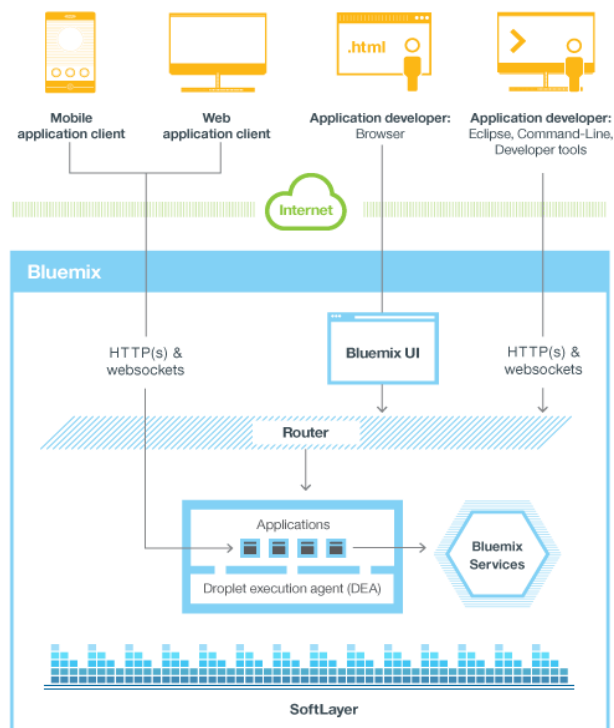


Практику по быстрому прототипированию решений Интернета вещей.

Цель работы – исследование принципов построения распределенных систем Интернета вещей с использованием платформы IBM Cloud.

Краткое описание концепций IBM Cloud

В терминологии IBM Cloud *приложение (application)* — это созданный вами артефакт, т. е. весь программный код (исходный код или исполняемые двоичные файлы), который необходимо запустить или на который необходимо сослаться в процессе исполнения. Мобильные приложения выполняются за пределами среды IBM Cloud и используют сервисы IBM Cloud, представленные приложениями. В случае веб-приложений приложение — это код, загруженный на платформу IBM Cloud с целью хостинга. Кроме того, платформа IBM Cloud способна осуществлять хостинг программного кода приложения, который вы хотите выполнять на внутреннем сервере в среде на базе контейнера.



Архитектура IBM Cloud

Сервис (service) — это код, работающий на платформе IBM Cloud и предлагающий некоторую функциональность, которую могут использовать приложения. Это может быть готовый сервис, используемый непосредственно — например, push-уведомления для мобильных приложений или эластичное кэширование для веб-приложения. Вы также можете

создавать собственные сервисы в диапазоне от простых служебных функций до сложной бизнес-логики.

Организация (organization) и пространство (space) — это организационные единицы инфраструктуры, способные хранить и отслеживать ресурсы приложения. Организация содержит домены (domain), пространства и пользователей. Пространство содержит приложения и сервисы. По умолчанию используется три пространства: Development (разработка), Production (производство) и Staging (подготовка).

Для приложений, которым требуется среда типа PaaS, предоставляются buildpack-пакеты, каждый из которых представляет собой набор скриптов для подготовки кода к исполнению на целевой PaaS-платформе. Buildpack-пакеты, которые включают необходимую вашим приложениям среду исполнения и могут также содержать специализированные инфраструктуры, упрощают развертывание приложения в облаке по сравнению с самостоятельной установкой и конфигурированием среды исполнения.

Использование сервисов в IBM Cloud

Использование сервисов в IBM Cloud включает три этапа:

- 1 Сообщите платформе IBM Cloud, что вам требуется новый экземпляр сервиса и какое конкретное приложение будет использовать этот новый экземпляр.
- 2 IBM Cloud автоматически инициализирует новый экземпляр этого сервиса и свяжет его с приложением.
- 3 Приложение взаимодействует с сервисом.

Пакеты сервисов (Service bundles) — это коллекции API-интерфейсов, используемых в конкретных областях. Например, пакет Mobile Services включает сервисы MobileData, Cloud Code, Push и Mobile Application Management. Доступные сервисы и среды исполнения представлены в каталоге IBM Cloud. Кроме того, вы можете зарегистрировать собственные сервисы.

Развертывание приложения и управление приложением

Чтобы развернуть свое приложение, необходимо загрузить его в среду IBM Cloud и указать, сколько экземпляров этого приложения должно исполняться, а затем сконфигурировать IBM Cloud, введя необходимую информацию для поддержки этого приложения.

В случае мобильного приложения среда IBM Cloud содержит артефакт, который представляет серверную часть мобильного приложения — набор сервисов, который использует приложение для взаимодействия с сервером. IBM Cloud поддерживает серверные компоненты мобильного приложения, взаимодействующие с сервисами PushWorks, Cloud Code и Mobile Data, непосредственно из пользовательского интерфейса IBM Cloud.

В случае веб-приложения необходимо предоставить в IBM Cloud соответствующую информацию о среде исполнения и среде разработки, чтобы платформа смогла сформировать надлежащую инфраструктуру для исполнения этого приложения.

При развертывании приложений и управлении ими можно использовать инструмент командной строки cf, веб-интерфейс IBM Cloud или сервисы DevOps Services.

Браузерные и мобильные клиенты — а также другие приложения, развернутые на платформе IBM Cloud и выполняющиеся за ее пределами — взаимодействуют с приложениями, работающими на платформе IBM Cloud, через API-интерфейсы типа REST/HTTP. Каждый клиентский запрос маршрутизируется к одному из экземпляров приложения или составляющих его сервисов. Среды исполнения приложений в IBM Cloud изолированы друг от друга даже тогда, когда они находятся на одной и той же физической машине.

В ходе управления приложениями можно запускать, останавливать, перезапускать экземпляры приложения (или, в случае веб-приложения, изменять их количество), а также изменять объем памяти, используемый приложением. Ключевая конструктивная особенность IBM Cloud — отличные показатели при хостинге масштабируемых приложений и артефактов приложений. На данный момент эта платформа не масштабирует приложение автоматически в соответствии с нагрузкой, поэтому этим процессом необходимо управлять самостоятельно посредством создания или удаления экземпляров при изменении рабочей нагрузки. По этой причине ваши приложения должны сохранять все персистентные данные за пределами приложения в одном из сервисов хранения данных, предоставляемых платформой IBM Cloud. При повторном развертывании приложения после обновления используется тот же процесс, что и при начальном развертывании. IBM Cloud останавливает все исполняющиеся экземпляры и переводит новые экземпляры в рабочее состояние автоматически.

Сервисы DevOps Services для IBM Cloud

При использовании DevOps Services требуется лишь несколько простых шагов для организации взаимодействия с другими специалистами с целью планирования, отслеживания и создания программного обеспечения в облаке. Вы можете воспользоваться встроенным в браузер редактором программного кода, который DevOps Services предоставляет для разработки приложений, или использовать DevOps Services с Eclipse, с VisualStudio или с инструментом командной строки Git для написания кода приложения и развертывания его на платформе IBM Cloud.

При работе с пользовательским интерфейсом, который помогает разработчику быстро добавлять сведения "кто", "что" и "когда" для своего рабочего проекта, требуется потратить всего несколько минут на задание дат, документирование первого сценария применения, назначение одной-двух задач и переход непосредственно к написанию программного кода.

DevOps Services включает встроенные средства управления исходным кодом — Jazz SCM и хостинговый Git. Каждый проект получает свой собственный репозиторий DevOps Services и рабочее пространство, в котором участники этого проекта могут регистрироваться свои изменения, ассоциировать изменения программного кода и просматривать историю недавних изменений. Кроме того, вы можете создать проект DevOps Services и указать на свой репозиторий GitHub.

Вы также можете с легкостью связать элементы работы с изменениями кода в GitHub. Кроме того, для написания кода в Git вы можете использовать имеющиеся у вас инструменты.

Типичными сценариями использования ресурса DevOps Services являются:

- Создание приложения для анализа данных социальных сетей с использованием Node.js, Node-RED, Express, sentiment и ntwitter.
- Создание приложения для создания интерактивных опросов в реальном времени с использованием Node.js, Node-RED, Express, AngularJS и MongoDB.
- Построение сервисов уведомления с использованием Node.js, Node-RED и MongoDB.
- Создание приложений для управления аппаратными устройствами с использованием Node-RED и IoT компонент.

Среда визуальной разработки JavaScript приложений Node-RED

Node-RED - это визуальная drag-and-drop среда разработки JavaScript рантаймов для IoT (неблокирующих приложений, управляемых событиями). Благодаря большому количеству примитивов и возможности быстрой визуальной настройки и созданию новых компонентов, в том числе непосредственно на языке JavaScript, Node-RED может быть использована как непрофессиональными пользователями, так и профессиональными разработчиками для ускорения создания веб-приложений в облаке. Это позволяет использовать Node-RED для взаимодействия с недорогими аппаратными платформами в рамках подхода IoT, перенося основную часть вычислительной нагрузки на облачную платформу. Помимо этого все созданные рантаймы могут быть объединены в библиотеки решений и впоследствии перенесены в другие проекты благодаря функциям экспорта и импорта.

Терминология Node-RED

Нод (node) - функционально законченный блок.

Поток обработки (flow) - цепь соединенных нодов и соответствующий им конфигурационный нод.

Входной нод (input node) - нод, принимающий внешние данные и задающий начало потоку обработки (flow). Входной нод имеет один или несколько выходных портов (output ports).

Выходной нод (output node) - нод, завершающий поток обработки и передающий результаты во внешнюю среду.

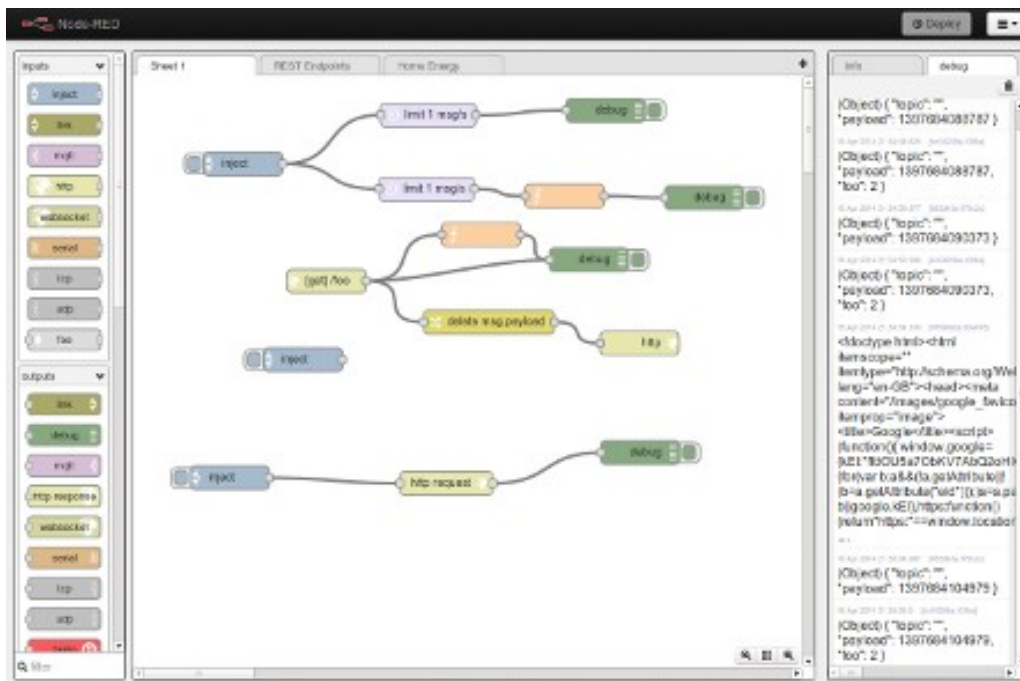
Функциональный нод (function node or query node) - нод, находящийся внутри потока обработки и имеющий один входной порт и один или несколько выходных портов.

Конфигурационный нод (config node) - нод, содержащий конфигурационную информацию, используемую в других нодах. Конфигурационный нод не связан портами с другими типами нодов.

Набор нодов (node set) - ноды, включенные в связанные js/html файлы (пары соответствующих друг другу файлов, содержащих связанный код JavaScript и HTML). Ошибка в js/html паре приводит к неработоспособности всех нодов набора.

Модуль нодов (node module) - множество связанных js/html файлов (и, соответственно, наборов нодов), где каждая пара описана в package.json файле.

Пакет нодов (node pack) - коллекция связанных по функциональности нодов, которые могут быть использованы разработчиком в рамках одного проекта.



Основное окно проекта в Node-RED

Пример использования Node-RED

Примером эффективного применения технологии Node-RED может служить проект, на разработку которого требуется около 15 минут. Приложение позволяет автоматически отслеживать текущее местоположение мобильного устройства, определяет прогноз погоды и сообщает о нем через твиттер самому пользователю (Рисунок 5).

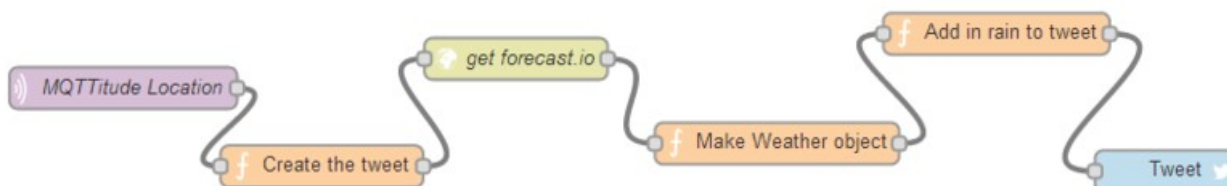


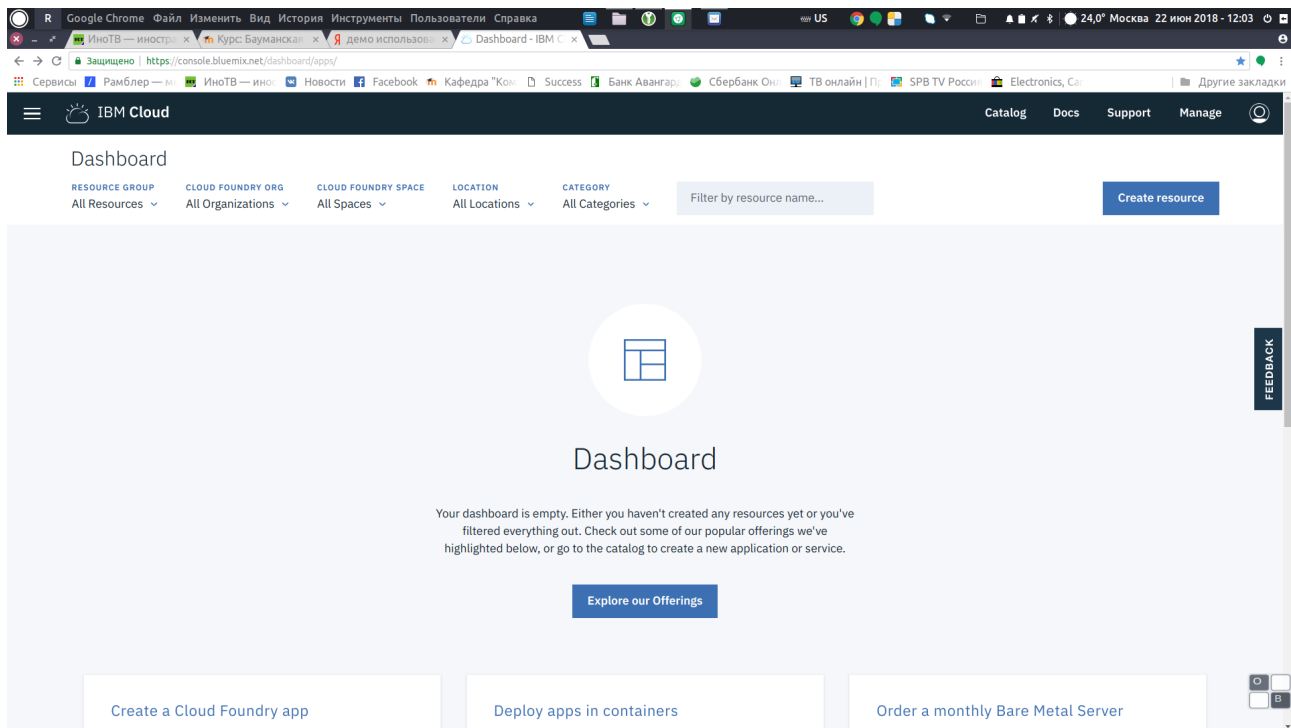
Рисунок 6 - Поток обработки для информирования о прогнозе погоды.

- 1 Приложение MQTTitude на мобильном телефоне передает телеметрическую информацию о положении устройства в обрабатывающий поток Node-RED.
1. Телеметрия о местоположении преобразуется в twitter сообщение.
2. Сообщение используется для доступа к прогнозу погоды в ноде Forecast.io.
3. Полученный прогноз форматируется в сообщение twitter.
4. Сообщение посылается на твиттер аккаунт пользователя и отображается на мобильной платформе.

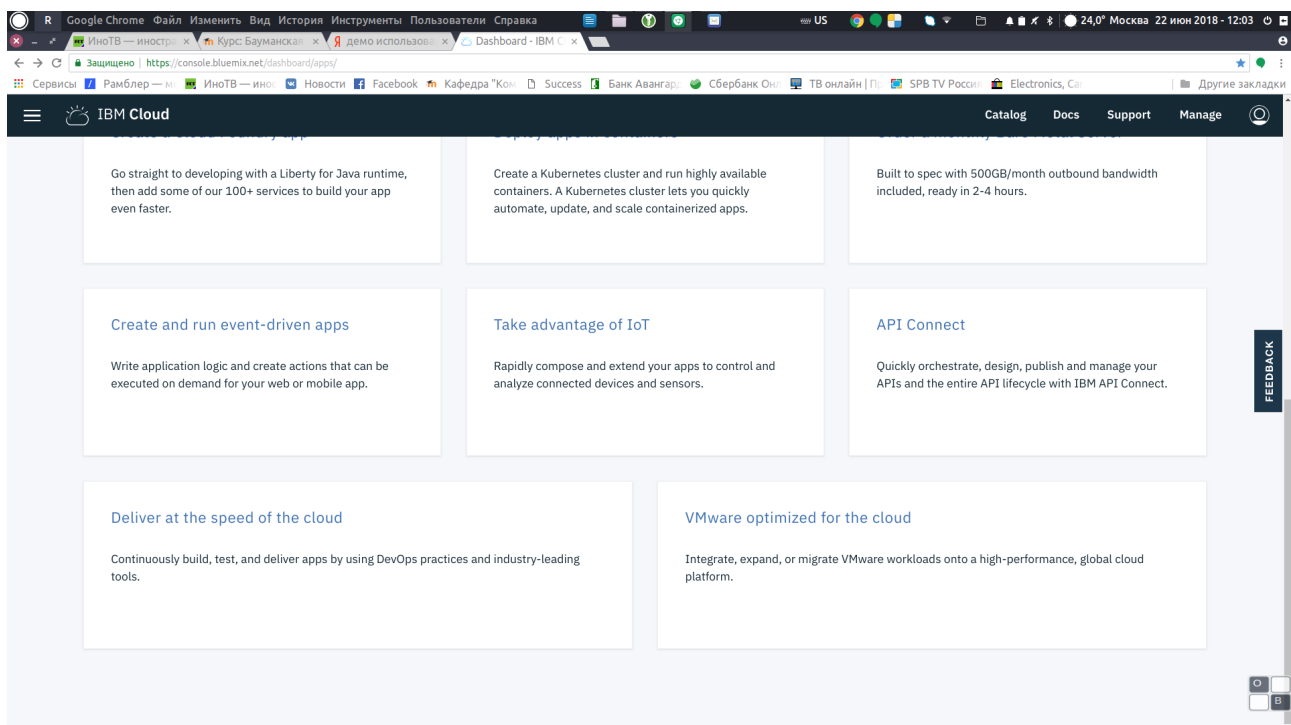
Приведенный пример позволяет реализовать логику взаимодействия с использованием протоколов MQTT, HTTP и Twitters API без глубокого погружения в их технические особенности.

ЭТАП 1. Создание проекта IoT в облачной платформе IBM Cloud (ранее IBM Cloud)

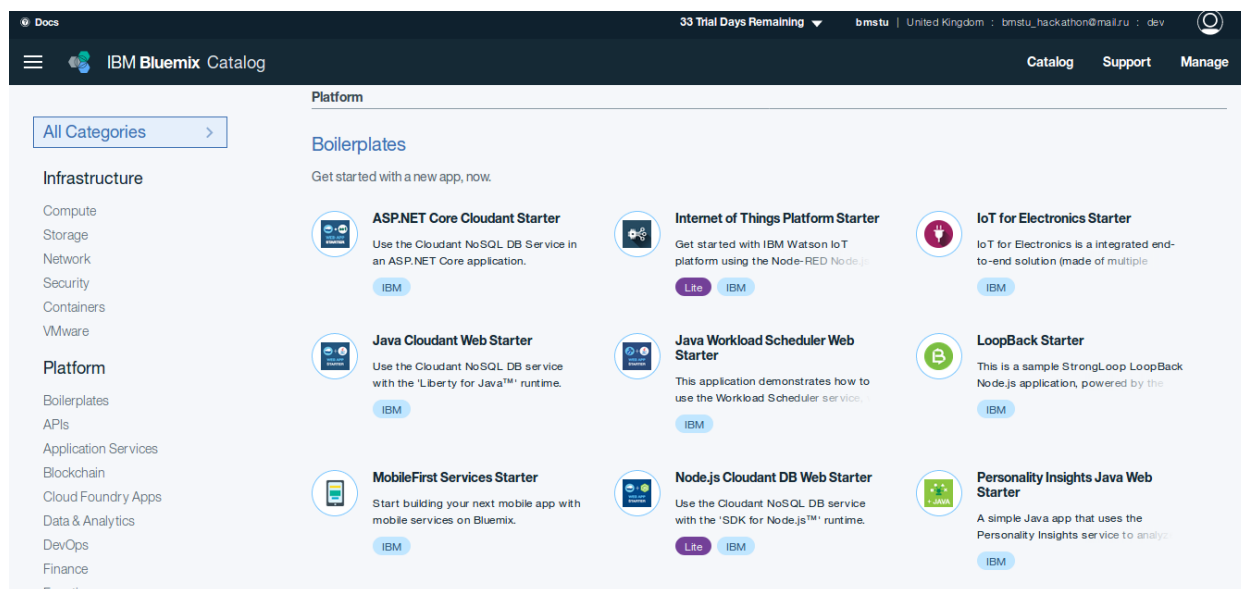
Перейти по ссылке: <https://bluemix.net/>. Пройдите процедуру регистрации, используя Вашу электронную почту (занимает около 5 минут). Далее введите Ваш логин и пароль. Перейдите в раздел DASHBOARD. При необходимости создайте новое рабочее пространство (например, dev).



Перейдите в разделе DASHBOARD ниже и выберите опцию «Take advantage of IoT».

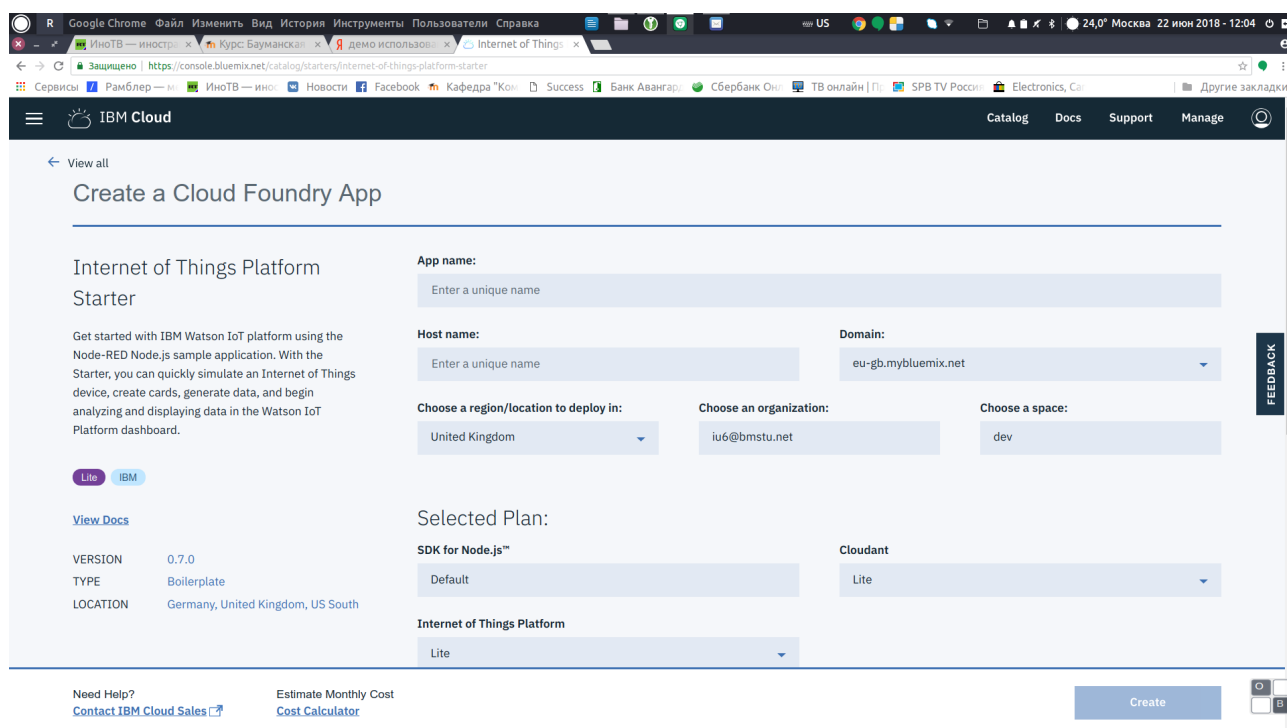


Аналогичное действие можно выполнить позднее из раздела CATALOG, далее в раздел BOILERPLATES и выбрать опцию «Internet of Things Platform Starter».

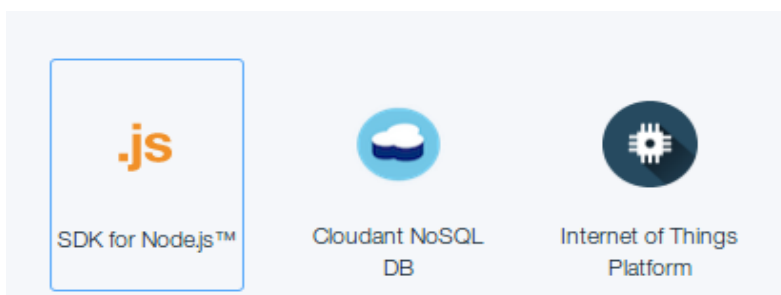


Рабочее поле IBM Cloud.

При создании в поле App укажите имя вашего приложения IoT Platform Starter, что позволит автоматически связать SDK for Node.js с брокером MQTT. Имя должно быть уникальным для всех проектов IBM Cloud.



В результате буду добавлены приложение SDK for Node.js и сервис Cloudbant NoSQL DB и Internet of Things Platform.



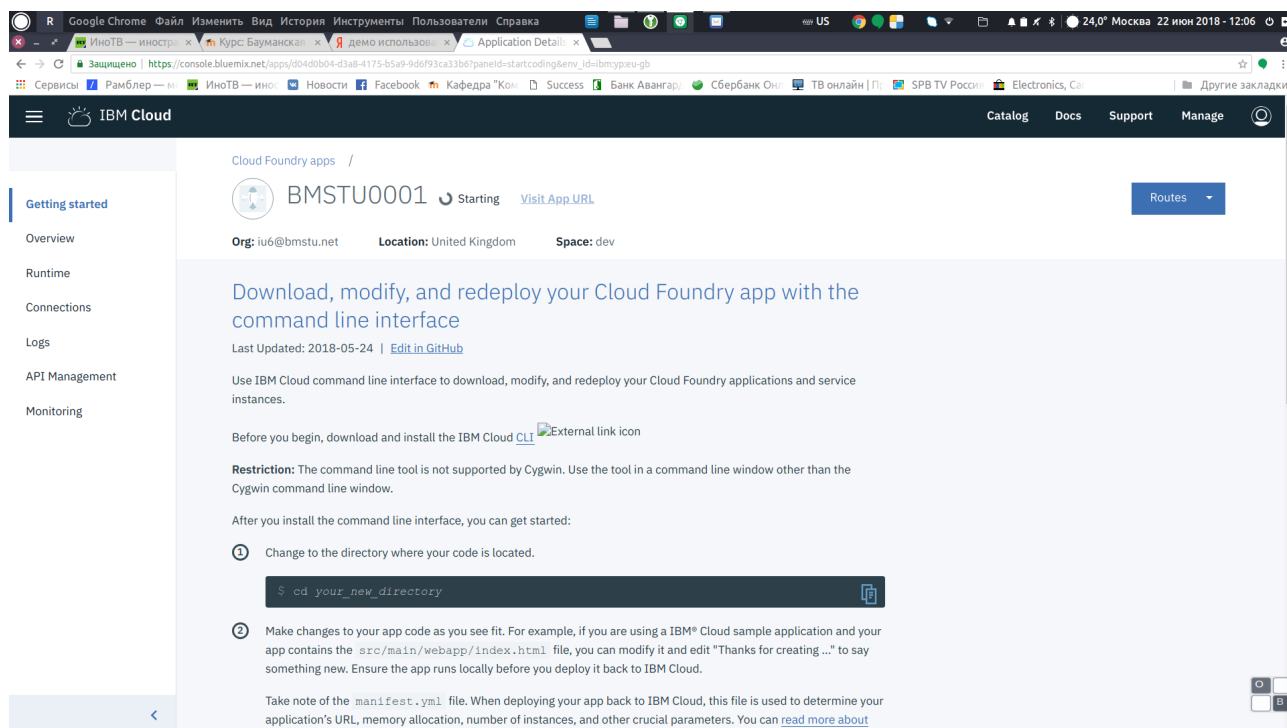
SDK Node.js - среда разработки Java Script приложений на стороне сервера.

Cloudbant NoSQL DB - база данных для хранения данных IoT и настроек проекта.

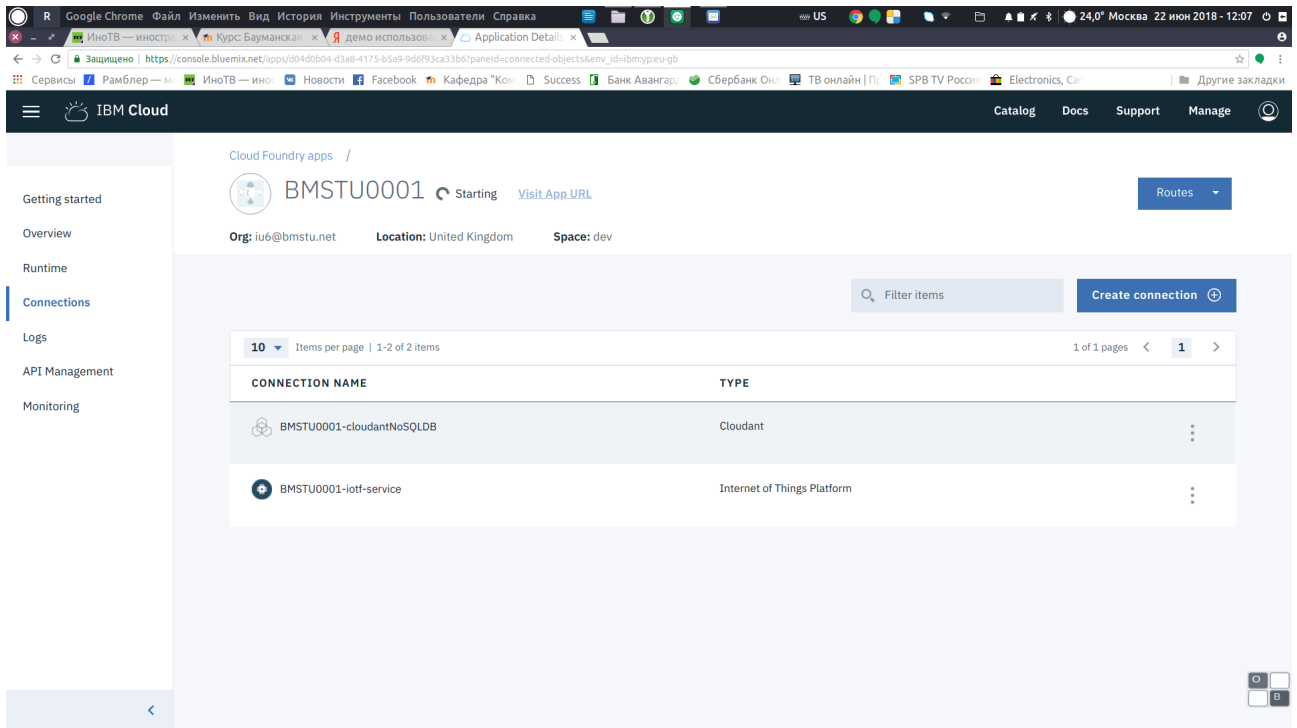
Internet of Things Platform — брокер запросом по протоколу MQTT.

Сервис IoT Foundation является интегрированным компонентом на основе брокера MQTT запросов. Для его использования необходимо выполнить конфигурацию брокера на прием пакетов от RaspberriPi. Создадим описание нового устройства. В результате ваш проект будет иметь следующий вид.

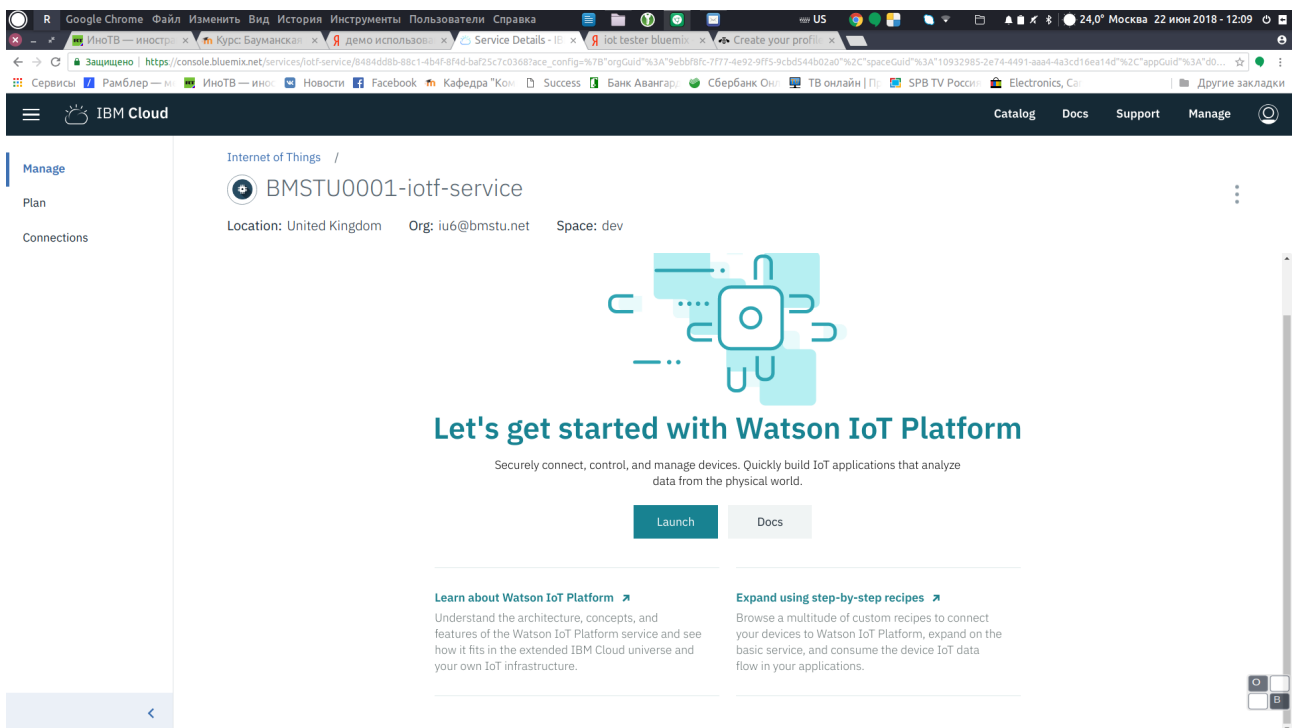
Далее приступим к настройкам полученной сборки.



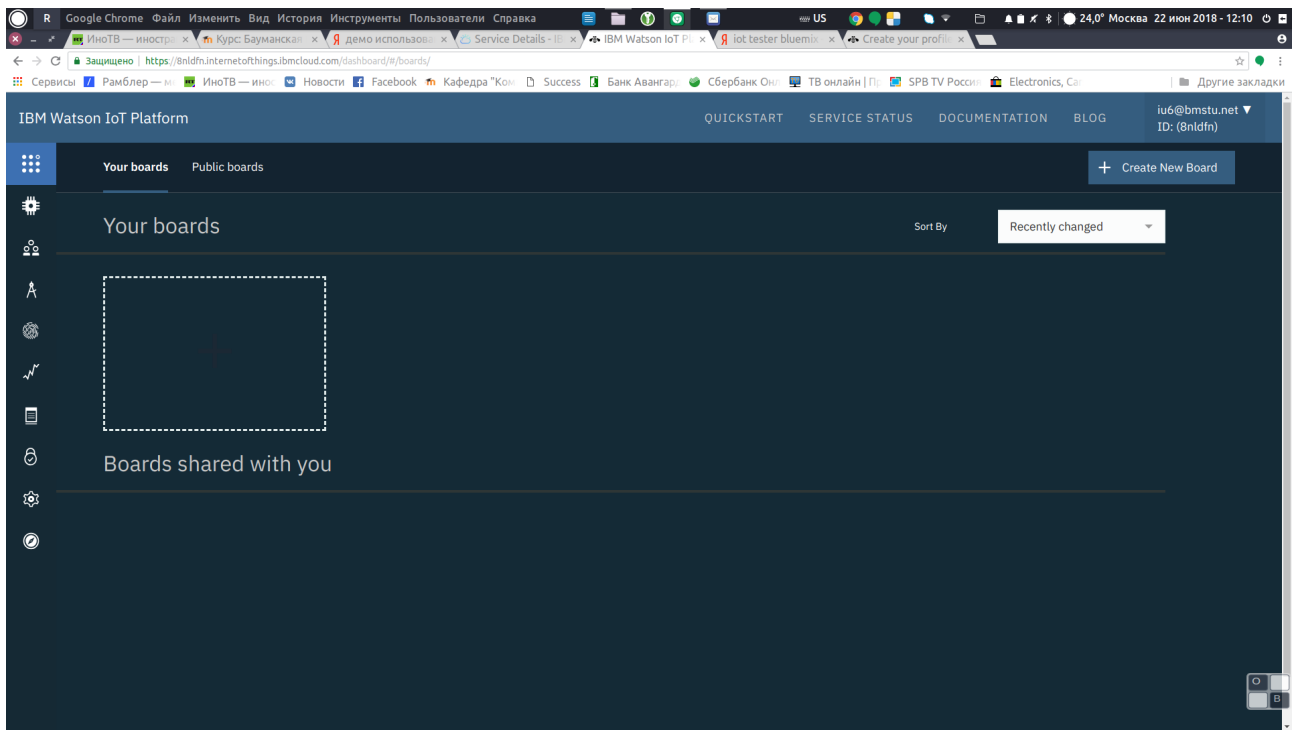
На вкладке Connections проверьте, что компоненты связаны друг с другом:



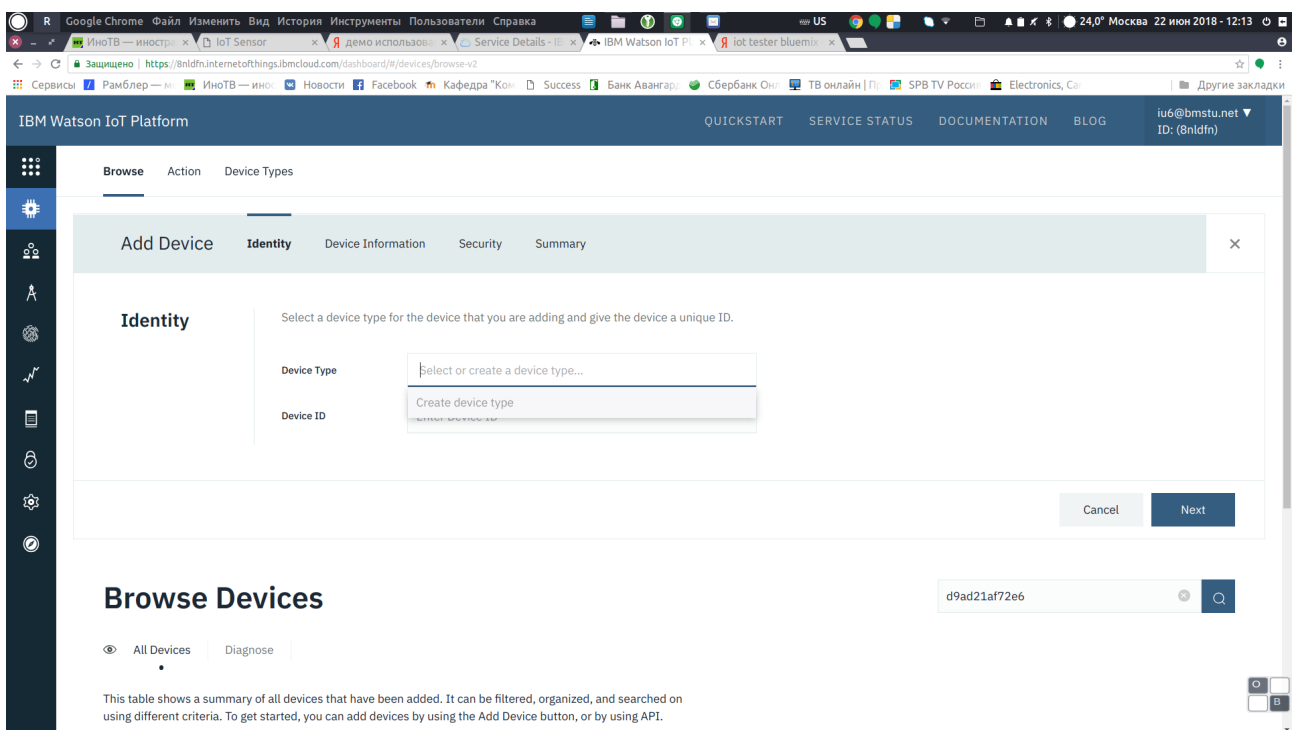
Откройте настройку компонента Internet of Things Platform.



Запустите приложение кнопкой Launch



Далее необходимо создать новое устройство на вкладке Devices.



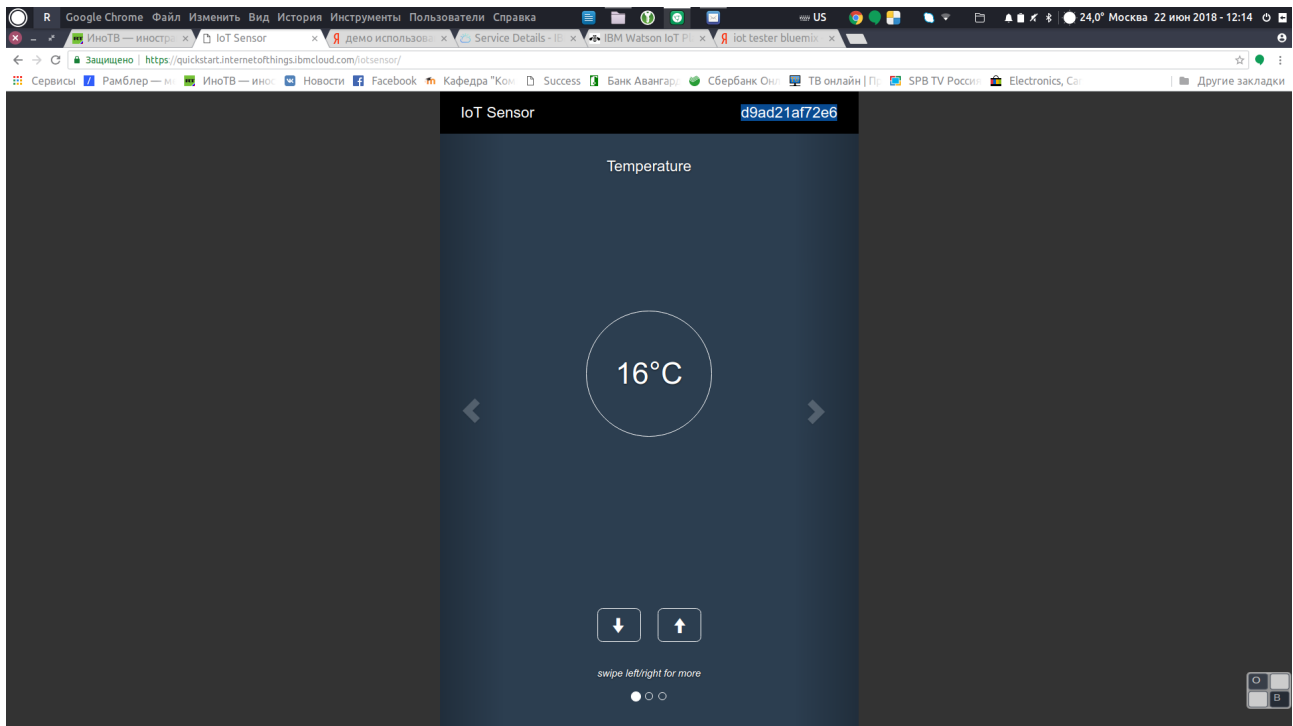
Укажите тип устройства (например, RPI). Далее нужно указать идентификатор устройства IoT. В нашем практикуме мы будем использовать симулятор устройства IoT Sensor. Для этого необходимо перейти на страницу симулятора по адресу:

<https://quickstart.internetofthings.ibmcloud.com/iotsensor/>

IoT Sensor генерирует следующие показатели:

- температуру окружающей среды

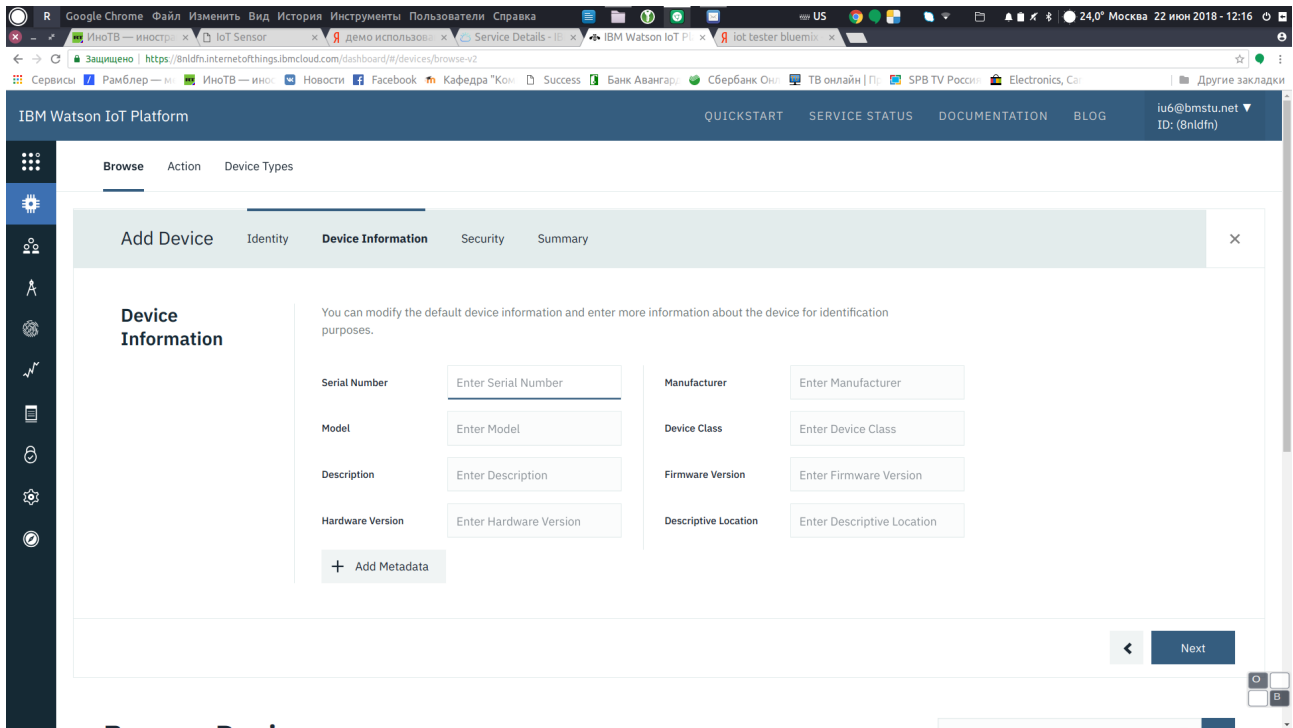
- влажность воздуха
- температуру устройства



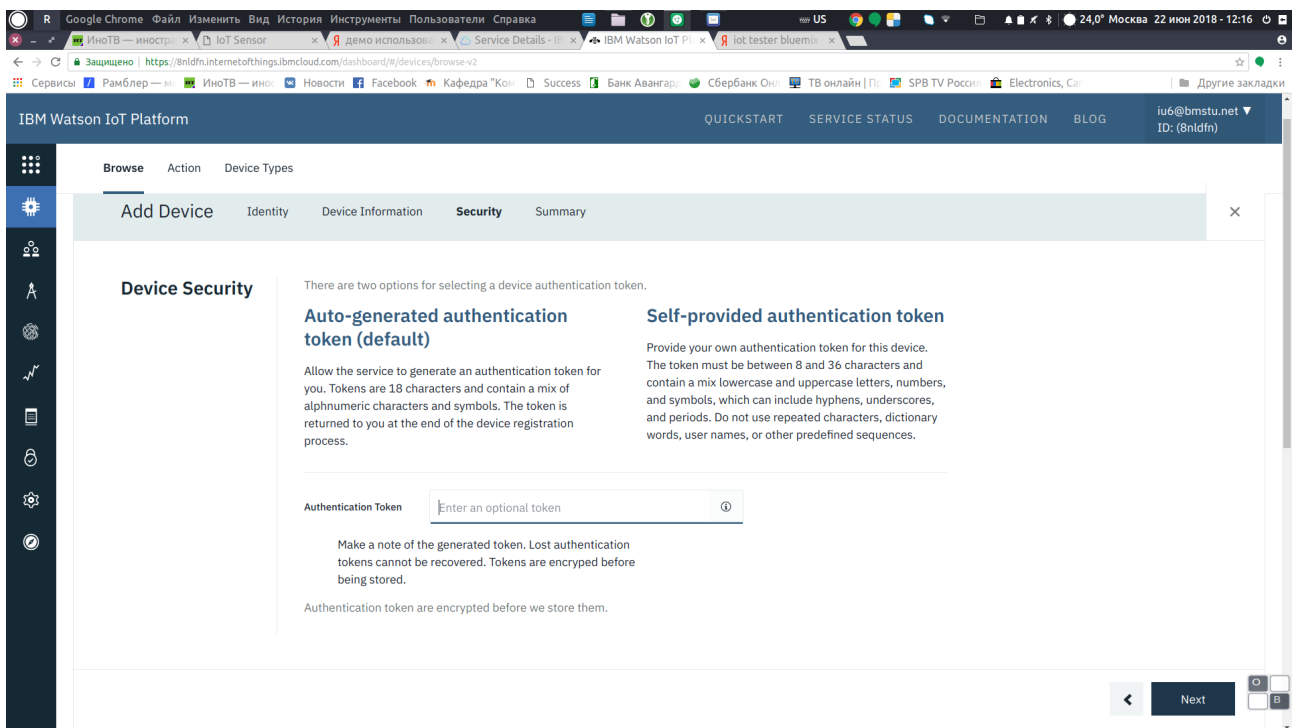
Симулятор предназначен для отладки простейших механизмов передачи телеметрической информации в облачную платформу по протоколу MQTT. Для идентификации потока данных используется автоматически генерируемый MAC адрес (указан в верхнем правом углу, например d9ad21af72e6).

Далее укажите MAC адрес в настройках Device ID компонента Internet of Things Platform.

Остальные поля можно оставить пустыми.



На странице Device Security не вводите информацию.






Откройте диалог Quickstart в правом верхнем углу и в верхнем поле снова введите MAC адрес, полученный в IoT Sensor.

Quickstart

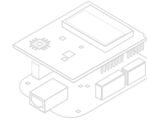
No sign-up required to see how easy it is to connect your device to Watson IoT Platform and view live sensor data.

I accept IBM's Terms of Use

-  Get your device (or simulate one)
-  Follow a recipe to get it connected
-  View live data from your device

Got a physical device?

We have a partner program for IoT along with a set of verified instructions, or 'recipes', for connecting devices, sensors, and gateways.



[VIEW RECIPES](#)

Don't have a device?

You don't need to have a physical device to see Quickstart in action. Try it out by using our simulator.



[FIND OUT MORE](#)

ЭТАП 2. Создание тестового приложения в среде NodeRED.

Перейдите по ссылке <https://xxxxxxx.eu-gb.myIBMCloud.net/ui>, где xxxxxxxx — название вашего проекта (например, bmstu0001). Вы также можете использовать линк «Visit App URL».

The screenshot displays the IBM Cloud console interface for a Node-RED application. The application is named "BMSTU0001" and is in a "Running" state. The console provides a summary of the application's runtime metrics and connections.

Metric	Value
Buildpack	SDK for Node.js™
Instances	1 (All instances are running, Health is 100%)
MB Memory per Instance	256
Total MB Allocation	256 (7.75 GB still available)

Connections (3):

- availability-monitoring-auto
- BMSTU0001-cloudantNoSQLDB
- BMSTU0001-iotf-service

Runtime cost:

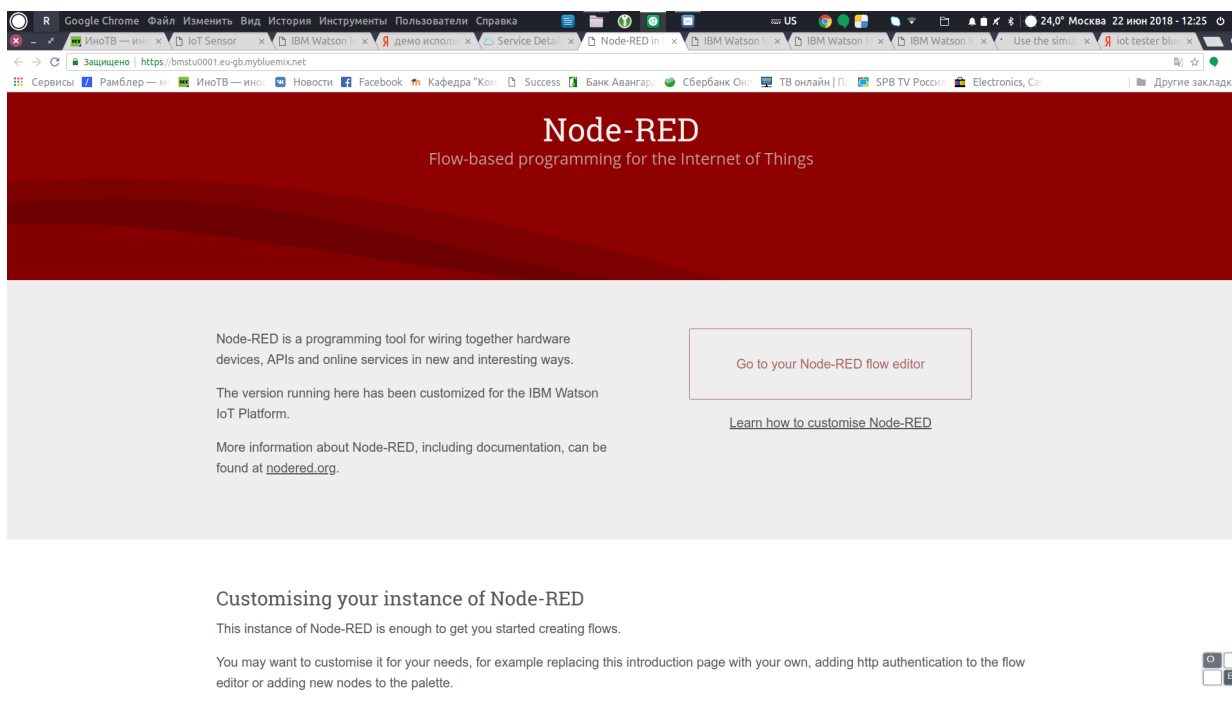
- Current charges for billing period: 0,00 \$
- Estimated total for billing period (2018年7月1日 - 2018年7月31日): 0,00 \$

The screenshot shows the security configuration screen for the Node-RED editor. It offers two options for securing the editor:

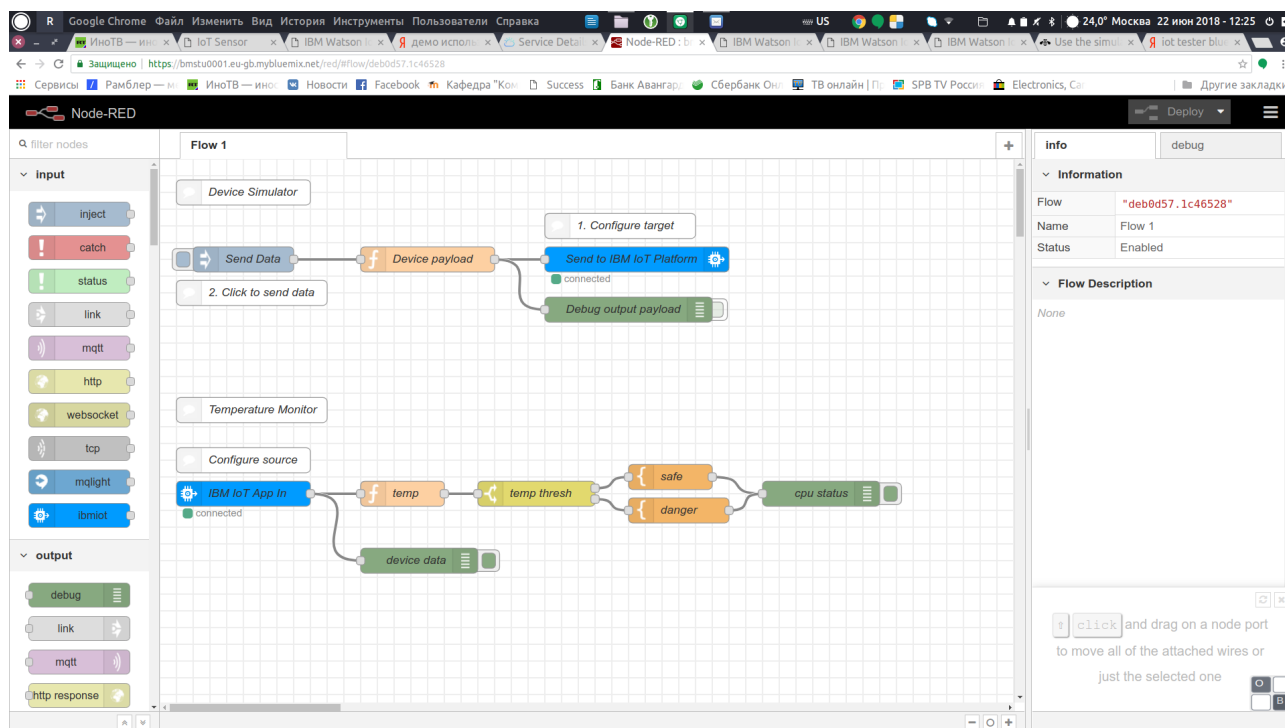
- Secure your editor so only authorised users can access it
- Not recommended: Allow anyone to access the editor and make changes**
Your editor will not be secured. Anyone with the URL will be able to access your flows, data and bound services.
 Tick this box to confirm you want your editor to be insecure

At the bottom, there is a progress indicator and navigation buttons labeled "Previous" and "Next".

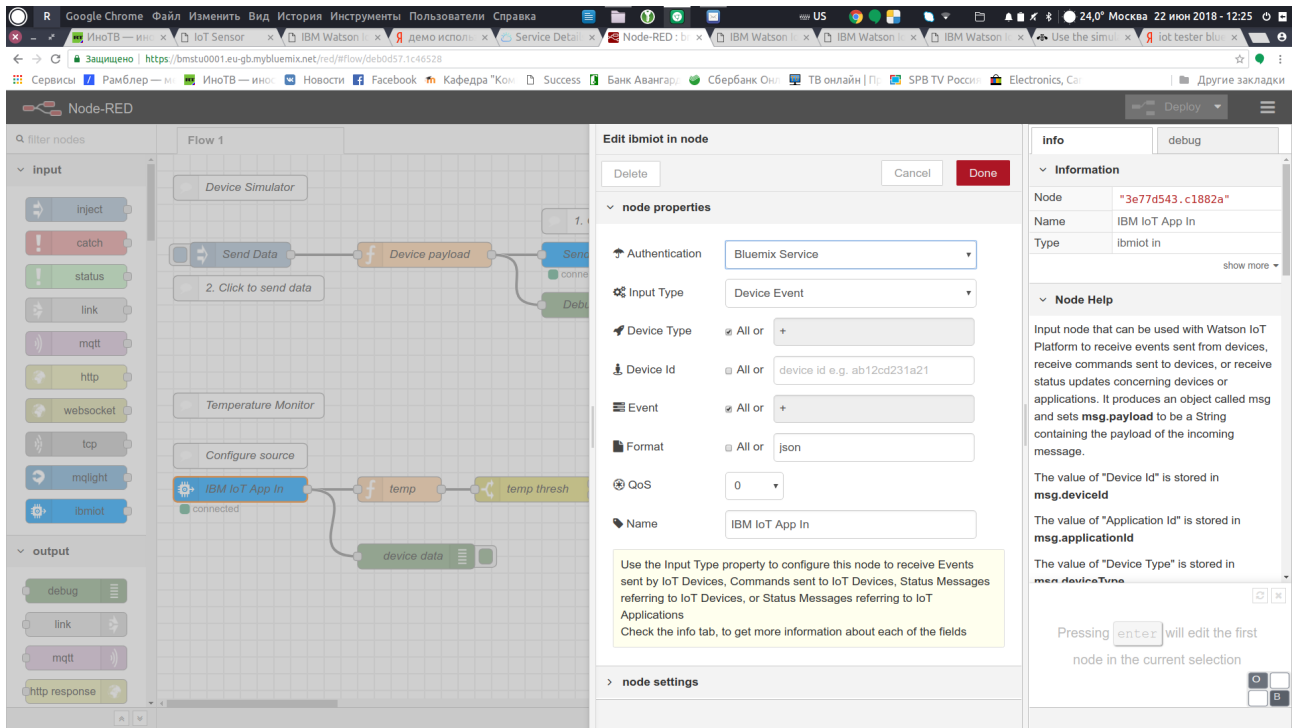
На странице настроек безопасности не включайте безопасный режим (доступ будет осуществляться всеми, у кого имеется ваш MAC номер).



Далее войдите в редактор NodeRED.



При первом запуске приложения вам доступен демо код для тестирования связи с IoT Sensor. Необходимо выполнить его настройку:



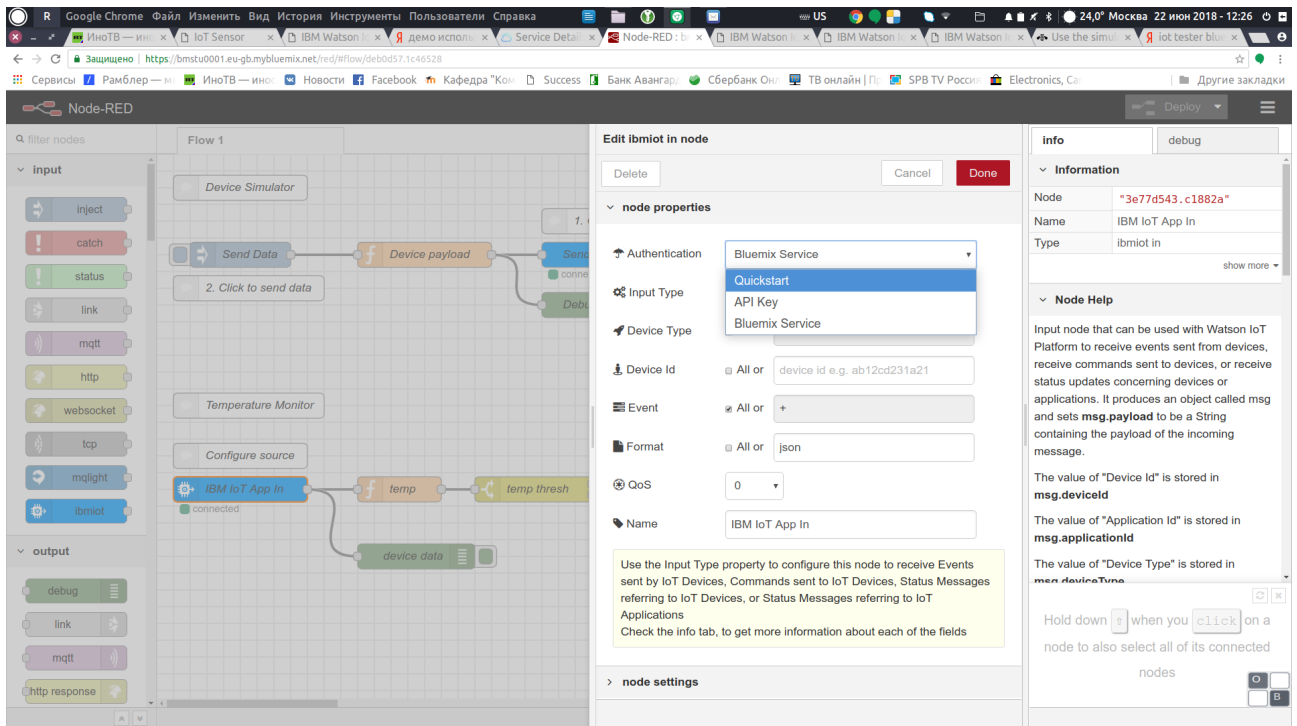
Для нода IBM IoT App In укажите следующие параметры:

Authentication: Quickstart

Input Type: Device Event

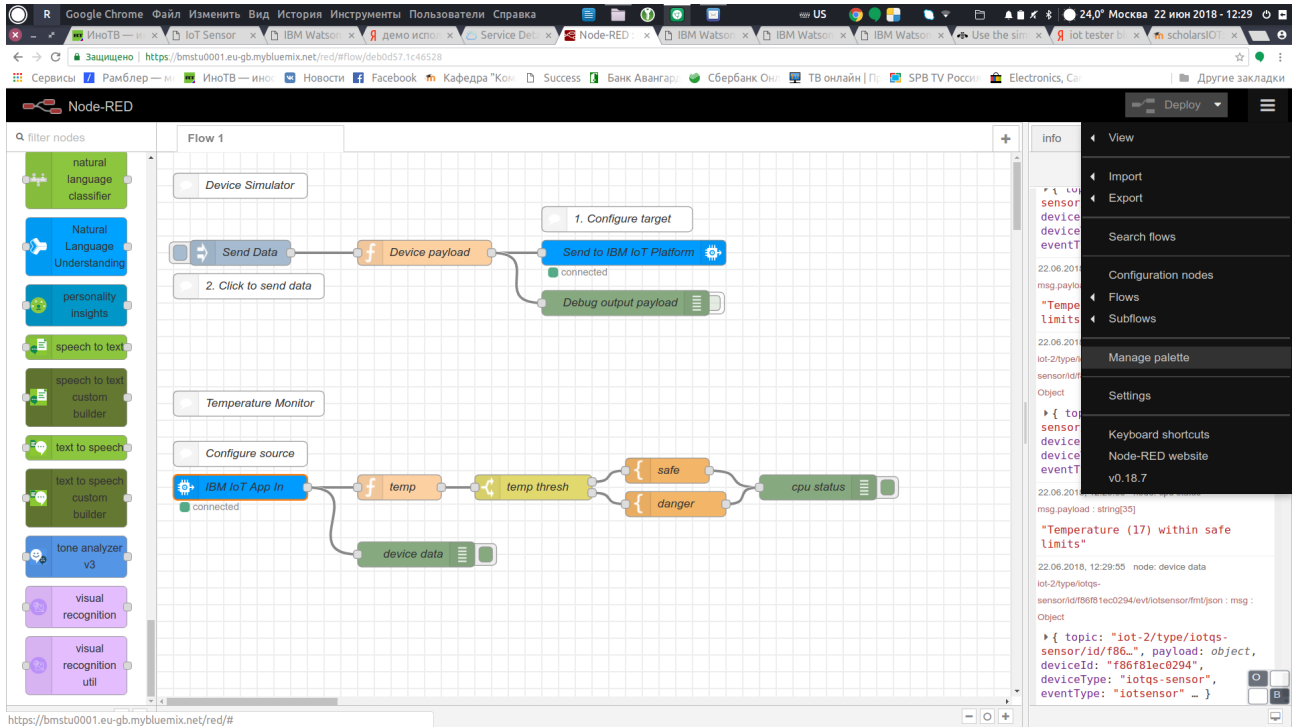
Device Type: RPI

Device ID: Ваш MAC адрес.

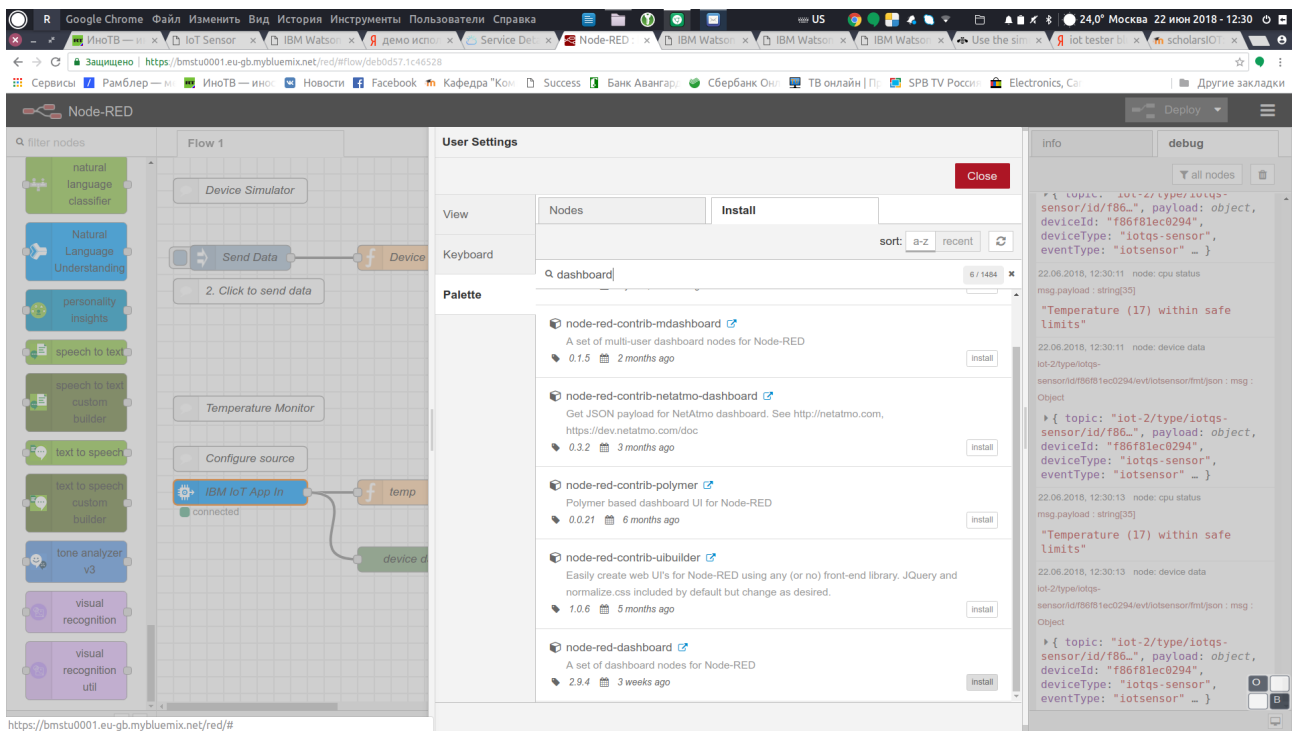


ЭТАП 3. Создание панели управления в среде NodeRED.

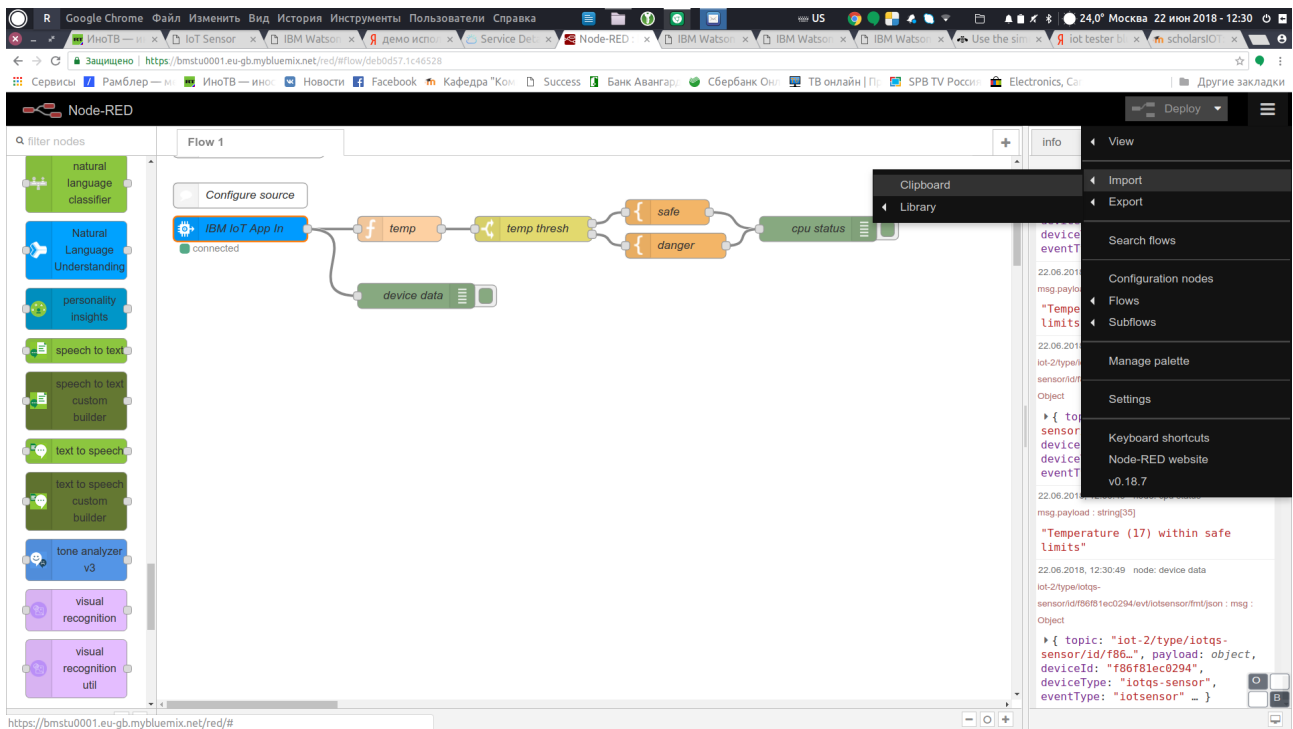
Перейдите в настройки проекта NodeRED в правом верхнем углу. Выберите пункт Manage palette.



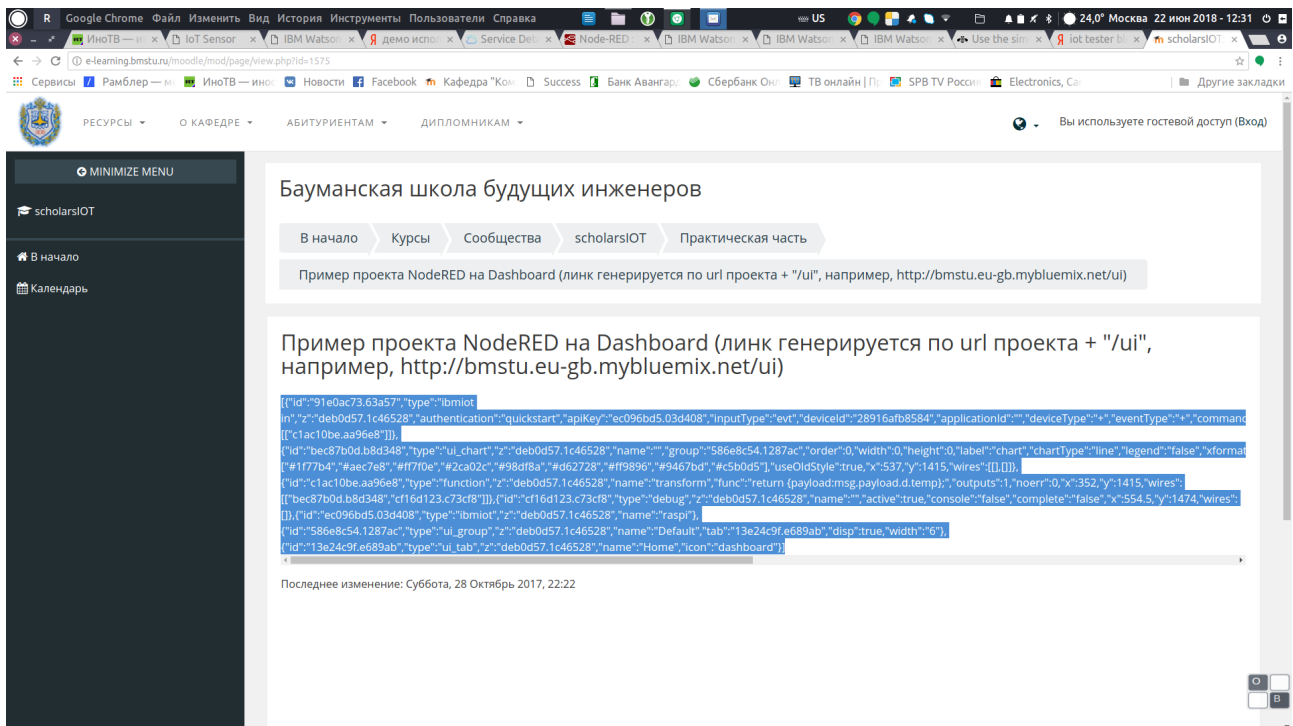
Найдите дополнительный компонент «dashboard».



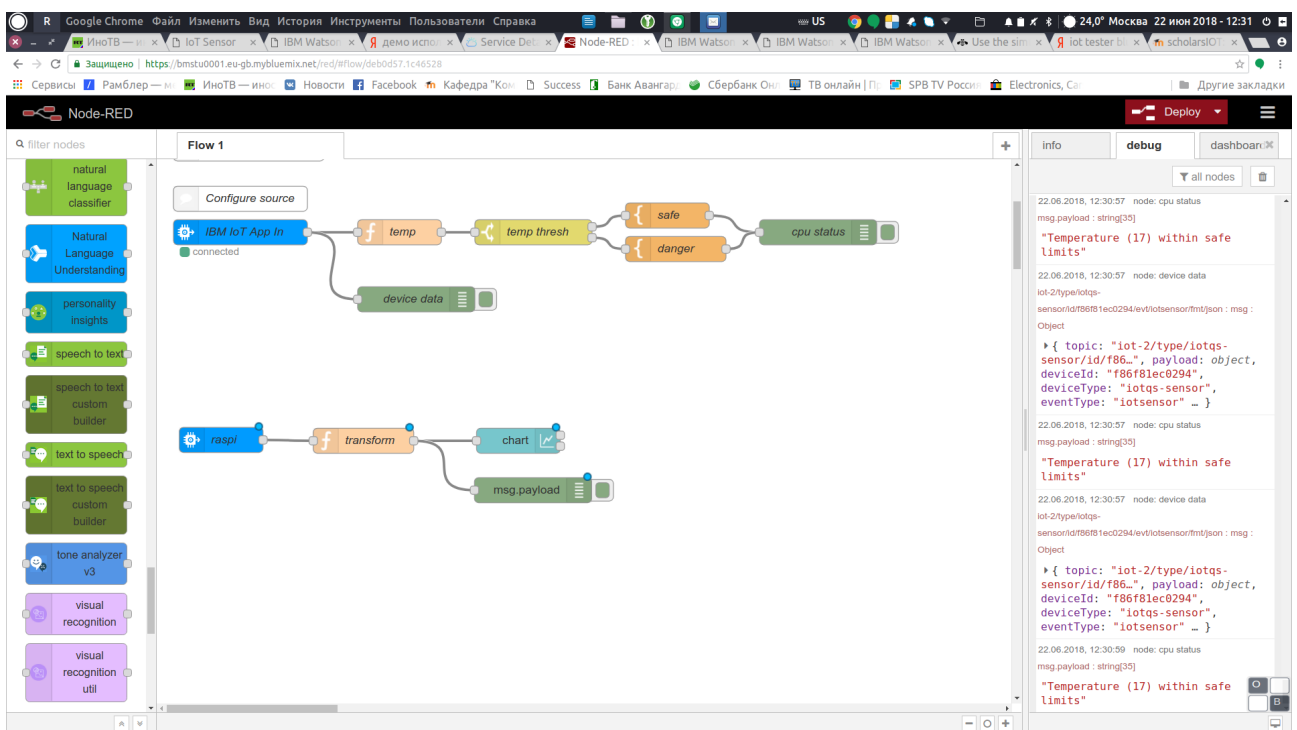
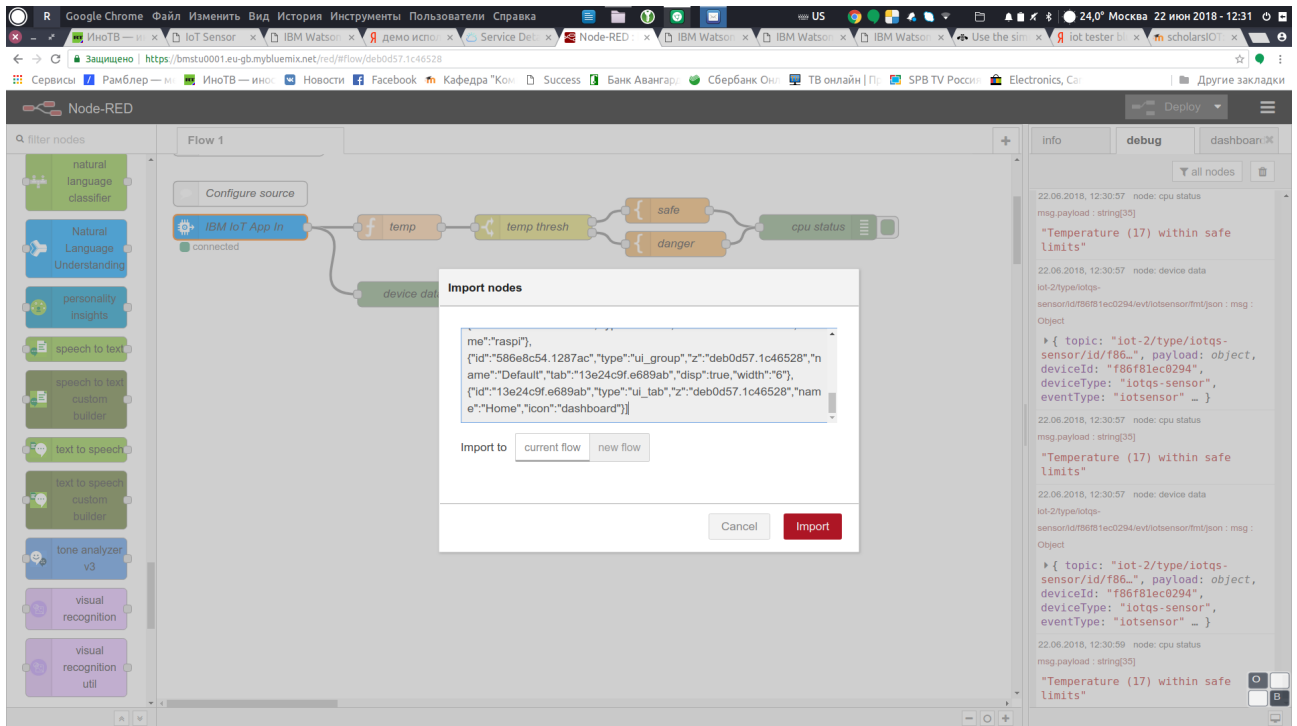
Далее в меню настроек выберите пункт Import и далее пункт Clipboard



Скопируйте код примера со страницы: <http://e-learning.bmstu.ru/moodle/mod/page/view.php?id=1575>



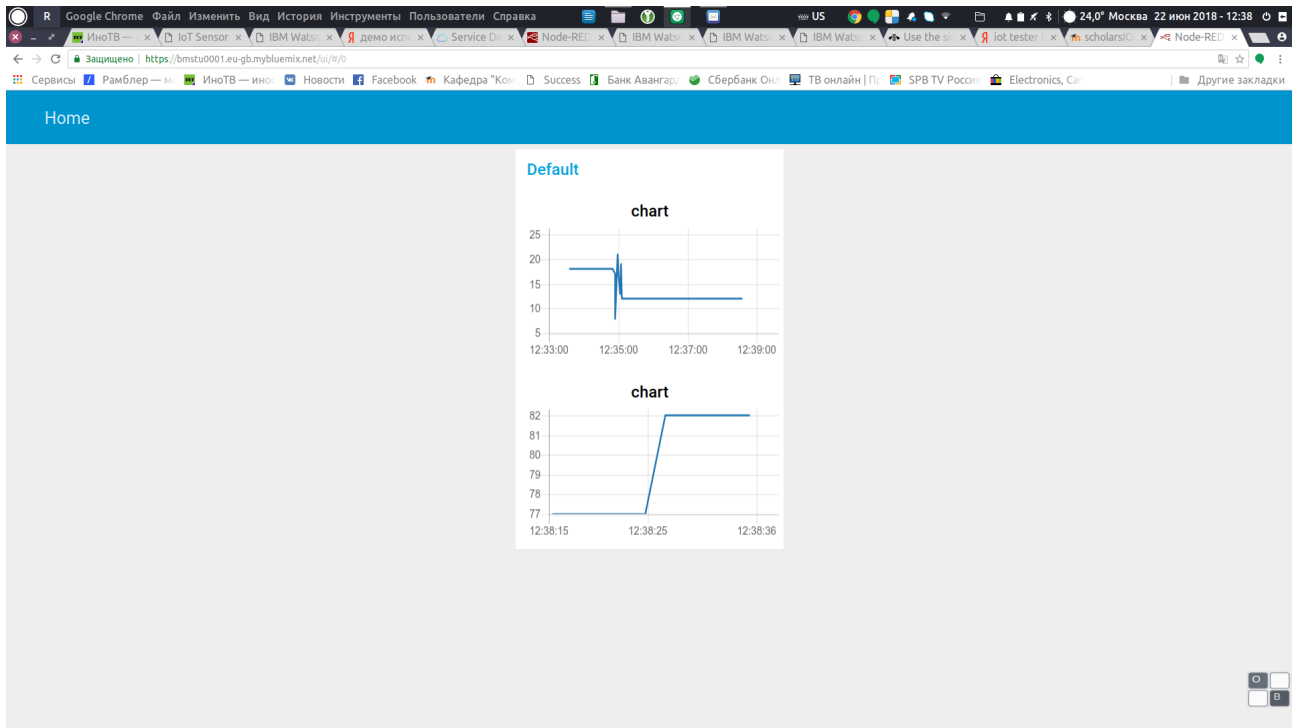
В результате на странице проекта появится дополнительный поток исполнения.



Выполните тестирование полученного приложения.

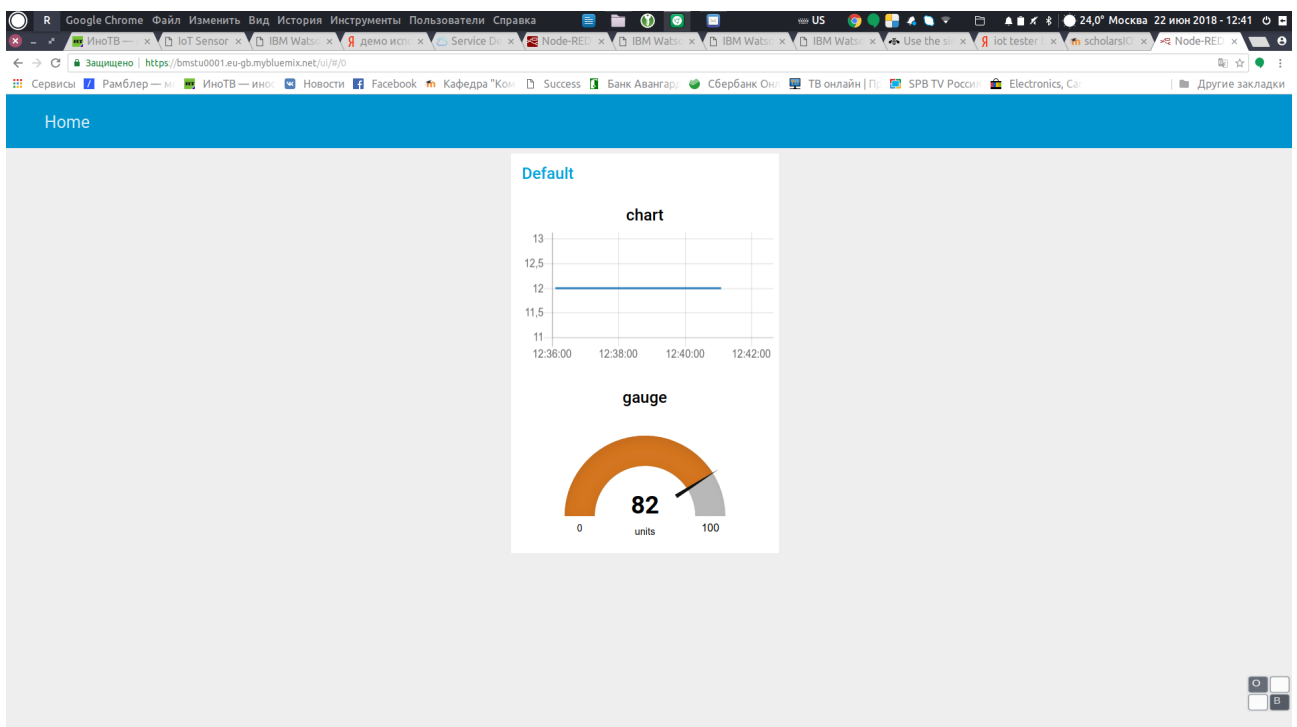
Доступ к панели dashboard осуществляется по url проекта + "/ui", например, <http://bmstu.eu-gb.myIBM Cloud.net/ui>).

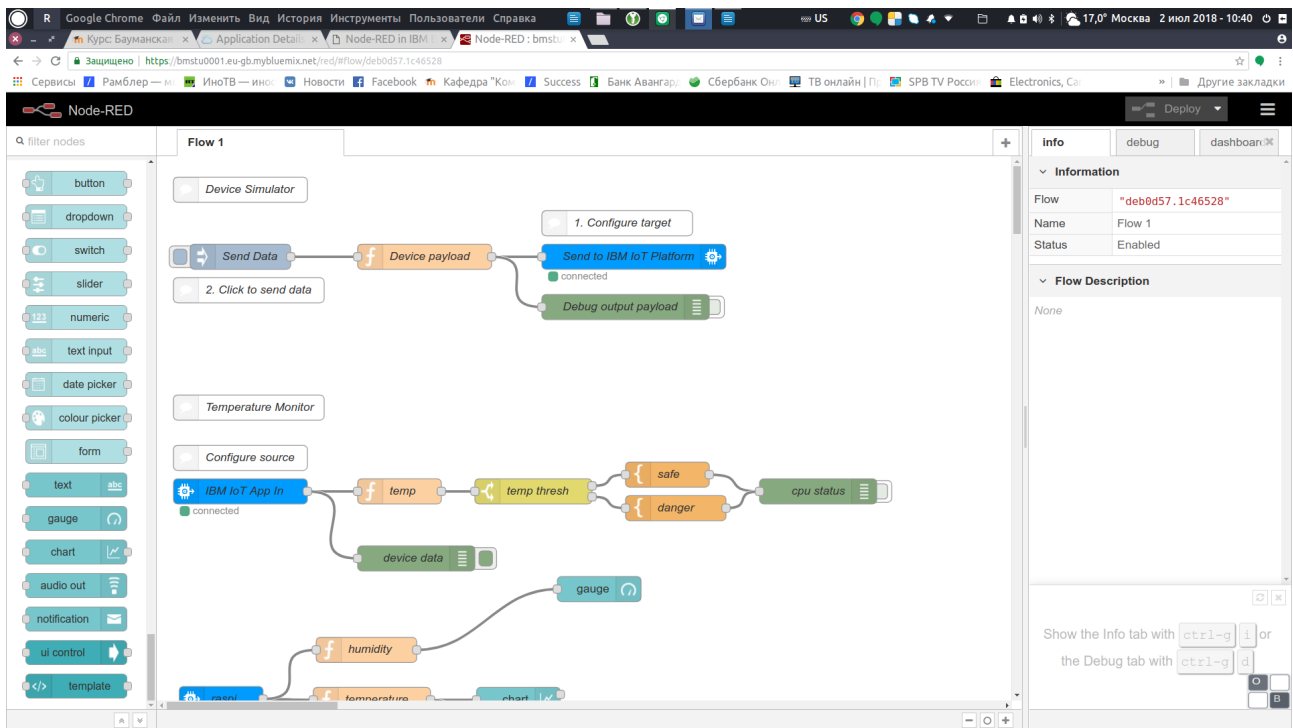
Измените состояние всех сенсоров IoT Sensor и проследите изменение на dashboard.



Индивидуальное задание

Добавьте компоненты dashboard для отображения других параметров: Влажности и температуры устройства. Исследуйте возможности других визуальных компонентов нодов dashboard.





ЭТАП 4. Настройка проекта для приема данных от физического устройства.

В качестве устройства будет использован одноплатный компьютер Raspberry Pi, работающий под управлением операционной системы Raspbian Linux.

Некоторые полезные команды для работы в ОС Raspbian:

"top" — запуск предустановленного в Raspbian диспетчера задач;

"sudo raspi-config" — запуск первоначального меню настроек;

"sudo passwd root" — создание пароля для пользователя root;

"startx" — запуск графической оболочки;

"sudo halt" — выключение RPi;

"logout" — выход из системы;

"sudo reboot" — перезагрузка RPi;

"cd" — переход в необходимую директорию, например, для перехода в директорию /etc/network/ - "cd /etc/network/";

"pwd" — путь до текущей директории;

"dir" — содержимое текущей директории;

"mkdir" — создание директории. Например, "mkdir /home/pitest/" создаст директорию "pitest";

"rmdir" — удаление директории. Например, "rmdir /home/pitest/" - удаление директории "pitest";

"cat" — открыть файл для чтения. Например, "cat /etc/network/interfaces" покажет содержимое файла "interfaces";

"nano" — открыть файл для редактирования. Например, "nano /etc/network/interfaces" откроет для редактирования файл "interfaces";

"ifconfig" — отобразит текущую конфигурацию сети;

"df" — выведет в консоли свободное и используемое дисковое пространство для всех разделов файловой системы;

"clear" — очистить экран терминала;

"Ctrl"+"Ins" — скопировать выделенное (текст);

"Shift"+"Ins" — вставить из буфера (текст);

"sudo" — выполнения команд с правами root пользователя. Например, это актуально, если вы зашли под пользователем "pi" и хотите из консоли отредактировать какой-нибудь системный файл - "sudo nano путь_до_файла";

"Ctrl"+"C" — остановка текущего действия/выход из консольного приложения;

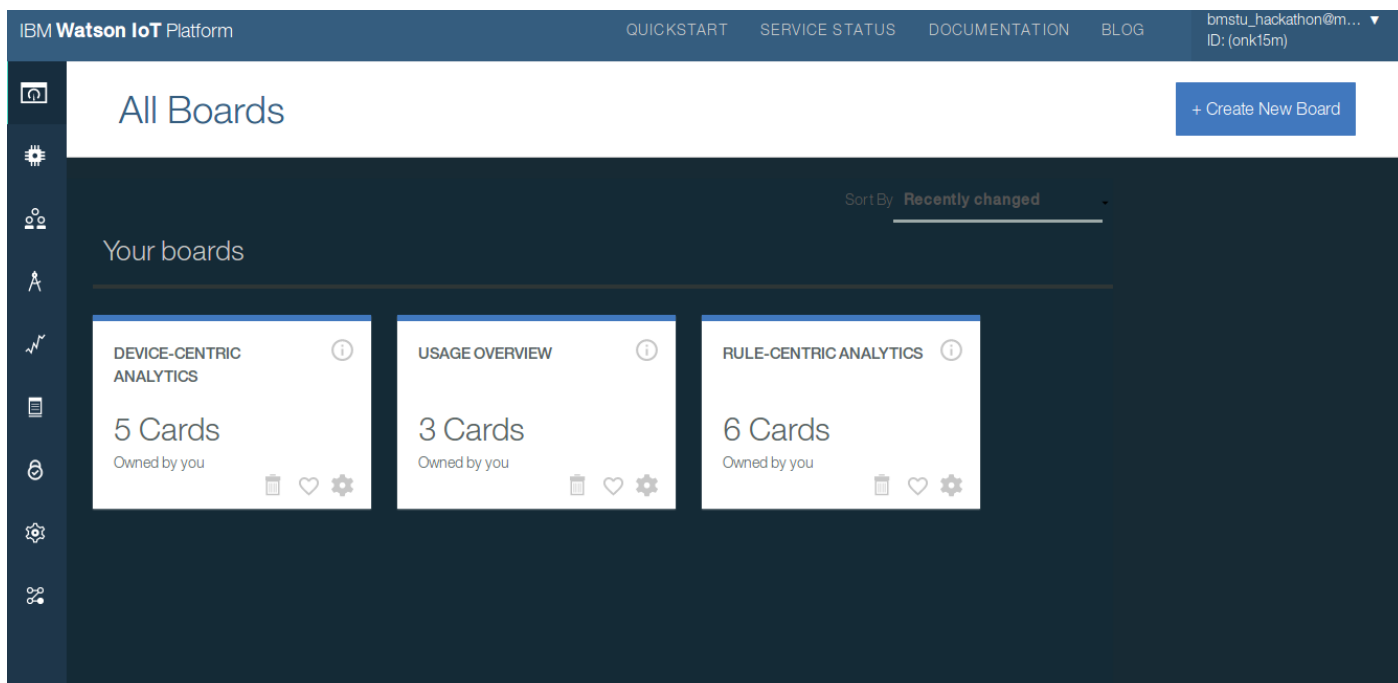
"sudo apt-get update" — обновление списка доступных пакетов;

"sudo apt-get upgrade" — обновление установленных пакетов;

"sudo apt-get install" — установка необходимого пакета. Например, для установки консольного браузера Links вводим "sudo apt-get install links".

Для настройки передачи данных в IBM Cloud необходимо сконфигурировать новое устройство. Для этого перейдите в Dashboard к настройкам сервиса Internet of Things Platform (В списке Services).

Перейдите на вкладку Device. Далее добавьте новое устройство (Add Device).



Создайте новый тип устройства Device Type (например, с именем: RPi). Все поля за исключением поля Name можно оставить пустыми.

Для добавления устройства созданного Вами типа понадобится определить уникальный DeviceID устройства (например, его mac адрес вашей платы RPi).

Add Device

Device Info

Device ID is the only required information, however other fields are populated according to the attributes set in the selected device type. These values can be overridden, and attributes not set in the device type can be added.

Device ID	<input type="text" value="b827eb79051a"/>
-----------	---

В следующем окне Вы можете выбрать один из двух вариантов: использовать автоматически сгенерированный токен или добавить существующий токен устройства. Выберем первый вариант (оставьте поле token пустым). В результате Вам будет выдана информация о настройках устройства, которые необходимо сохранить для последующего использования (см. следующее задание).

Organization ID	gavygq
Device Type	RPi
Device ID	b827eb79051a
Authentication Method	token
Authentication Token	Wzfs02uxa)KNM&&G3X

Полезной функцией IoT Platform консоли является возможность просмотра логов сообщений MQTT. Данная возможность может быть использована для проверки работоспособности всей инфраструктуры, поиска ошибок и пр.

Для связи с IBM Cloud логично использовать протоколы, основанные на TCP или UDP. В данном примере используется MQTT. IBM Cloud предоставляет библиотеку `ibmotf` — небольшую обёртку над MQTT, инициализация соединения с которой выглядит как:

После регистрации устройства в IBM Cloud мы получаем данные для авторизации, которые поместим в файл [device.cfg](#):

```
[device]
org=md8qpm
type=bmstu001
id=b827eb79051a
auth-method=token
auth-token=3BZu_drqRuAKH*K*+i
```

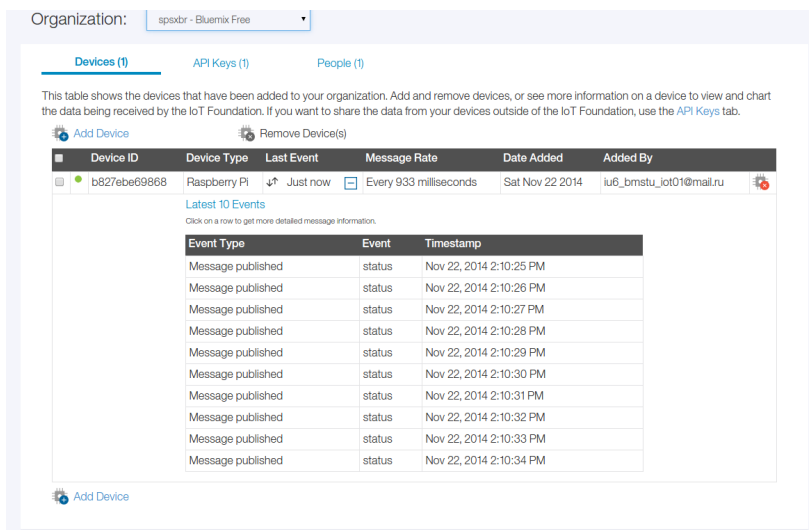
Внесите изменения в файл `/etc/iotsample-raspberrypi/device.cfg`.

Запустите процесс `iot`.


```
$sudo service iot start
```

Проверьте работоспособность устройства Rpi на вкладке Devices.

Если наблюдается регулярное поступление сообщений от Rpi, конфигурация выполнена успешно.

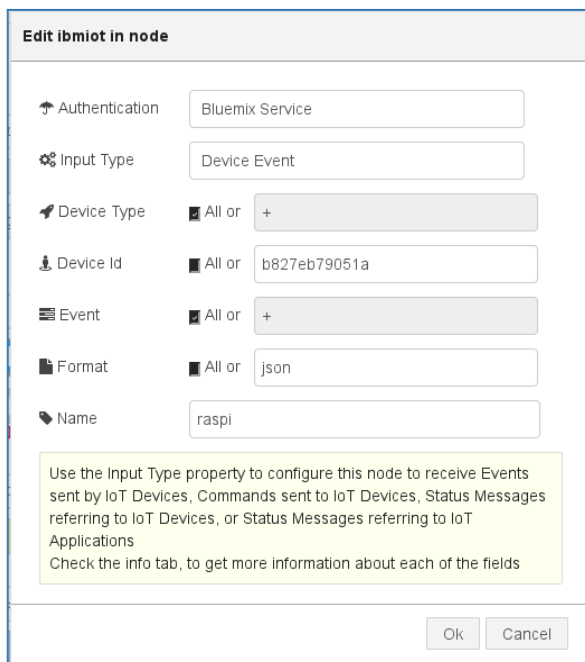


ЭТАП 5. Разработка приложения NodRED для приема данных от Rpi

Измените настройки блока IBM IoT App In следующим образом:

Authentication: IBM Cloud Service

Device ID: mac адрес вашего устройства.



Запустите приложение и определите на вкладке debug, поступают ли данные в приложение.

The screenshot shows the Node-RED interface with a flow named 'Flow 1'. The flow starts with a 'Device Simulator' node, followed by a 'Send Data' node, a 'Device payload' node, and a 'Send to IBM IoT Platform' node. A 'Debug output payload' node is connected to the 'Send to IBM IoT Platform' node. The 'Send to IBM IoT Platform' node is connected to an 'IBM IoT App In' node. The 'IBM IoT App In' node is connected to a 'temp' node, which is connected to a 'temp thresh' node. The 'temp thresh' node is connected to two nodes: 'safe' and 'danger'. The 'safe' and 'danger' nodes are connected to a 'cpu status' node. The 'cpu status' node is connected to a 'device data' node. The debug console shows the following messages:

```
iot-2/type/iotqs-sensor/id/28916afb8584/evt/iotqsensor
//msg/json : msg : Object
> { topic: "iot-2/type/iotqs-sensor/id/289...", payload: object, deviceId: "28916afb8584", deviceType: "iotqs-sensor", eventType: "iotsensor" ... }
28.10.2017, 13:44:46 node: device data
iot-2/type/iotqs-sensor/id/28916afb8584/evt/iotqsensor
//msg/json : msg : Object
> { topic: "iot-2/type/iotqs-sensor/id/289...", payload: object, deviceId: "28916afb8584", deviceType: "iotqs-sensor", eventType: "iotsensor" ... }
28.10.2017, 13:44:48 node: device data
iot-2/type/iotqs-sensor/id/28916afb8584/evt/iotqsensor
//msg/json : msg : Object
> { topic: "iot-2/type/iotqs-sensor/id/289...", payload: object, deviceId: "28916afb8584", deviceType: "iotqs-sensor", eventType: "iotsensor" ... }
28.10.2017, 13:44:50 node: device data
iot-2/type/iotqs-sensor/id/28916afb8584/evt/iotqsensor
//msg/json : msg : Object
> { topic: "iot-2/type/iotqs-sensor/id/289...", payload: object, deviceId: "28916afb8584", deviceType: "iotqs-sensor", eventType: "iotsensor" ... }
```

В случае возникновения проблем, проверьте получение данных непосредственно в Internet of Things Platform на вкладке Devices.

The screenshot shows the IBM Watson IoT Platform 'Devices' page. The page has a navigation bar with 'QUICKSTART', 'SERVICE STATUS', 'DOCUMENTATION', and 'BLOG'. The user is logged in as 'bmstu_hackathon@m... ID: (onk15m)'. The main content area has a 'Devices' header and a 'Browse' button. Below the header, there are tabs for 'Diagnose', 'Action', 'Device Types', and 'Manage Schemas'. A table displays the following data:

Device ID	Device Type	Class ID	Date Added	Location
28916afb8584	PI	Device	28 окт. 2017 г. 13:48:26	

Выполните тестирование полученного приложения с помощью компонентов dashboard.

