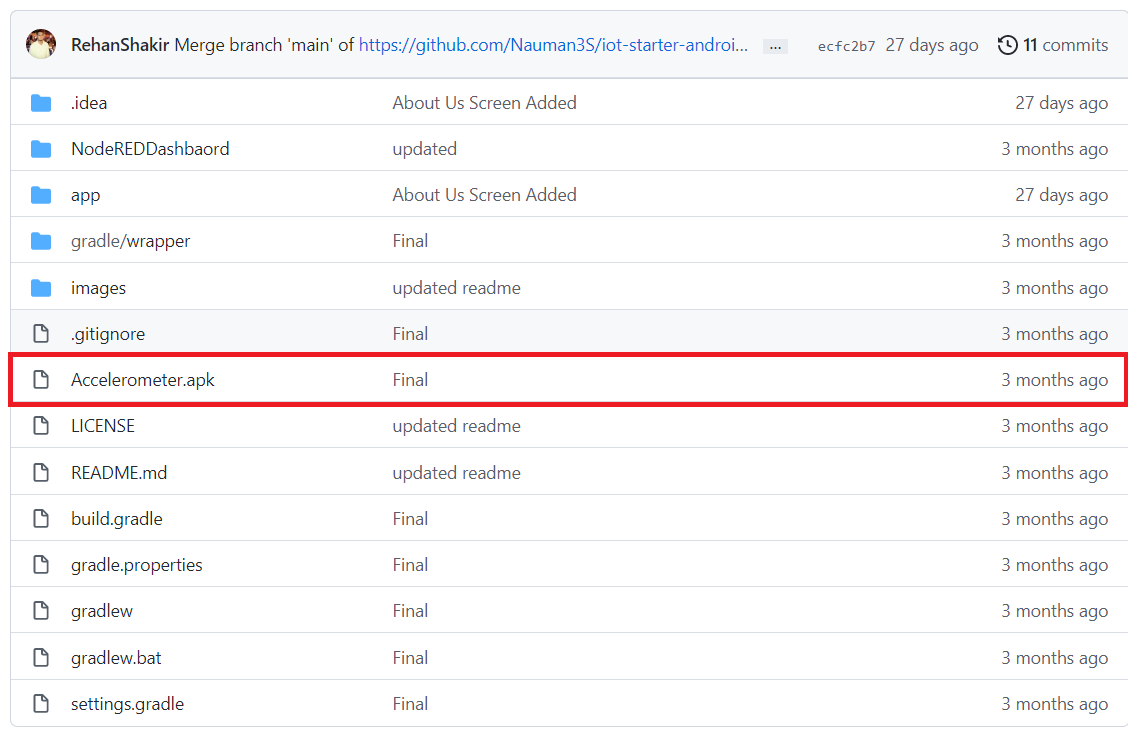
http://<server public ip>:1880/ui



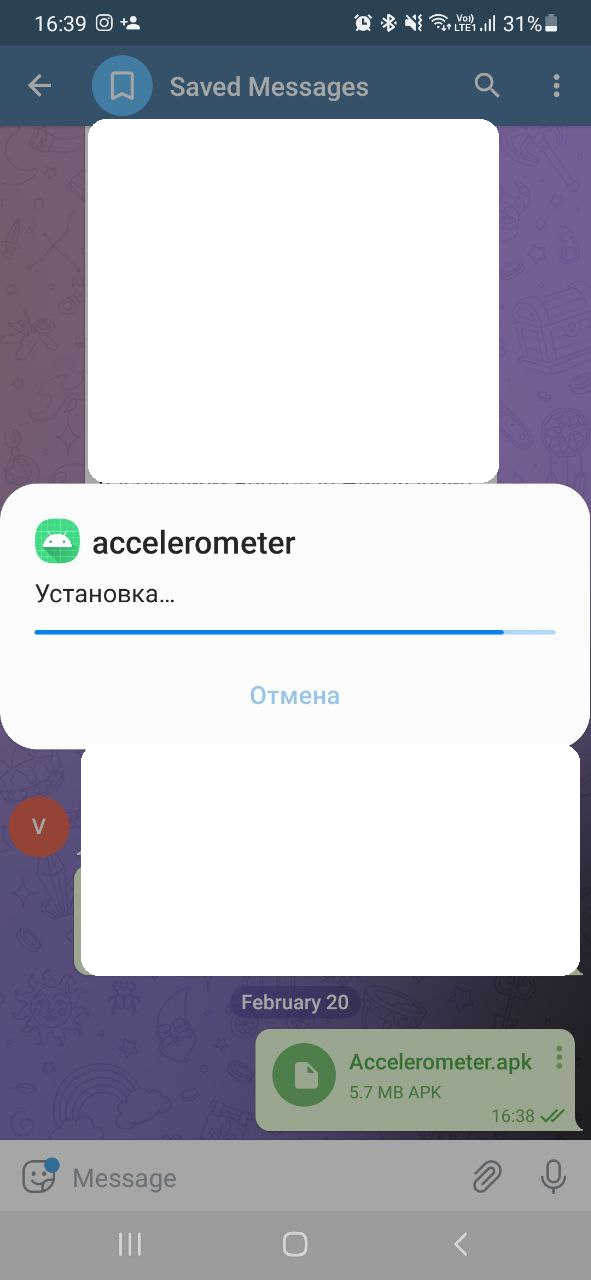
Мы готовы принимать данные от приложения.

1. Установка приложения на Android

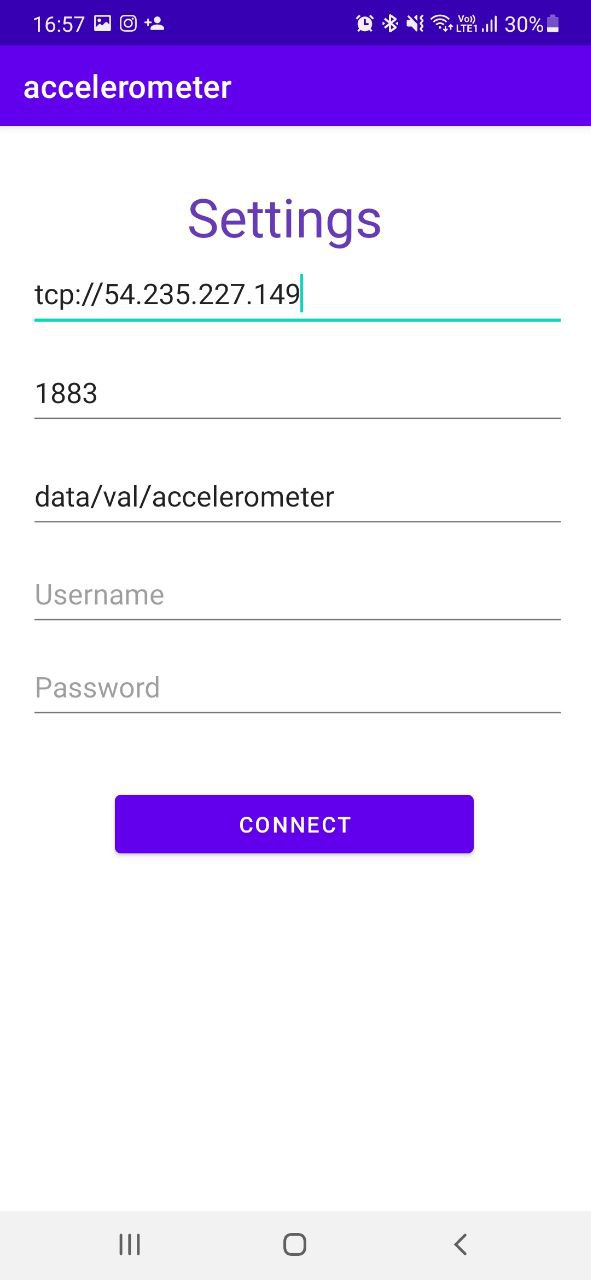
Сначала надо перейти в репозиторий из методических указаний и скачать *Accelerometer.apk* файл



И установить приложение на смартфон на платформе Android.



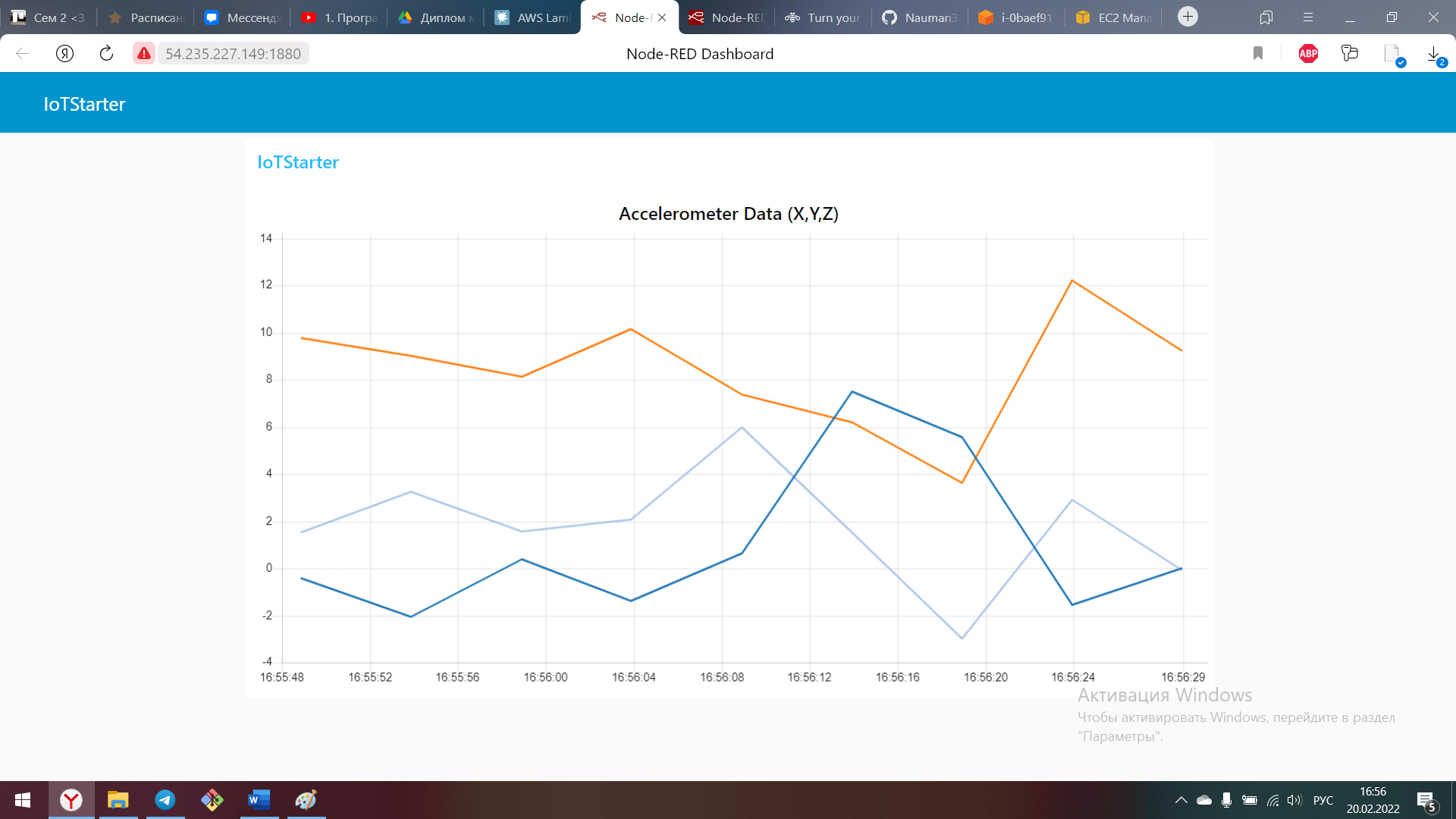
Сконфигурируем приложение, чтобы оно могло работать с нашим сервером. Укажем открытый IP-адрес, открытый порт (по умолчанию 1883) и топик, который будет слушать сервер. Так как я не меняла никаких настроек на сервере, осталю имя топика по умолчанию data/val/accelerometer.



Нажмём кнопку Start и вернёмся к интерфейсу, на котором теперь отображается диаграмма.

Изображение выглядит как стол

Автоматически созданное описание



**Выводы**

В ходе этой работы мне удалось получить новые знания как с помощью смартфона и сервера сконфигурировать простую IoT систему. Я также познакомилась с новым сервисом Amazon AWS EC2 и научилась конфигурировать собственные сервера.