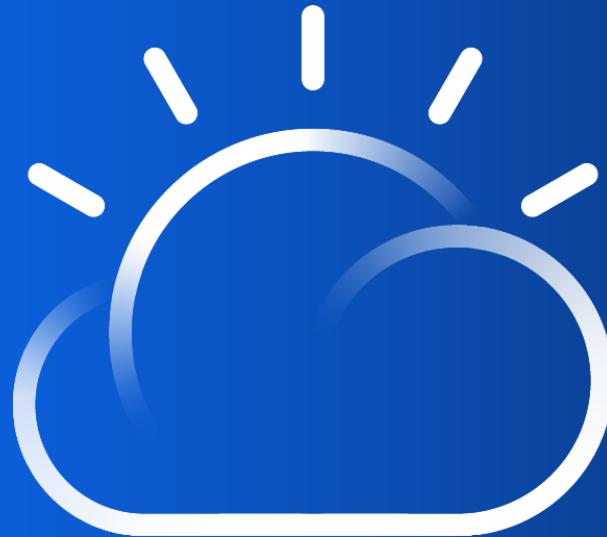


# Controlando dispositivos IoT remotamente através de linguagem natural



—  
Eduardo Petecof Mattoso  
João Pedro Poloni Ponce

CodeBakery Developers



# Conteúdo

<b>Introdução</b>	<b>01</b>	<b>Importe o Flow</b>	<b>49</b>
Internet das coisas	03	Configurando a conexão MQTT	51
Interagindo em linguagem natural	07	Verificando a conexão	57
Computação em nuvem	09		
Arquitetura proposta	14	<b>Enviando comandos com Cloud Functions</b>	<b>63</b>
		Criando as funções	63
	<b>16</b>	Testando a primeira função	72
<b>Watson IoT Platform</b>	17	Criando a segunda função	77
Criando o serviço	20	Obtendo as credenciais de uso	84
Criando um Device Type	23		
Criando um Device	29	<b>Criando o Assistente Virtual</b>	<b>88</b>
Configurando o tipo de conexão	32	Criando uma instância do Watson Assistant	88
Ativando o Cache de eventos		Fazendo o upload do Workspace	92
	<b>33</b>	Credenciais da sua Cloud Function	94
<b>Simulando um Device com o Node-RED</b>		Testando o Assistente Virtual	103
Criando uma instância do Node-RED	36		
Configurando o seu Node-RED	44		
Instale o Dashboard	49	<b>Conclusão</b>	<b>104</b>

Internet das coisas:

# Digitalização do mundo físico.



# Crie a representação virtual de um ativo real.

Monitoramento contínuo do estado desse ativo através de sensores.

Controle remoto do ativo submetendo comandos para ele em qualquer lugar.

Armazenamento de dados de utilização para aprendizado contínuo.

**Integração com ferramentas cognitivas: Interaja com o usuário em linguagem natural.**

“The Digital Twin is the virtual, statefull representation of a physical object or system across its life-cycle (design, build, operate) using operational real-time data and other sources to enable understanding, learning, reasoning, and dynamic recalibration for improved decision making.”

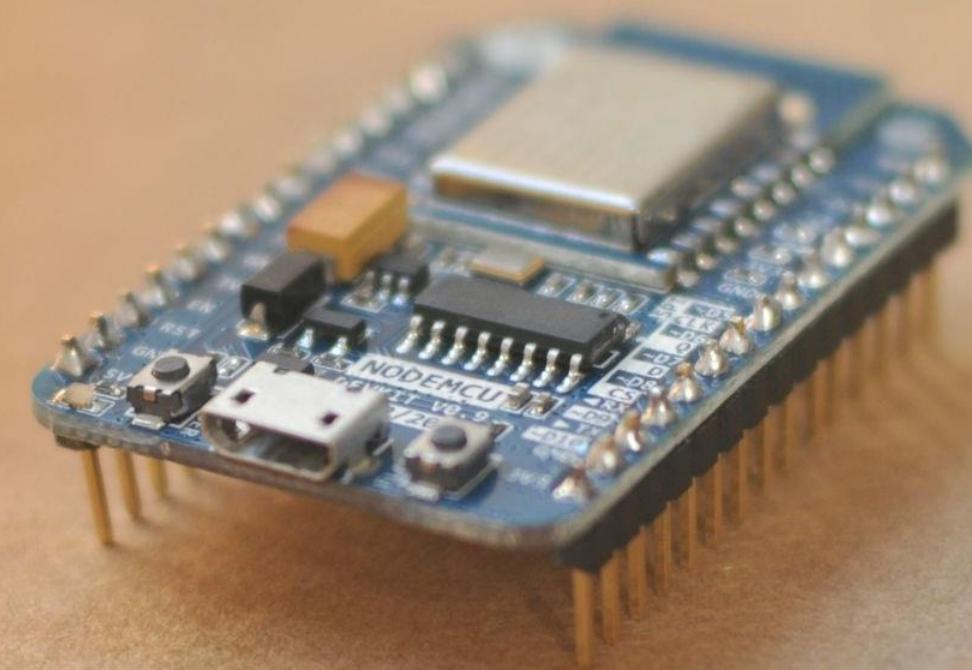
## **Andrew Foster**

Senior Offering Manager - Watson IoT Platform

## NodeMCU

“The Development Kit based on ESP8266, integrates GPIO, PWM, IIC, 1-Wire and ADC all in one board. Power your development in the fastest way combining with NodeMcu Firmware!”

<https://nodemcu.com>



## Omega2

“WiFi-enabled devices ideal for rapid IoT development. Learn, prototype to validate your idea, and go to production, all with one platform.”

<https://onion.io/>



Assistentes Virtuais:

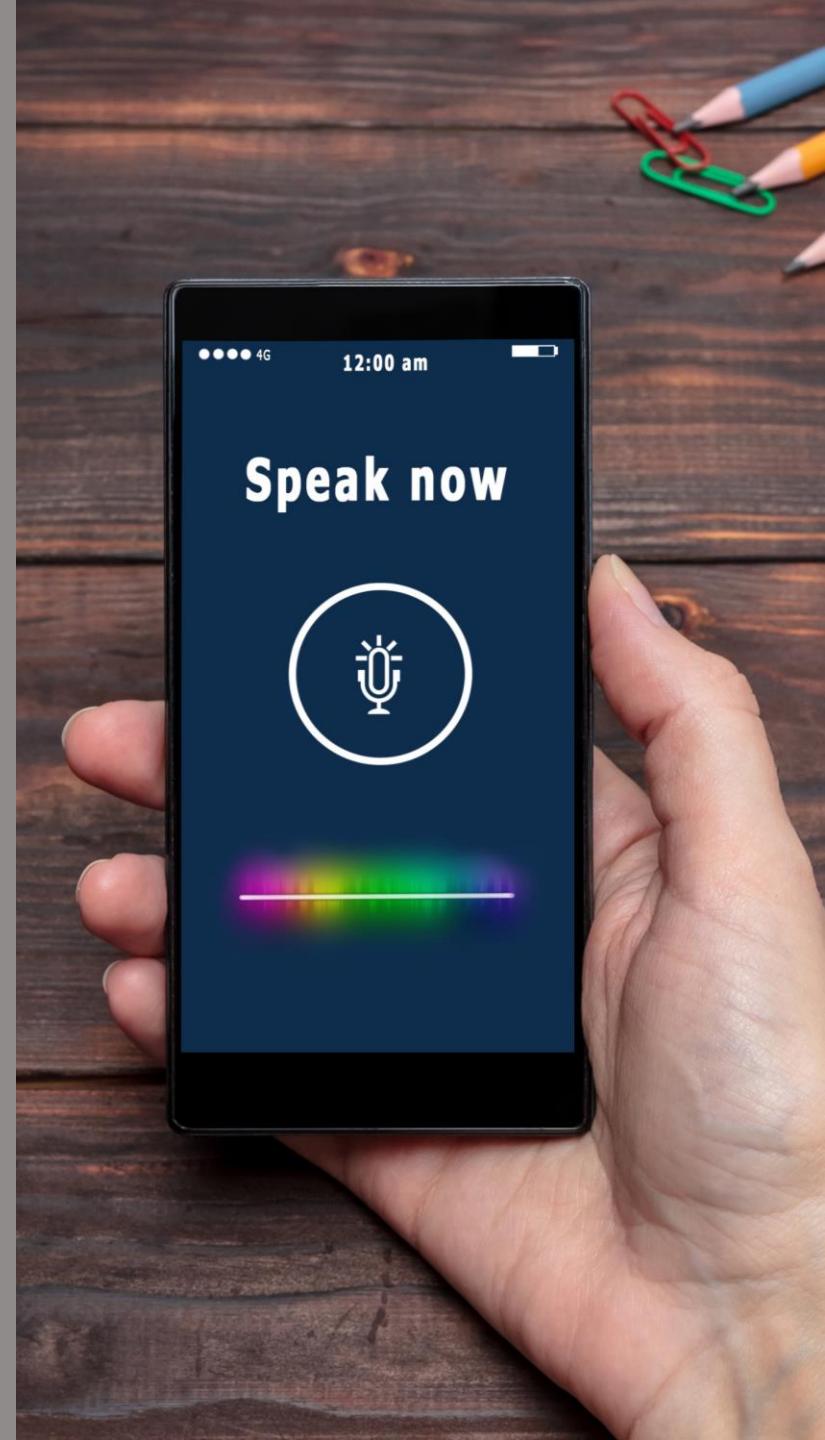
# Interagindo em linguagem natural.



A linguagem natural permite que o usuário vá **direto ao ponto** que o interessa, sem precisar de navegações.

Um diálogo permite a extração de várias informações sobre o usuário. **Muito além de um simples apertar de botão.**

Também é possível o uso da voz e do som, os quais permitem que o usuário **mantenha livre a sua visão e as suas mãos.**



Computação em nuvem:

**Escalável e  
sob demanda.  
24h por dia.**

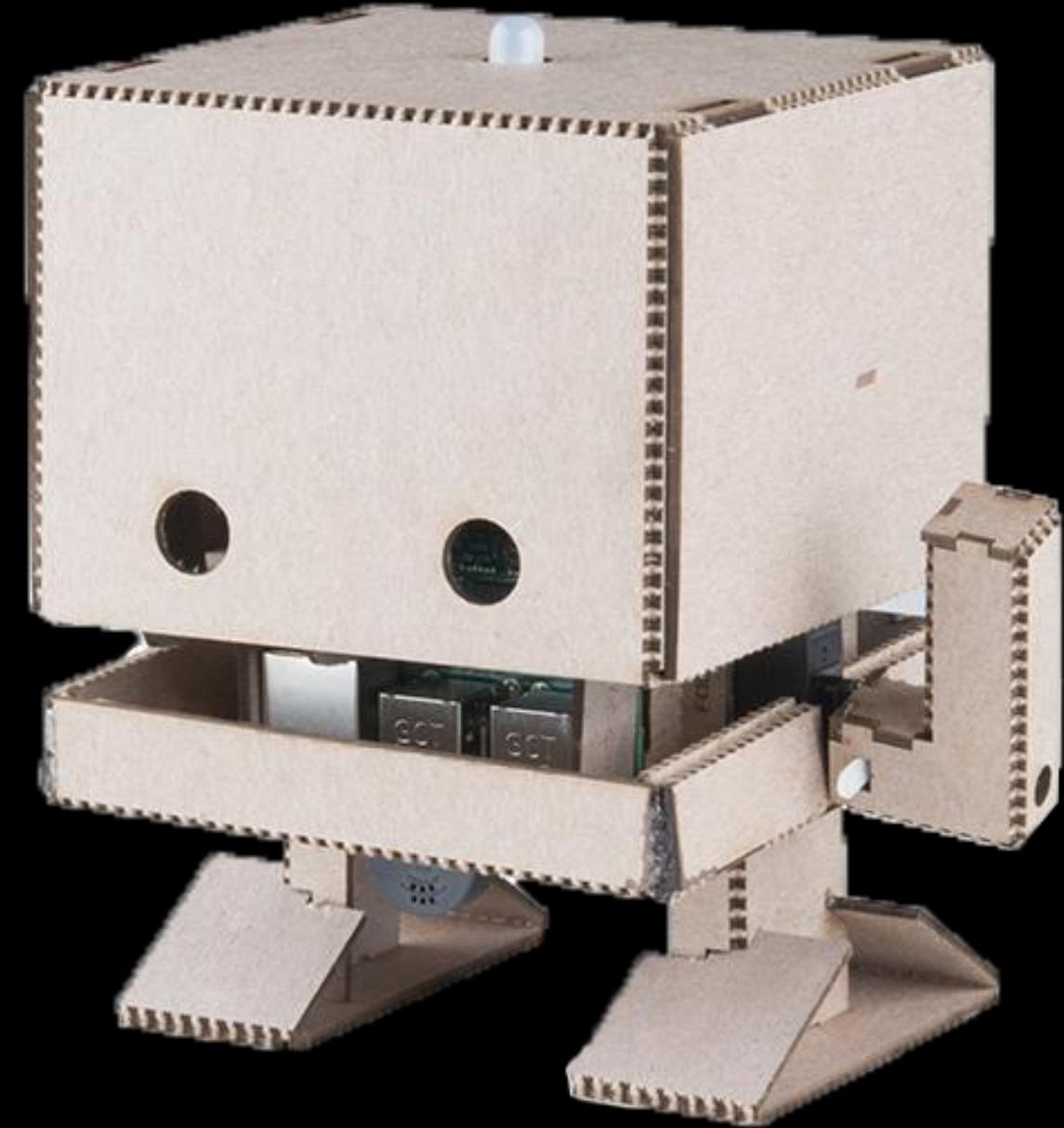


Como você  
faria?

Opção válida:

**Dispositivo no centro:**  
Um dispositivo poderoso gerencia as interações.

Ex: TJ BOT



# Gerenciamento na nuvem:

Permite o uso de diversos dispositivos, até os mais simples.

Monitore todos os passos da sua aplicação.

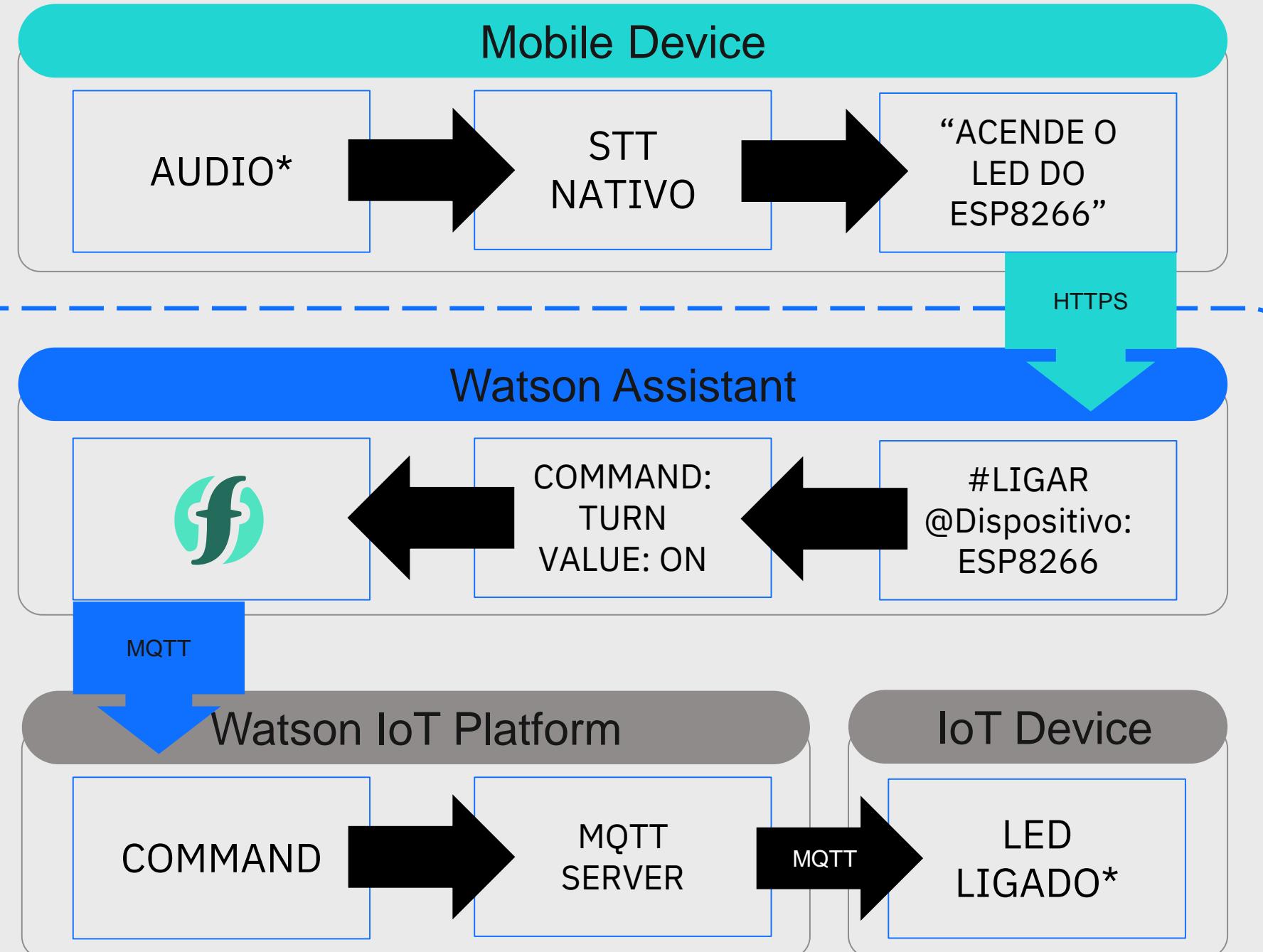
Consuma recursos sob demanda e aumente a escala de forma gradual.

Inclua novas ferramentas e serviços facilmente. Sem mudar o código do seu dispositivo.

# Arquitetura completa



IBM Cloud



# Serviços utilizados:



## Watson Assistant

Ferramenta para construção de assistentes virtuais em linguagem natural



## IBM Cloud

Nuvem que suporta a aplicação, fornecendo os recursos apresentados



## Watson IoT Platform

Ambiente comum para gerenciar e se comunicar com os dispositivos IoT



## IBM Cloud Functions

Código executado sob demanda na nuvem

Por que comprar um  
assistente virtual se você  
pode fazer o seu? É simples,  
quer ver?

# 1. Watson IoT Platform

## 1.1. Criando um novo recurso:

- No seu Dashboard clique no botão “Create Resource”

The screenshot shows the IBM Cloud Dashboard interface. At the top, there are tabs for 'Dashboard - IBM Cloud' and 'Node-RED : lab-iot-jppp.mybluemix.net'. The main content area is titled 'Dashboard' and contains sections for 'Cloud Foundry Applications' and 'Cloud Foundry Services'. In the top right corner, there is a blue button labeled 'Create resource' which is highlighted with a red rectangular box. The 'Cloud Foundry Applications' section lists one application: 'Lab-IoT-JPPP' with details: Region US South, CF Org JoaoPedroPP52..., CF Space dev, Memory (MB) 256, Status Running (1/1). The 'Cloud Foundry Services' section lists three services: 'Lab-IoT-JPPP-cloudantNoSQLDB' (with a refresh icon), 'Watson Assistant-2s', and 'omega', all with similar details. The bottom right corner of the dashboard has a 'FEEDBACK' button.

Name	Region	CF Org	CF Space	Memory (MB)	Status
Lab-IoT-JPPP	US South	JoaoPedroPP52...	dev	256	Running (1/1)

Name	Region	CF Org	CF Space	Plan	Service Offering
Lab-IoT-JPPP-cloudantNoSQLDB	US South	JoaoPedroPP52...	dev	Lite	Cloudant NoSQL ...
Watson Assistant-2s	US South	JoaoPedroPP52...	dev	Lite	Watson Assistan...
omega	US South	jp052@hotmail....	dev	Lite	Internet of Thing...

# 1. Watson IoT Platform

## 1.1. Criando um novo recurso:

- No seu Dashboard clique no botão “Create Resource”

- Na aba se busca digite “iot platform” e ache a “Internet of Things Platform”

The screenshot shows the IBM Cloud Catalog interface. At the top, there are two tabs: "Catalog - IBM Cloud" and "Node-RED : lab-iot-jppp.mybluemix.net". The URL in the address bar is <https://console.bluemix.net/catalog/?search=iot%20platform>. The main content area is titled "Catalog" and has a search bar containing "iot platform". To the left of the search bar is a red arrow pointing to it. Below the search bar is a sidebar with "All Categories (2)" and a list of categories: Compute, Containers, Networking, Storage, AI, Analytics, Databases, Developer Tools, Integration, Internet of Things (1), Security and Identity, Starter Kits (1), Web and Mobile, and Web and Application. On the right side, under the "Internet of Things" heading, there is a service card for "Internet of Things Platform" (Lite • IBM). This card is highlighted with a red box. The card description states: "This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and". Below this is another section titled "Starter Kits" with a card for "Internet of Things Platform Starter" (Lite • IBM). The card description starts with: "Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the Starter, you can quickly simulate an Internet of".

# 1. Watson IoT Platform

## 1.1. Criando um novo recurso:

- No seu Dashboard clique no botão “Create Resource”
- Na aba se busca digite “iot platform” e ache a “Internet of Things Platform”
- Dê um nome para o serviço e clique no botão “Create”

The screenshot shows the IBM Cloud Catalog interface. In the top navigation bar, there are tabs for Catalog, Docs, Support, and Manage. A search bar is also present. The main content area displays a service card for the "Internet of Things Platform". The service name is "IoT-Lab", and it is being deployed to "US South" under the organization "JoaoPedroPP520ORG" and space "dev". The "Create" button at the bottom right is highlighted with a red box.

Internet of Things Platform

Lite • IBM

This service is the hub for IBM Watson IoT and lets you communicate with and consume data from connected devices and gateways. Use the built-in web console dashboards to monitor your IoT data and analyze it in real time. Then, enhance and customize your IBM Watson IoT Platform experience by building and connecting your own apps by using messaging and REST APIs.

Service name: IoT-Lab

Choose a region/location to deploy in: US South

Choose an organization: JoaoPedroPP520ORG

Choose a space: dev

Features

- Connect
- Information Management

Need Help? Contact IBM Cloud Support

Estimate Monthly Cost Cost Calculator

Create

# 1. Watson IoT Platform

## 1.1. Criando um novo recurso:

- Clique no botão Launch Tool para começar a usar o serviço.

The screenshot shows the IBM Cloud Service Details page for the IoT-Lab service. The URL in the browser is <https://console.bluemix.net/services/iotf-service/c86bba6b-3a47-43f>. The page title is "Service Details - IBM Cloud" and the sub-title is "Node-RED : lab-iot-jppp.mybluemix.net". The main content area displays the IoT-Lab service details: "1.52% Used | 196.96 Megabyte exchanged available", "Location: US South", "Org: JoaoPedroPP520RG", and "Space: dev". A large graphic of a central processing unit (CPU) with arrows indicating data flow is centered on the page. Below it, the text "Let's get started with Watson IoT Platform" is displayed in a large, bold, teal font. A red box highlights the "Launch" button, which is located at the bottom right of the main content area. The top navigation bar includes links for Catalog, Docs, Support, Manage, and a search bar. The right side of the screen shows a user profile with the ID 1507559 - IBM.

# 1. Watson IoT Platform

## 1.2. Criando um Device Type:

- Vá para a aba Device Types;
- Clique no botão “Add Device Type”

The screenshot shows a browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform' (which is active), 'Node-RED : device-simulator-d', and 'New Tab'. The main content area is titled 'Device Types'. On the left is a sidebar with icons for various services: three dots, gear, sun, A, fingerprint, waveform, document, lock, gear, and compass. The 'Device Types' tab is highlighted with a red box. To its right is a search bar with placeholder text 'Type the name to search for' and a magnifying glass icon. On the far right of the header is a user profile with the email 'epetecof@br.ibm.com' and ID '7p8n56'. Below the header is a table with columns: Name (with a checkbox), Description (with a dropdown arrow), Number of Devices, and Class ID. The table is currently empty. At the bottom center is a large bee icon with the text 'You don't have any device types created.' and a blue 'Add Device Type' button.

# 1. Watson IoT Platform

## 1.2. Criando um Device Type:

- Vá para a aba Device Types;
- Clique no botão “Add Device Type”
- Dê um nome e uma descrição para o seu Device Type, em seguida clique em “Next”

The screenshot shows the 'Add Type' page in the IBM Watson IoT Platform. The 'Type' dropdown is set to 'Device'. The 'Name' field contains 'Simulated-Device' and the 'Description' field contains 'Um device simulado pelo Node-RED'. Both fields are highlighted with red boxes. The 'Next' button at the bottom right is also highlighted with a red box.

# 1. Watson IoT Platform

## 1.2. Criando um Device Type:

- Vá para a aba Device Types;
- Clique no botão “Add Device Type”
- Dê um nome e uma descrição para o seu Device Type, em seguida clique em “Next”.
- Deixe vazios os campos de informação e clique em “Done”

The screenshot shows the 'IBM Watson IoT Platform' dashboard with the 'Device Types' tab selected. On the left, there is a vertical sidebar with various icons. The main area is titled 'Add Type' and has three tabs: 'Identity', 'Device Information', and 'Device Types'. The 'Device Information' tab is active, showing fields for Serial Number, Model, Description, Hardware Version, Manufacturer, Device Class, Firmware Version, and Descriptive Location. A blue button labeled '+ Add Metadata' is visible. In the bottom right corner of the main form, there is a blue 'Done' button with a red rectangular border around it.

# 1. Watson IoT Platform

## 1.3. Criando um Device:

- Depois de criado seu Device Type, clique no botão “Register Devices”

The screenshot shows a browser window with four tabs: "Dashboard - IBM Cloud", "IBM Watson IoT Platform" (active), "Node-RED : device-simulator-", and "New Tab". The main content area is titled "IBM Watson IoT Platform" and has a sidebar with various icons. The "Device Types" tab is selected. A message says "You added the new device type: Simulated-Device". Below it are two buttons: "Register Device" (highlighted with a red box) and "Advanced Flow". A large circular graphic with a central gear icon is on the right. The URL in the address bar is <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/deviceTypes-v2>.

# 1. Watson IoT Platform

## 1.3. Criando um Device:

- Depois de criado seu Device Type, clique no botão “Register Devices”
- Dê um nome para o seu device e clique em “Next”

The screenshot shows the 'Add Device' interface in the IBM Watson IoT Platform. On the left, there's a vertical sidebar with various icons. The main area has tabs for 'Browse', 'Action', 'Device Types', 'Add Device', 'Identity', 'Device Information', 'Security', and 'Summary'. The 'Identity' tab is active. It prompts the user to select a device type and provide a unique ID. The 'Device Type' dropdown is set to 'Simulated-Device'. The 'Device ID' input field contains 'my-first-device', which is highlighted with a red box. At the bottom right of the form, there are 'Cancel' and 'Next' buttons, with 'Next' also highlighted with a red box.

# 1. Watson IoT Platform

## 1.3. Criando um Device:

- Depois de criado seu Device Type, clique no botão “Register Devices”
- Dê um nome para o seu device e clique em “Next”
- Deixe vazios os campos de informação e clique em “Next”

The screenshot shows the 'Device Information' page of the IBM Watson IoT Platform. The left sidebar has icons for various device types: Grid, Sensor, Actuator, Fingerprint, Network, Storage, Security, Configuration, and Location. The main navigation bar includes 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Node-RED : lab-iot-jppp.mybluemix.net'. The URL is https://1603q6.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2. The user is logged in as jp052@hotmail.com (ID: 1603q6). The page title is 'IBM Watson IoT Platform' and the sub-section is 'Device Information'. A note says: 'You can modify the default device information and enter more information about the device for identification purposes.' There are input fields for 'Serial Number', 'Model', 'Description', 'Hardware Version', 'Manufacturer', 'Device Class', 'Firmware Version', and 'Descriptive Location'. A blue button labeled '+ Add Metadata' is at the bottom left of the form area. At the bottom right, there are 'Back' and 'Next' buttons, with the 'Next' button being red.

# 1. Watson IoT Platform

## 1.3. Criando um Device:

- Deixe vazio o campo do token e clique em “Next” para gerar um token automaticamente.

The screenshot shows the IBM Watson IoT Platform interface for creating a new device. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Node-RED : lab-iot-jppp.myblu'. The main title is 'IBM Watson IoT Platform' with 'Browse' selected. On the left, there's a sidebar with icons for various device types: Grid, Sensors, People, Location, Actions, Fingerprint, Accelerometer, Light, Temperature, Humidity, and GPS. The central panel displays a section titled 'TOKEN (DEFAULT)' which explains that the service generates an authentication token for the device. It specifies that tokens are 18 characters long and contain a mix of alphanumeric characters and symbols. A note states that tokens are returned at the end of the registration process. Below this, there's a field labeled 'Authentication Token' with a placeholder 'Enter an optional token' and an information icon. A note below the field says: 'Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.' Another note states: 'Authentication token are encrypted before we store them.' At the bottom right, there are 'Back' and 'Next' buttons, with the 'Next' button being highlighted with a red box.

# 1. Watson IoT Platform

## 1.3. Criando um Device:

- Deixe vazio o campo do token e clique em “Next” para gerar um token automaticamente.

- Clique em “Done” para terminar a criação e gerar o token.

The screenshot shows a browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform', 'Node-RED : device-simulator-d', and 'New Tab'. The active tab is 'IBM Watson IoT Platform' at the URL <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2?add=Simulated-Device>. The page title is 'IBM Watson IoT Platform' and the sub-page title is 'Add Device'. The main content area is titled 'Summary' and contains the following information:

- Device Type: Simulated-Device
- Device ID: my-first-device
- Security Token: To be generated

A blue button labeled 'View Metadata' is visible. At the bottom right, there are two buttons: a blue square with a white left arrow and a red-bordered blue rectangle labeled 'Done'.

# 1. Watson IoT Platform

## 1.3. Criando um Device:

- Deixe vazio o campo do token e clique em “Next” para gerar um token automaticamente.

- Clique em “Done” para terminar a criação e gerar o token.

**- Copie seu token e guarde no bloco de notas para usar no futuro.**

The screenshot shows a web browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform', 'Node-RED : device-simulator-d', and 'New Tab'. The active tab is 'IBM Watson IoT Platform' at the URL <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2?add=Simulated-Device>. The page title is 'IBM Watson IoT Platform' and the user is identified as 'epetecof@br.ibm.com ID: 7p8n56'. On the left, there's a sidebar with icons for various device management tasks. The main content area shows a device named 'my-first-device' with a section titled 'Device Credentials'. It displays the following information:

Organization ID	7p8n56
Device Type	Simulated-Device
Device ID	my-first-device
Authentication Method	use-token-auth
Authentication Token	(Redacted)

A warning message at the bottom states: 'Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token.'

# 1. Watson IoT Platform

## 1.4. Configurando o tipo de conexão utilizada.

- Usando a navegação lateral, entre no menu de segurança (Cadeado). Em seguida clique no botão relacionado a “Connection Security”.

The screenshot shows the IBM Watson IoT Platform Policies dashboard. The left sidebar has a 'Policies' tab selected, along with other icons for Service Details, Watson Assistant, Node-RED, and NLU. The main content area is titled 'Policies' and contains three sections: 'Connection Security' (with a red box around the edit icon), 'Blacklist' (disabled), and 'Whitelist' (disabled).

**Policies**

You can configure policies to enhance connection security and control access to the server from devices.

**Connection Security**  
Configure the security level for device connection.

**Blacklist**  
Block access from specific IP addresses and countries. Activating a blacklist disables an active whitelist.  
Disabled

**Whitelist**  
Allow access from specific IP addresses and countries. Activating a whitelist disables an active blacklist.  
Disabled

Cookie Preferences

# 1. Watson IoT Platform

## 1.4. Configurando o tipo de conexão utilizada.

- Usando a navegação lateral, entre no menu de segurança (Cadeado). Em seguida clique no botão relacionado a “Connection Security”.

- Na opção Security Level selecione “TLS Optional”

The screenshot shows the IBM Watson IoT Platform security configuration interface. The left sidebar has a 'Connection Security' icon highlighted. The main page title is 'Default Rule'. It defines the default connection security level for all device types. A note states that device numbers and predicted compliance values are estimates based on a report running at varying intervals. The table shows one device with TLS with Token Authentication selected. A dropdown menu for 'Security Level' is open, showing 'TLS Optional' selected and highlighted with a red box. Other options in the dropdown include 'TLS with Client Certificate Authentication', 'TLS with Client Certificate AND T...', and 'TLS with either Client Certificate ...'. A note below the table says 'Custom Rules' allow defining connection rules for specified device types. The top navigation bar shows tabs for Service Details, Watson Assistant, Watson Assistant, Node-RED, and NLU-IoT/esp-sensor.js, along with a search bar and user information.

Scope	Security Level	Predicted Compliance	# of Devices
Default	TLS with Token Authentication	<div style="width: 100px; height: 10px; background-color: #f00;"></div> 0 Pass 1 Fail 0 Unknown	1 device
	TLS Optional		
	TLS with Client Certificate Authentication		
	TLS with Client Certificate AND T...		
	TLS with either Client Certificate ...		

# 1. Watson IoT Platform

## 1.4. Configurando o tipo de conexão utilizada.

- Usando a navegação lateral, entre no menu de segurança (Cadeado). Em seguida clique no botão relacionado a “Connection Security”.

- Na opção Security Level selecione “TLS Optional”. Salve.

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, Node-RED : lab-iot-jp, and NLU-IoT/esp-sensor.js. The main title is "IBM Watson IoT Platform" under "Default Rule". The left sidebar has icons for various services: Cloud Foundry, Watson Assistant, Watson Assistant, Node-RED, and NLU-IoT/esp-sensor.js. The main content area displays a table with one row:

Scope	Security Level	Predicted Compliance	# of Devices
Default	TLS Optional	Refresh compliance	1 device

A red box highlights the "TLS Optional" dropdown in the "Security Level" column. Below the table, there is a section titled "Custom Rules" with the following text: "You can define custom connection rules for specific device types. Custom rules overwrite the default rule for the specified device types. The predicted compliance value is updated to reflect the default settings and the custom settings." At the bottom right of the main content area are "Cancel" and "Save" buttons, with "Save" being highlighted by a red box.

# 1. Watson IoT Platform

## 1.5. Ativando o cache de eventos.

- Na aba se configurações gerais (Engrenagem). Procure pela seção “Last Event Cache” e ative a opção.

The screenshot shows the IBM Watson IoT Platform dashboard with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/settings>. The left sidebar has a dark theme with various icons. The main content area is titled "Last Event Cache". It contains a description of what LEC is, a toggle switch labeled "Activate Last Event Cache" which is turned on (indicated by a checked checkbox), and a section for "Events stored in cache" with two options: "1 day" and "7 days", where "7 days" is highlighted with a red box. Below this is a section for "Custom Device Management Packages" with a "Add Package" button.

IBM Watson IoT Platform

E-mail | Reset | Verify | Email | Service | IBM V | IBM C | IBM V | Dash | Node | Node | GitHub | +

https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/settings

eduardo.petecof@aluno.ufabc.edu.br  
ID: 7y60pl

IBM Watson IoT Platform

PLATFORM

- About
- Identity
- Experimental Features**
- Last Event Cache

DATA AND DEVICES

- Custom Device Management
- Packages
- Device Simulator

SECURITY

- Connection Security
- CA Certificates
- Messaging Server Certificates
- Group Access beta

Last Event Cache

The last event cache (LEC) stores information about the last event a connected device sent to the platform. For more information, see the last event cache [documentation](#)

Activate Last Event Cache

Events stored in cache

1 day **7 days**

Custom Device Management Packages

Custom device management packages can be uploaded to add new device management functions. Custom device management packages must consist of valid JSON in order to be successfully added. For more details see the custom device management package [documentation](#)

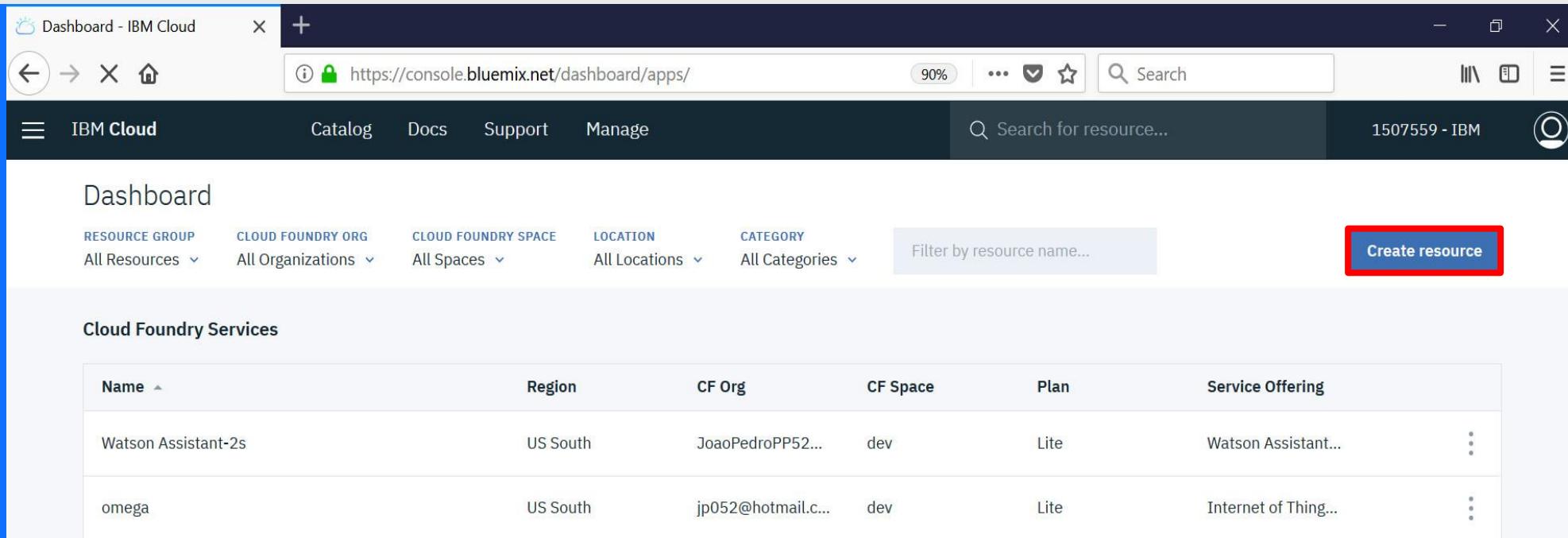
+ Add Package

Package	Version	Description

# 3. Criando um dispositivo simulado com o Node-RED

## 3.1. Criando uma instância do Node-RED.

- Voltando ao Dashboard. Crie um novo recurso.



The screenshot shows the IBM Cloud Dashboard interface. At the top, there is a header with the title 'Dashboard - IBM Cloud', a search bar containing 'https://console.bluemix.net/dashboard/apps/', and a red box highlighting the 'Create resource' button on the right side of the header. Below the header, there is a navigation bar with links for 'Catalog', 'Docs', 'Support', and 'Manage'. A search bar labeled 'Search for resource...' is also present. The main area is titled 'Cloud Foundry Services' and contains a table with two rows of data. The columns in the table are 'Name', 'Region', 'CF Org', 'CF Space', 'Plan', and 'Service Offering'. The first row has a name of 'Watson Assistant-2s', a region of 'US South', a CF Org of 'JoaoPedroPP52...', a CF Space of 'dev', a Plan of 'Lite', and a Service Offering of 'Watson Assistant...'. The second row has a name of 'omega', a region of 'US South', a CF Org of 'jp052@hotmail.c...', a CF Space of 'dev', a Plan of 'Lite', and a Service Offering of 'Internet of Thing...'. Each row has a three-dot menu icon on the far right.

Name	Region	CF Org	CF Space	Plan	Service Offering
Watson Assistant-2s	US South	JoaoPedroPP52...	dev	Lite	Watson Assistant...
omega	US South	jp052@hotmail.c...	dev	Lite	Internet of Thing...

# 3. Criando um dispositivo simulado com o Node-RED

## 3.1. Criando uma instância do Node-RED.

- Voltando ao Dashboard. Crie um novo recurso.
- Na aba de busca digite “node-red” e ache o “Node-RED Starter”.

The screenshot shows the IBM Cloud Catalog interface. At the top, there is a search bar with the query "node-red". Below the search bar, there is a sidebar with categories like Compute, Containers, Networking, Storage, AI, Analytics, Databases, Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits (2), Web and Mobile, and Web and Application. The main area displays "Starter Kits" with two items: "Internet of Things Platform Starter" and "Node-RED Starter". The "Node-RED Starter" item is highlighted with a red box. It includes a description: "This application demonstrates how to run the Node-RED open-source project within IBM Cloud." At the bottom right, there is a section titled "Looking for more?" with a link to "IBM Cloud Experimental Services".

# 3. Criando um dispositivo simulado com o Node-RED

## 3.1. Criando uma instância do Node-RED.

- Voltando ao Dashboard. Crie um novo recurso.
- Na aba de busca digite “node-red” e ache o “Node-RED Starter”.
- Dê um nome para a sua aplicação (recomenda-se que ele tenha uma referência ao seu nome) e crie.

The screenshot shows a browser window with four tabs: "Node-RED Starter - IBM Cloud", "IBM Watson IoT Platform", "Node-RED : device-simulator-d", and "New Tab". The main content is the IBM Cloud Catalog. A search bar at the top right contains "Search". Below it, a navigation bar has links for "IBM Cloud", "Catalog", "Docs", "Support", "Manage", and a search bar "Search for resource...". On the far right, it says "1713407 - IBM" and has a user profile icon. The main area displays a "Create a Cloud Foundry App" section with a "Node-RED Starter" card. The card includes a "View all" link, a "Create" button with a wrench icon, the text "Lite • Community", and a brief description: "This application demonstrates how to run the Node-RED open-source project within IBM Cloud." It also lists "View Docs", "VERSION 0.8.1", "TYPE Boilerplate", and "LOCATION Sydney, Germany, United Kingdom, US East, US South". To the right of the card, there are configuration fields: "App name:" with value "epetecof-simulated-device" (highlighted with a red box), "Host name:" with value "epetecof-simulated-device", "Domain:" with value "mybluemix.net", "Choose a region/location to deploy in:" with value "US South", "Choose an organization:" with value "WCP Code Bakery", "Choose a space:" with value "epetecof", and a "Selected Plan:" dropdown with "SDK for Node.js™" and "Cloudant" options. At the bottom, there are links for "Need Help?", "Contact IBM Cloud Support", "Estimate Monthly Cost", and "Cost Calculator", along with a large blue "Create" button (also highlighted with a red box). A "FEEDBACK" button is visible on the far right edge.

# 3. Criando um dispositivo simulado com o Node-RED

## 3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”

The screenshot shows the IBM Cloud Application Details page for an application named "edupetecof-simulated-device". The application is currently "Starting". A red box highlights the "Visit App URL" link. The page includes sections for Overview, Runtime, Connections, Logs, API Management, and Monitoring. It also provides instructions for using the IBM Cloud command line interface (CLI) and lists restrictions regarding Cygwin support. Step-by-step instructions for getting started are provided, including changing to the directory and making changes to the app code.

Cloud Foundry apps /

edupetecof-simulated-device Starting [Visit App URL](#) Routes

Org: eduardo.petecof@aluno.ufabc.edu.br Location: US South Space: dev

Download, modify, and redeploy your Cloud Foundry app with the command line interface

Last Updated: 2018-05-24 | [Edit in GitHub](#)

Use IBM Cloud command line interface to download, modify, and redeploy your Cloud Foundry applications and service instances.

Before you begin, download and install the IBM Cloud CLI [External link icon](#)

**Restriction:** The command line tool is not supported by Cygwin. Use the tool in a command line window other than the Cygwin command line window.

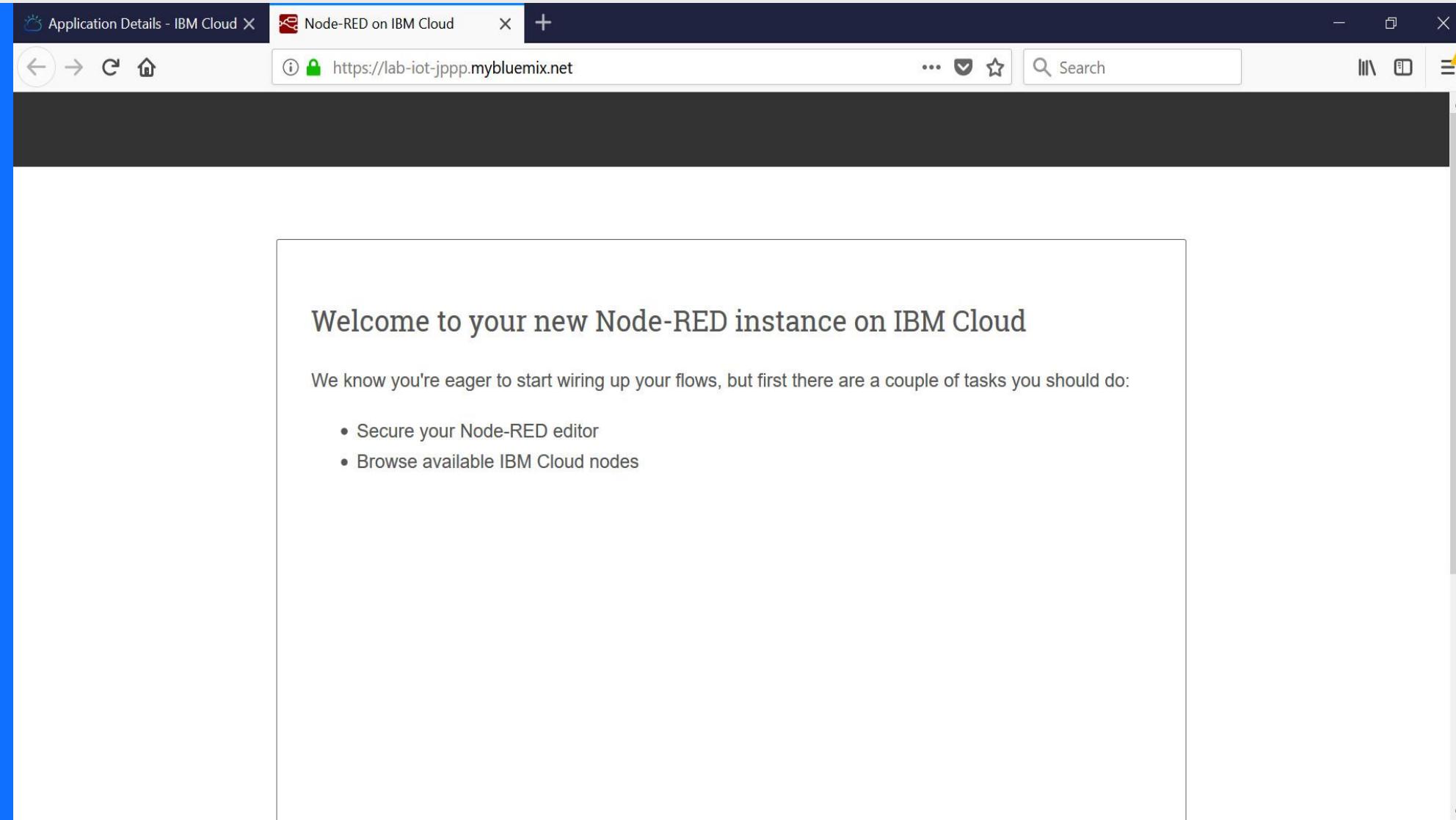
After you install the command line interface, you can get started:

- ① Change to the directory where your code is located.  
\$ cd your\_new\_directory [Copy icon](#)
- ② Make changes to your app code as you see fit. For example, if you are using a IBM® Cloud sample application and your

### 3. Criando um dispositivo simulado com o Node-RED

#### 3.2. Configurando o seu Node-RED.

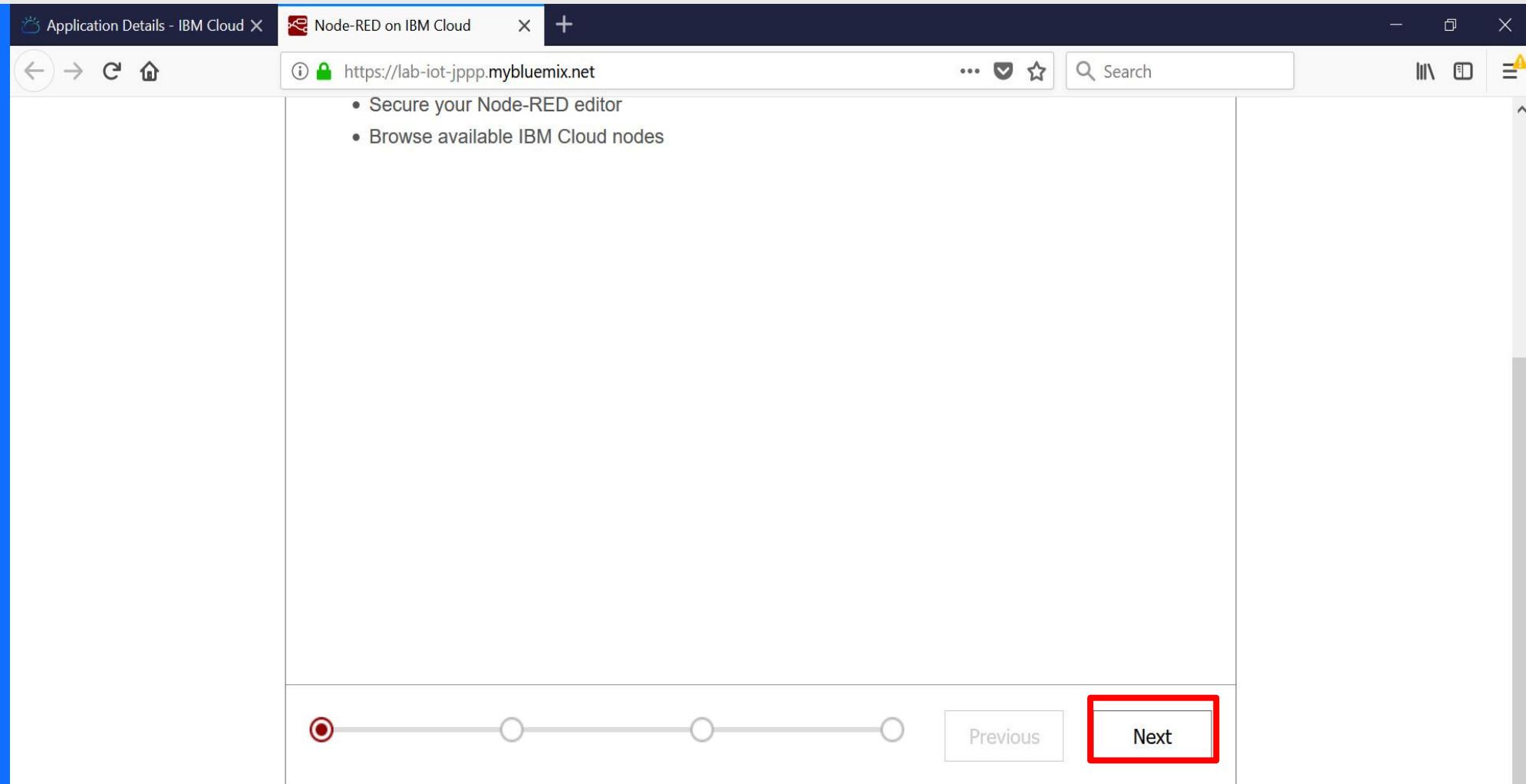
- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.



### 3. Criando um dispositivo simulado com o Node-RED

#### 3.2. Configurando o seu Node-RED.

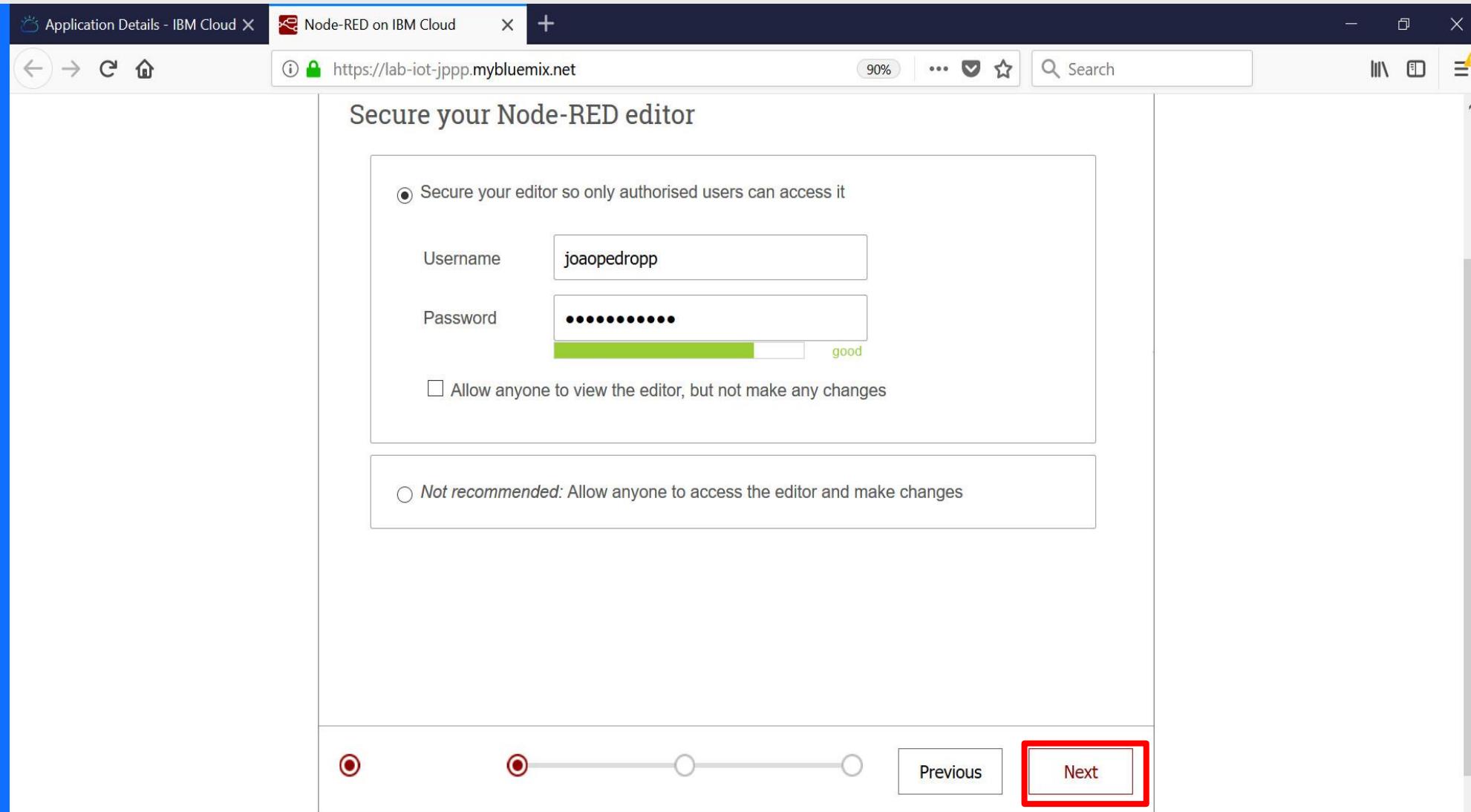
- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.



### 3. Criando um dispositivo simulado com o Node-RED

#### 3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.
- Defina um username e uma senha para editors.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.
- Defina um username e uma senha para editores.
- Pule as próximas telas.

Application Details - IBM Cloud X Node-RED on IBM Cloud X +

https://lab-iot-jppp.mybluemix.net 90% Search

Browse available IBM Cloud nodes

There are lots of nodes available from the community that can be used to add more capabilities to your application. The list below is just a small selection.

You can find many more nodes on the [Flow Library](#).

You can use the Palette Manager built into editor to search for and install nodes. Alternatively, you can also edit your application's package.json file and adding them to the dependencies section.

node-red-dashboard  
Quickly create dashboards driven by Node-RED

node-red-contrib-ibm-wiotp-device-ops  
Perform device and gateway operations using the Watson IoT Platform

node-red-contrib-iot-virtual-device  
Simulate device behavior and use it to run many device instances

node-red-contrib-objectstore  
Store, delete and restore objects in the ObjectStore service

node-red-contrib-bluemix-hdfs

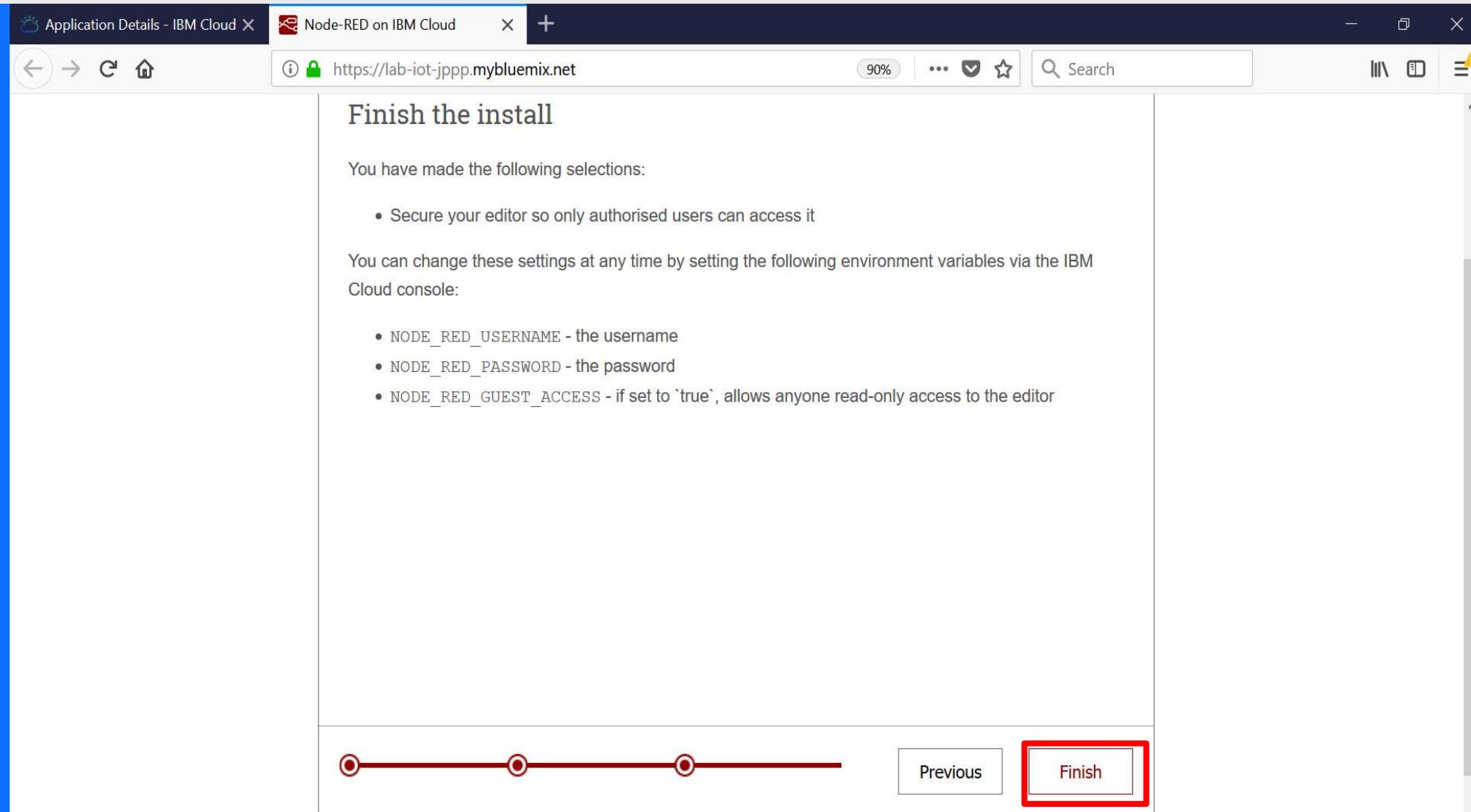
node-red-contrib-ibmpush

Previous Next

### 3. Criando um dispositivo simulado com o Node-RED

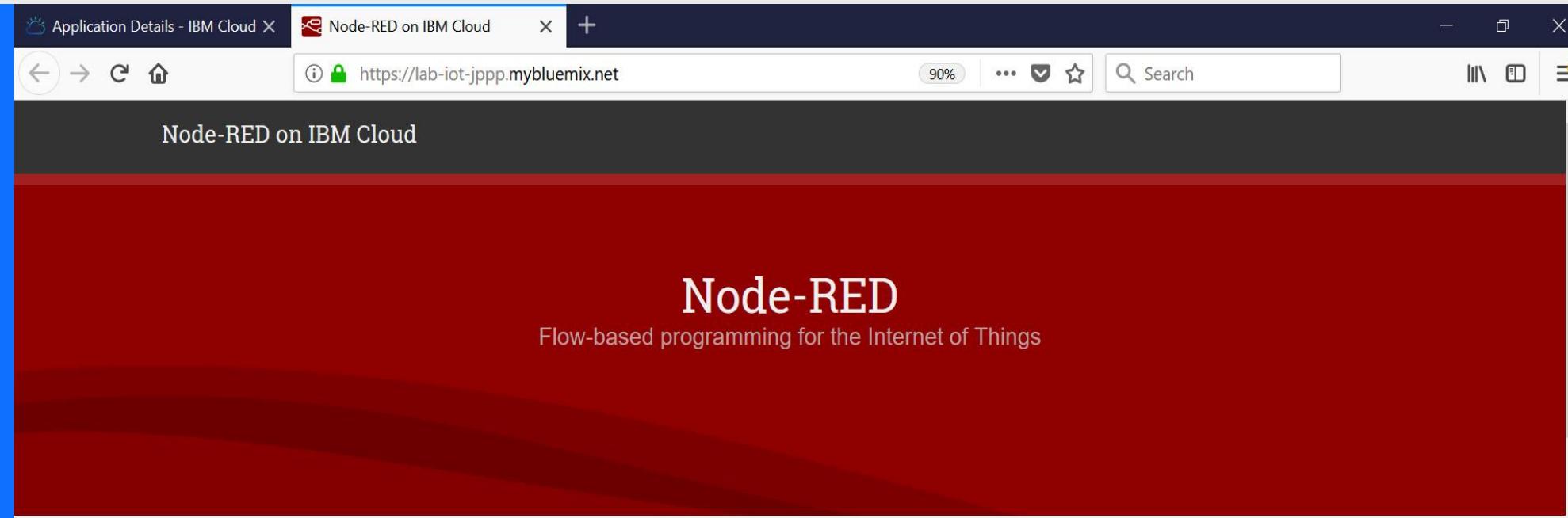
#### 3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.
- Defina um username e uma senha para editors.
- Pule as próximas telas.



### 3. Criando um dispositivo simulado com o Node-RED

#### 3.2. Configurando o seu Node-RED.



Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at [nodered.org](http://nodered.org).

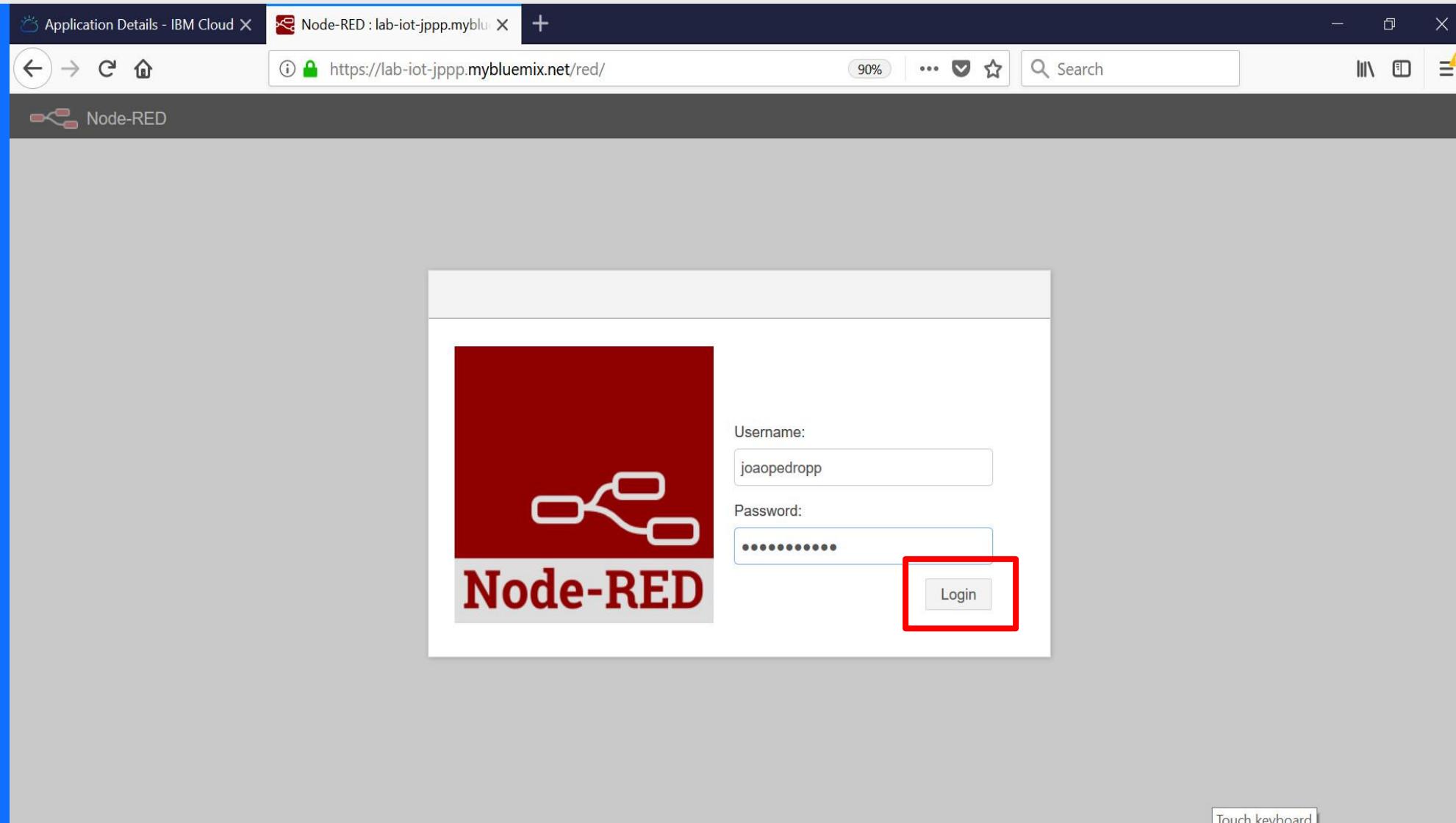
[Go to your Node-RED flow editor](#)

[Learn how to customise Node-RED](#)

### 3. Criando um dispositivo simulado com o Node-RED

#### 3.2. Configurando o seu Node-RED.

- Faça login com o username e senha escolhidos anteriormente.



# 3. Criando um dispositivo simulado com o Node-RED

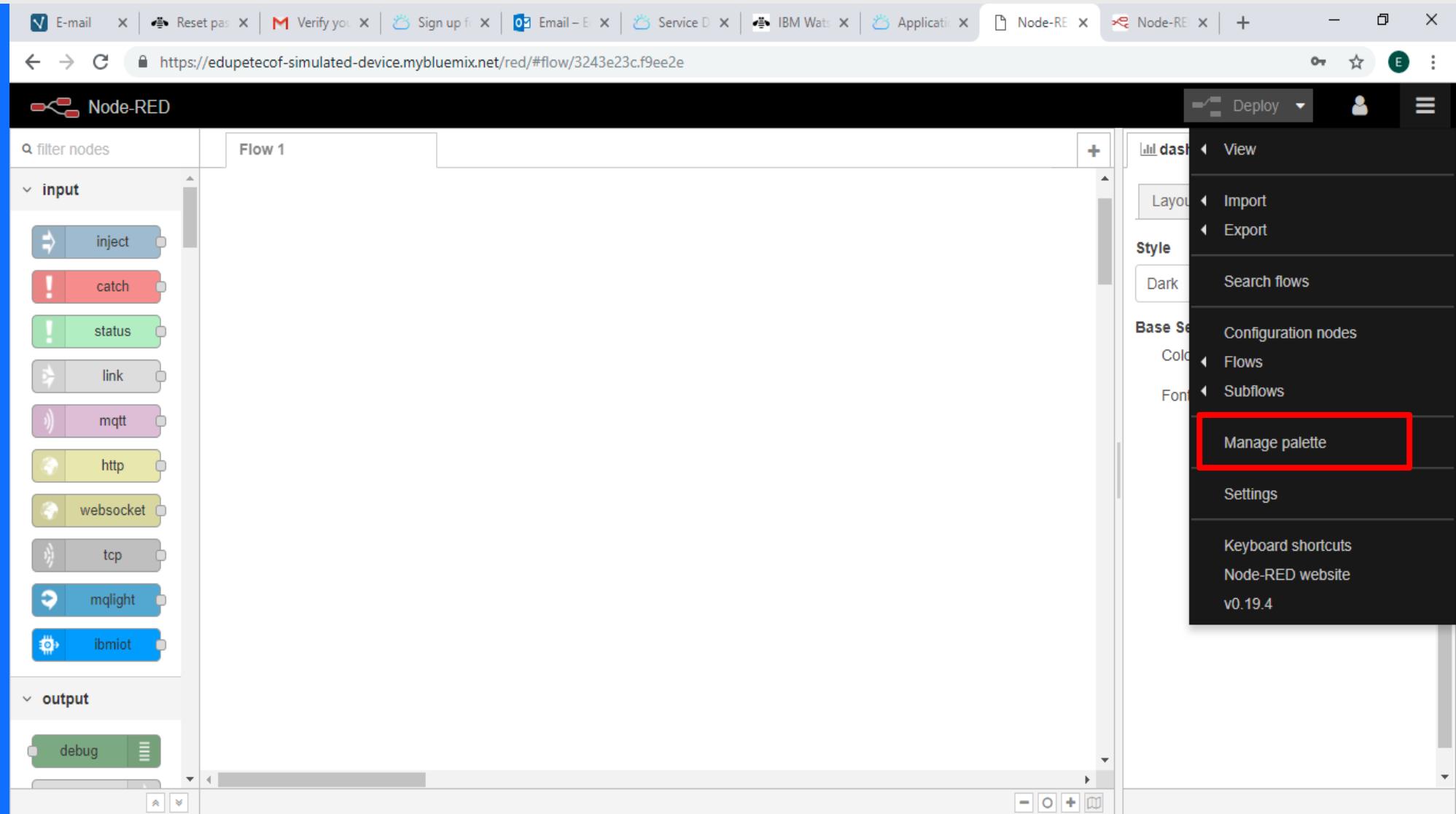
## 3.3. Instale o dashboard.

The screenshot shows the Node-RED interface running on IBM Cloud. The browser title is "Node-RED : lab-iot-jppp.mybluemix.net". The main area displays "Flow 1" which contains a single "inject" node connected to a "debug" output node. The left sidebar lists various input nodes: inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, and ibmiot. The right sidebar shows the "info" panel with the flow details: Flow ID "1ebc4a3c.e6699e", Name "Flow 1", and Status "Enabled". A message at the bottom right says, "You can confirm your changes in the node edit tray with **ctrl-enter** or cancel them with **ctrl-escape**".

# 3. Criando um dispositivo simulado com o Node-RED

## 3.3. Instale o dashboard.

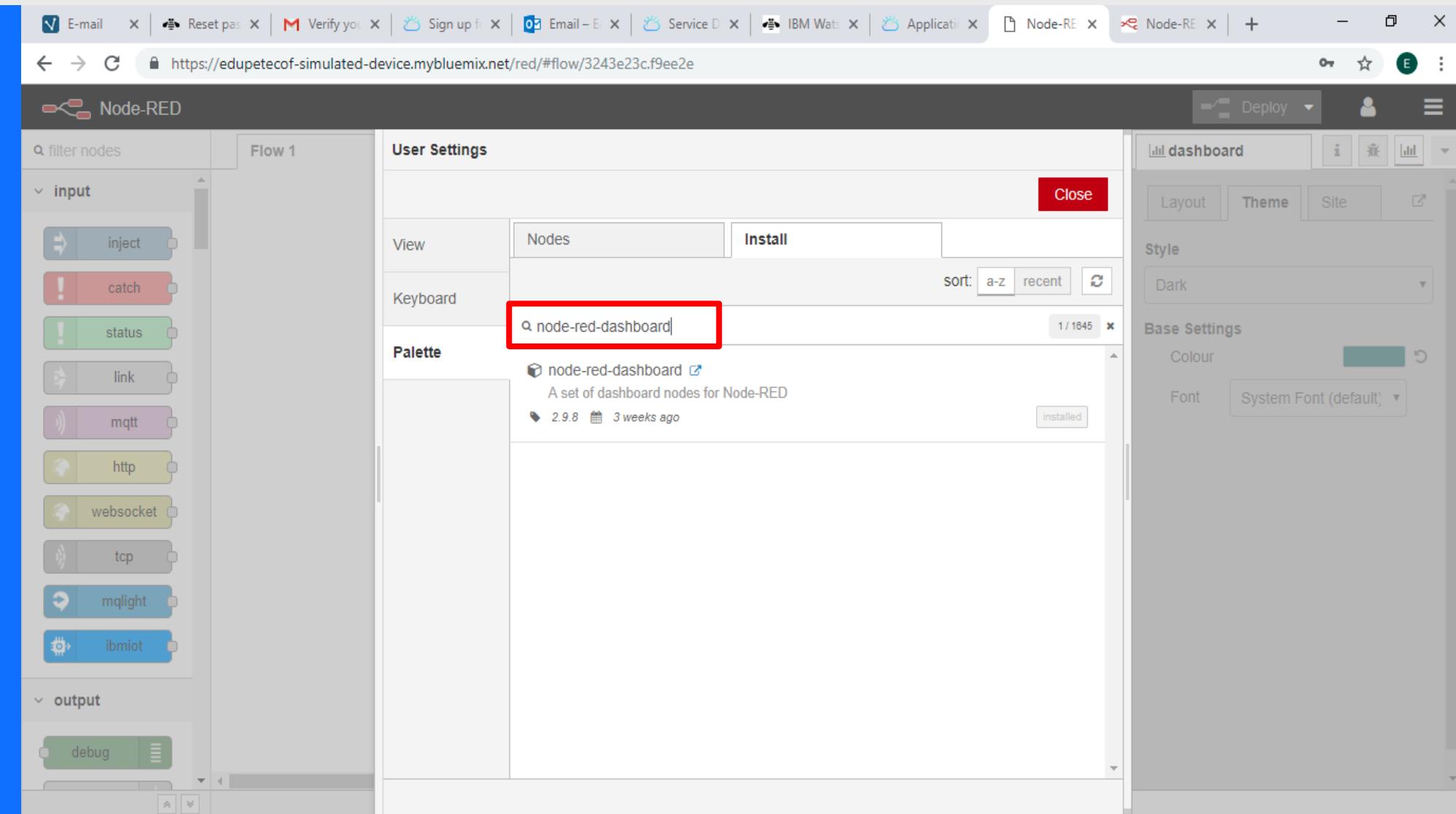
- No menu à direta, navegue para “Manage Palette”.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.3. Instale o dashboard.

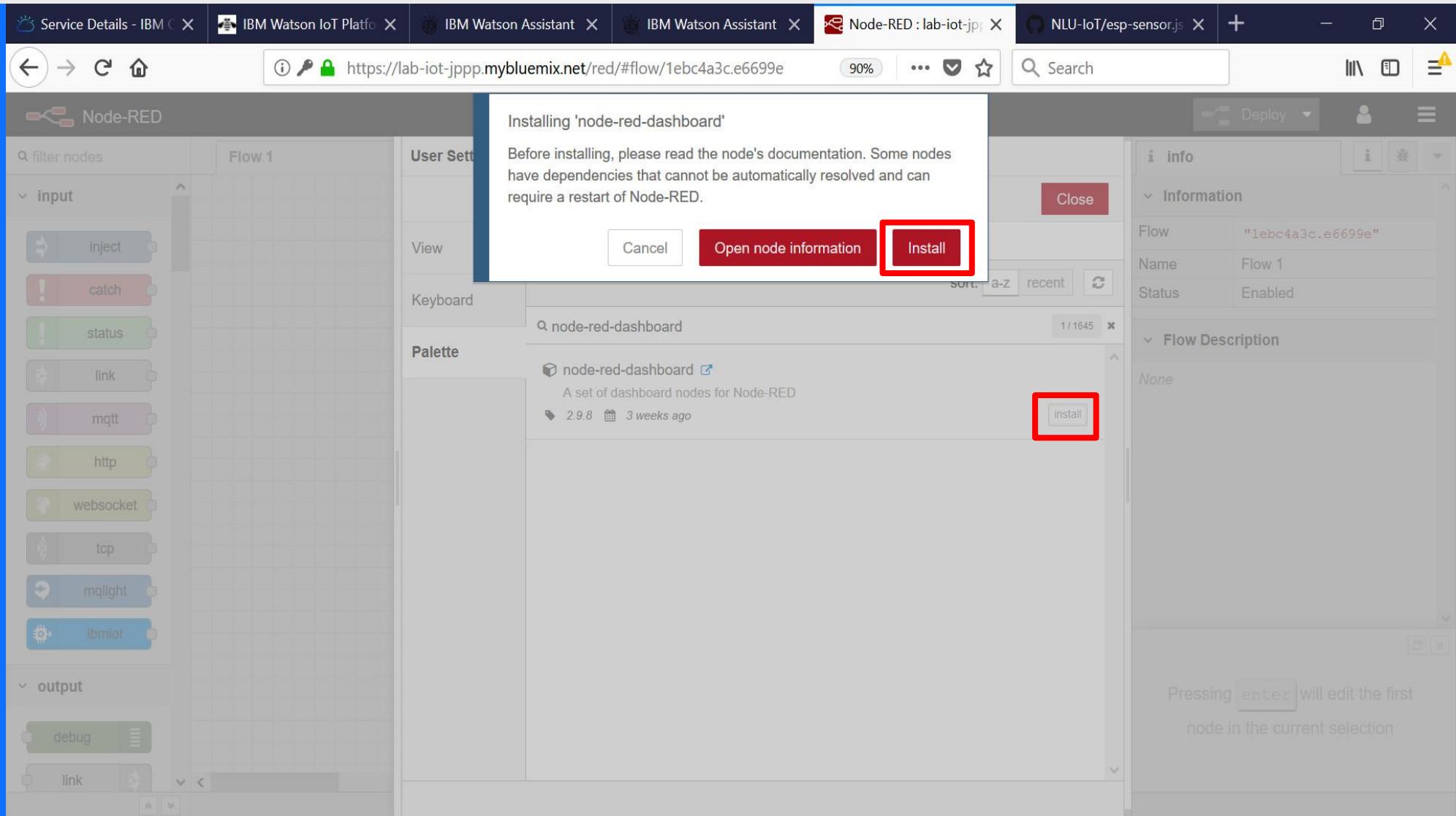
- No menu à direta, navegue para “Manage Palette”.
- Na aba de busca procure por “node-red-dashboard”.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.3. Instale o dashboard.

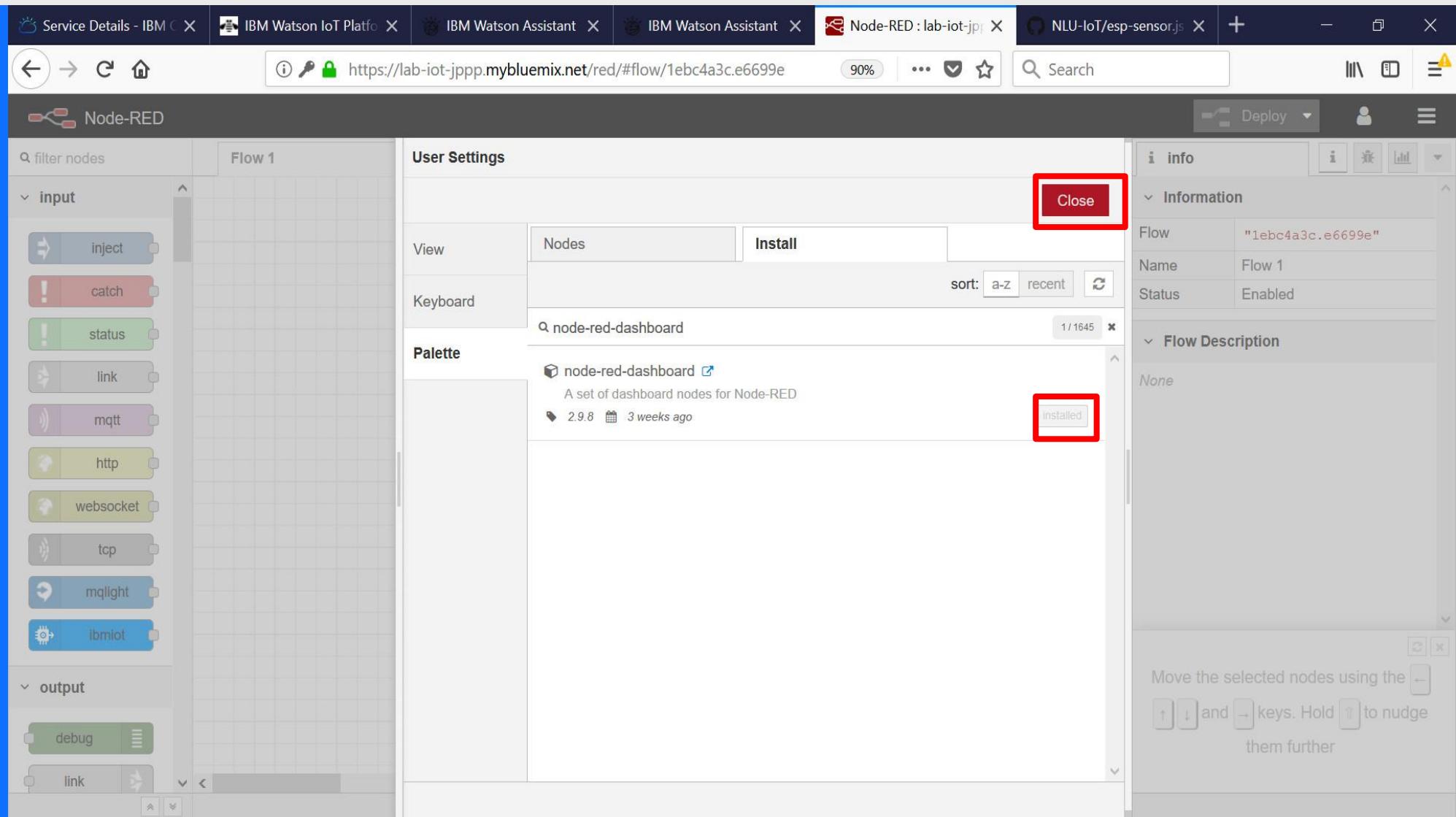
- No menu à direta, navegue para “Manage Palette”.
- Na aba de busca procure por “node-red-dashboard”.
- Instale o pacote.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.3. Instale o dashboard.

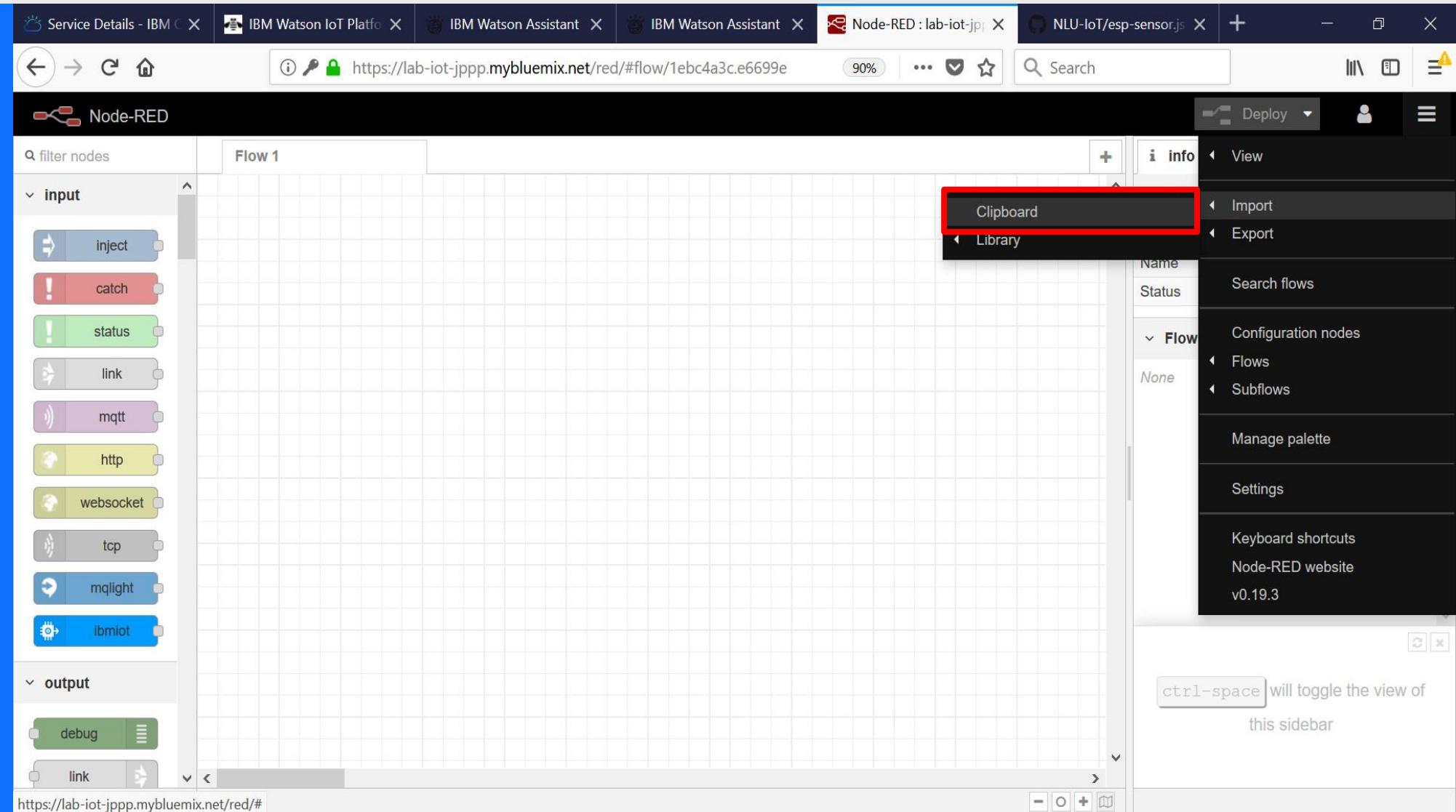
- No menu à direta, navegue para “Manage Palette”.
- Na aba de busca procure por “node-red-dashboard”.
- Instale o pacote. Volte para a tela principal.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.4. Import o flow.

- No menu à direta,  
navegue para  
“Import >  
Clipboard”.



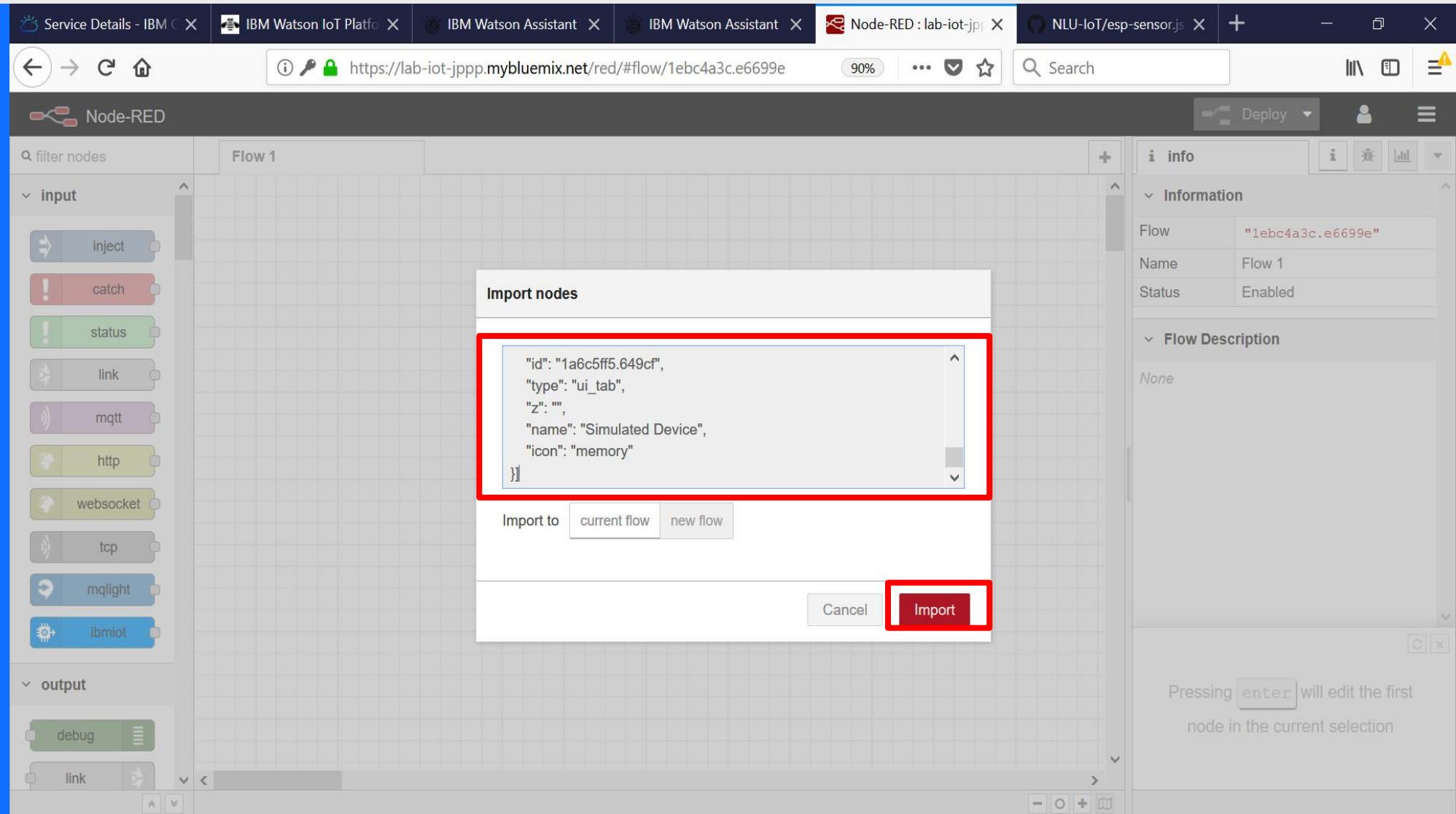
# 3. Criando um dispositivo simulado com o Node-RED

## 3.4. Import o flow.

- No menu à direta, navegue para “Import > Clipboard”.

- Copie o código no git:  
<https://github.com/Bluemix-Estag/NLU-IoT/blob/master/Node-RED/hands-on-flow.json>

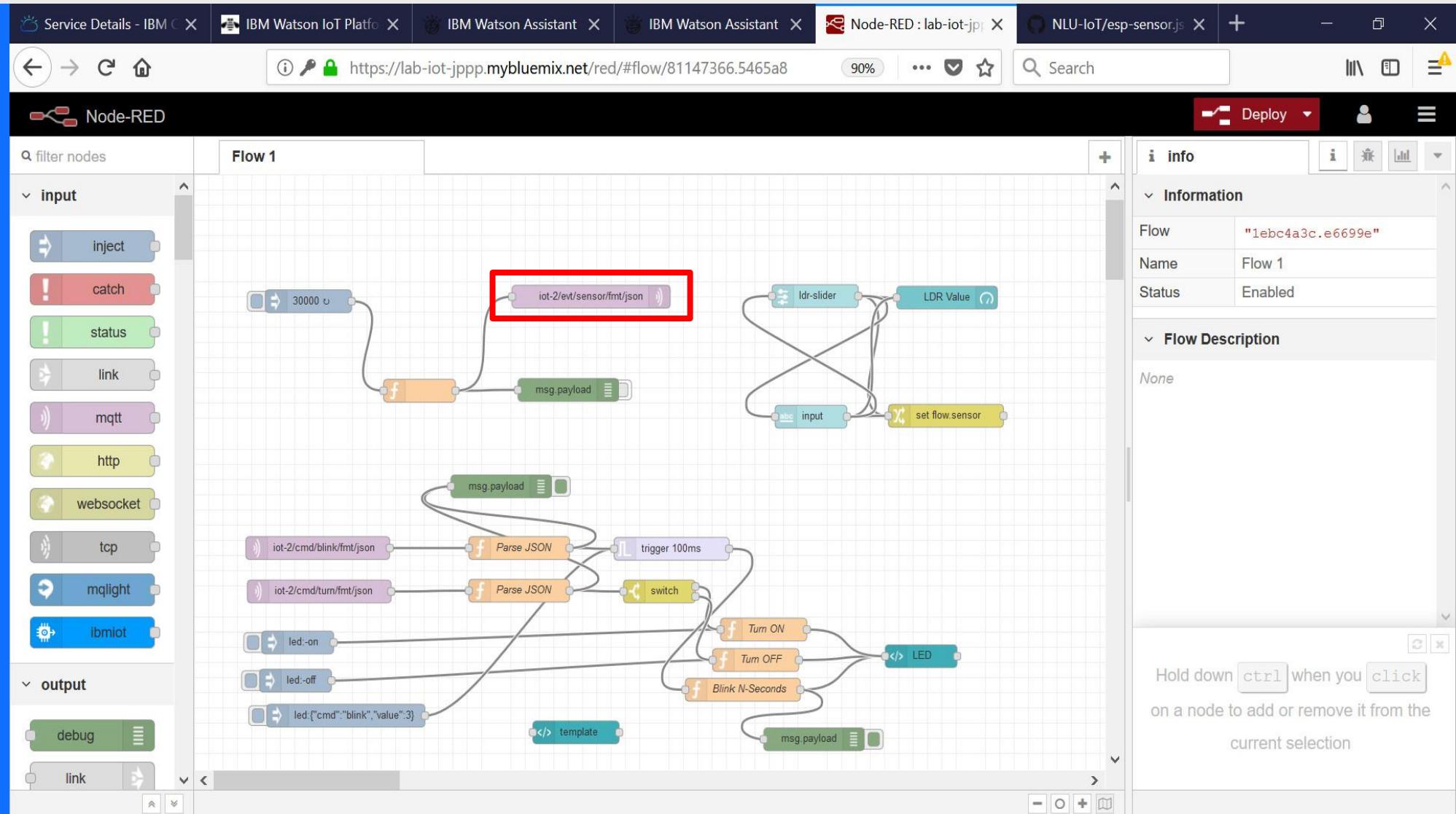
- Cole na caixa e importe.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.
- Edite as configurações do broker.

The screenshot shows the Node-RED interface with a flow titled "Flow 1". The flow starts with an "inject" node, followed by a "catch" node, a "status" node, a "link" node, an "mqtt" node, an "http" node, a "websocket" node, a "tcp" node, a "mqlight" node, and an "ibmiot" node. The "output" node is a "debug" node. A specific "mqtt out" node is highlighted and open for configuration. The configuration dialog shows the following settings:

- Server:** d:{your\_org}:{your\_device\_type}:{your\_d} (with a red box around the edit icon)
- Topic:** iot-2/evt/sensor/fmt/json
- QoS:** 0
- Retain:** true
- Name:** Name

A tip message at the bottom of the dialog box states: "Tip: Leave topic, qos or retain blank if you want to set them via msg properties."

The right panel displays the "info" and "Node Help" sections for the mqtt out node. The "info" section shows the node's ID ("fe2028e7.65baa") and type ("mqtt out"). The "Node Help" section provides details about connecting to an MQTT broker and publishing messages.

# 3. Criando um dispositivo simulado com o Node-RED

## 3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.
- Edite as configurações do broker.
- Coloque as informações da plataforma de IoT. O nome da sua org, o seu Device type e o Device ID.

The screenshot shows the Node-RED interface with a flow editor and a configuration panel for an mqtt-broker node.

**Flow Editor:** On the left, there's a sidebar with categories like 'input' (inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot) and 'output' (debug, link). The main area shows a flow starting with an 'inject' node, followed by a 'status' node, a '30000' timer node, and an 'f' function node. This is connected to an 'iot-2/cmd/blink/fmt/json' message node, which then connects to an 'led-on' node. Another 'iot-2/cmd/turn/fmt/json' message node connects to an 'led-off' node, and finally to a 'led:{"cmd": "blink", "value": 3}' message node.

**Configuration Panel:** A modal window titled 'Edit mqtt out node > Edit mqtt-broker node' is open. It has tabs for 'Connection', 'Security', and 'Messages'. The 'Connection' tab is selected, showing fields for 'Server' (set to '{your\_org}.messaging.internetofthings') and 'Client ID' (set to 'd:{your\_org}:{your\_device\_type}:{your\_device\_id}'). Both fields are highlighted with red boxes. Other settings include 'Port' (1883), 'Enable secure (SSL/TLS) connection' (unchecked), 'Keep alive time (s)' (60), 'Use clean session' (checked), and 'Use legacy MQTT 3.1 support' (checked).

**Info Panel:** On the right, there's an 'info' panel with 'Information' and 'Node Help' sections. The 'Information' section shows the node details: Node ID is "12dc1008.03a508", Type is "mqtt-broker". The 'Node Help' section provides instructions for connecting to an MQTT broker, mentioning the creation of a single connection that can be reused by MQTT In and MQTT Out nodes. It also notes that a random Client ID will be generated if none is set and the node is configured for Clean Session.

# 3. Criando um dispositivo simulado com o Node-RED

## 3.5. Configurando a comunicação MQTT.

- Adicione a autenticação: No campo “Username”, escreva “use-auth-token”. No campo “Password”, use o token gerado automaticamente no momento da criação do device.

The screenshot shows the Node-RED interface with a flow titled "Flow 1". On the left, there's a sidebar with categories like "input" (inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot) and "output" (debug, link). The main canvas has nodes for mqtt, http, and ibmiot. A modal window titled "Edit mqtt out node > Edit mqtt-broker node" is open, showing the "Security" tab. It has fields for "Username" (set to "use-auth-token") and "Password" (represented by a redacted string of dots). The "Connection" and "Messages" tabs are also visible. To the right of the modal, the "info" tab displays node details: "Node" is "12dc1008.03a508", "Type" is "mqtt-broker", and "show more" is available. Below the info tab, "Node Help" provides instructions for configuration. At the bottom of the modal, it says "3 nodes use this config" and "On all flows".

# 3. Criando um dispositivo simulado com o Node-RED

## 3.5. Configurando a comunicação MQTT.

- Adicione a autenticação: No campo “Username”, escreva “use-auth-token”. No campo “Password”, use o token gerado automaticamente no momento da criação do device.

- Clique no botão “Update”.

The screenshot shows the Node-RED interface on a web browser. The URL is <https://lab-iot-jppp.mybluemix.net/red/#flow/81147366.5465a8>. The main area displays a flow titled "Flow 1" with various nodes like inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot, debug, and led-on/off. On the right, a modal window titled "Edit mqtt out node > Edit mqtt-broker node" is open, showing configuration for a connection. The "Security" tab is selected, displaying fields for "Username" (set to "use-auth-token") and "Password" (represented by a series of dots). A red box highlights the "Update" button at the top right of the modal. To the right of the modal, the "info" tab of the sidebar shows details for the node, including its ID ("#12dc1008.03a508"), type ("mqtt-broker"), and a note about generating a random Client ID if none is set. Below the info tab, "Node Help" provides instructions for using the MQTT In and MQTT Out nodes. At the bottom of the modal, it says "3 nodes use this config" and "On all flows".

# 3. Criando um dispositivo simulado com o Node-RED

## 3.5. Configurando a comunicação MQTT.

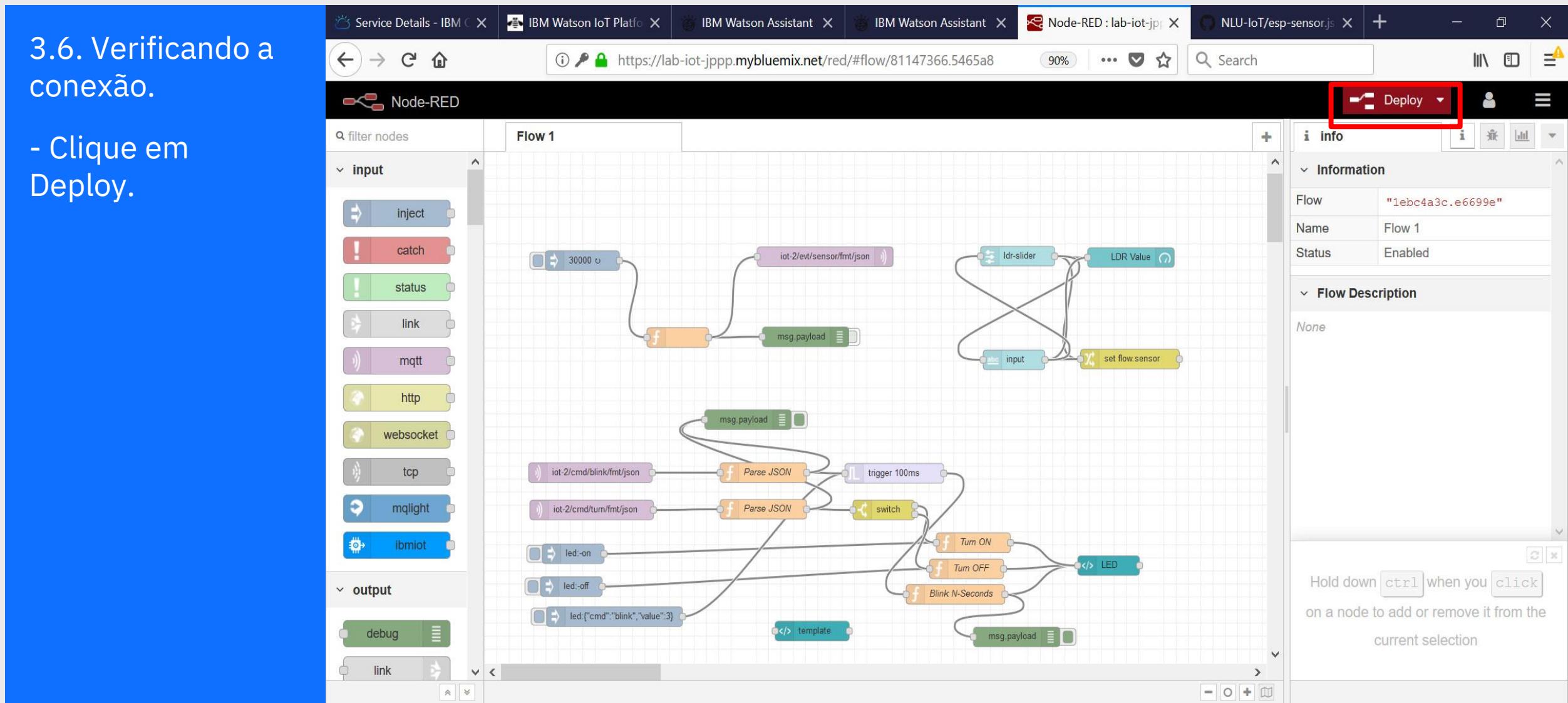
- Verifique o tópico: Deve ser algo parecido com “iot-2/evt/sensor/fmt/json”.
- Clique em “Done”

The screenshot shows the Node-RED interface with a flow titled "Flow 1". The flow consists of several nodes: an "inject" node, a "catch" node, a "status" node, a "link" node, an "mqtt" node, an "http" node, a "websocket" node, a "tcp" node, an "mqlight" node, an "ibmiot" node, a "debug" node, and a "link" node. A central part of the flow involves an "mqtt out" node. The "Topic" field for this node is highlighted with a red box and contains the value "iot-2/evt/sensor/fmt/json". To the right of the flow, a modal dialog titled "Edit mqtt out node" is open, showing the "node properties" section. The "Topic" field here also contains "iot-2/evt/sensor/fmt/json". A red box highlights the "Done" button in the top right corner of this modal. The Node-RED interface has tabs at the top for Service Details, IBM Watson IoT Platfo, IBM Watson Assistant, IBM Watson Assistant, Node-RED, NLU-IoT/esp-sensor.js, and a Deploy button. The right side of the interface includes an "info" panel with "Information" and "Node Help" sections, and a "Inputs" panel for the payload node.

# 3. Criando um dispositivo simulado com o Node-RED

## 3.6. Verificando a conexão.

- Clique em Deploy.

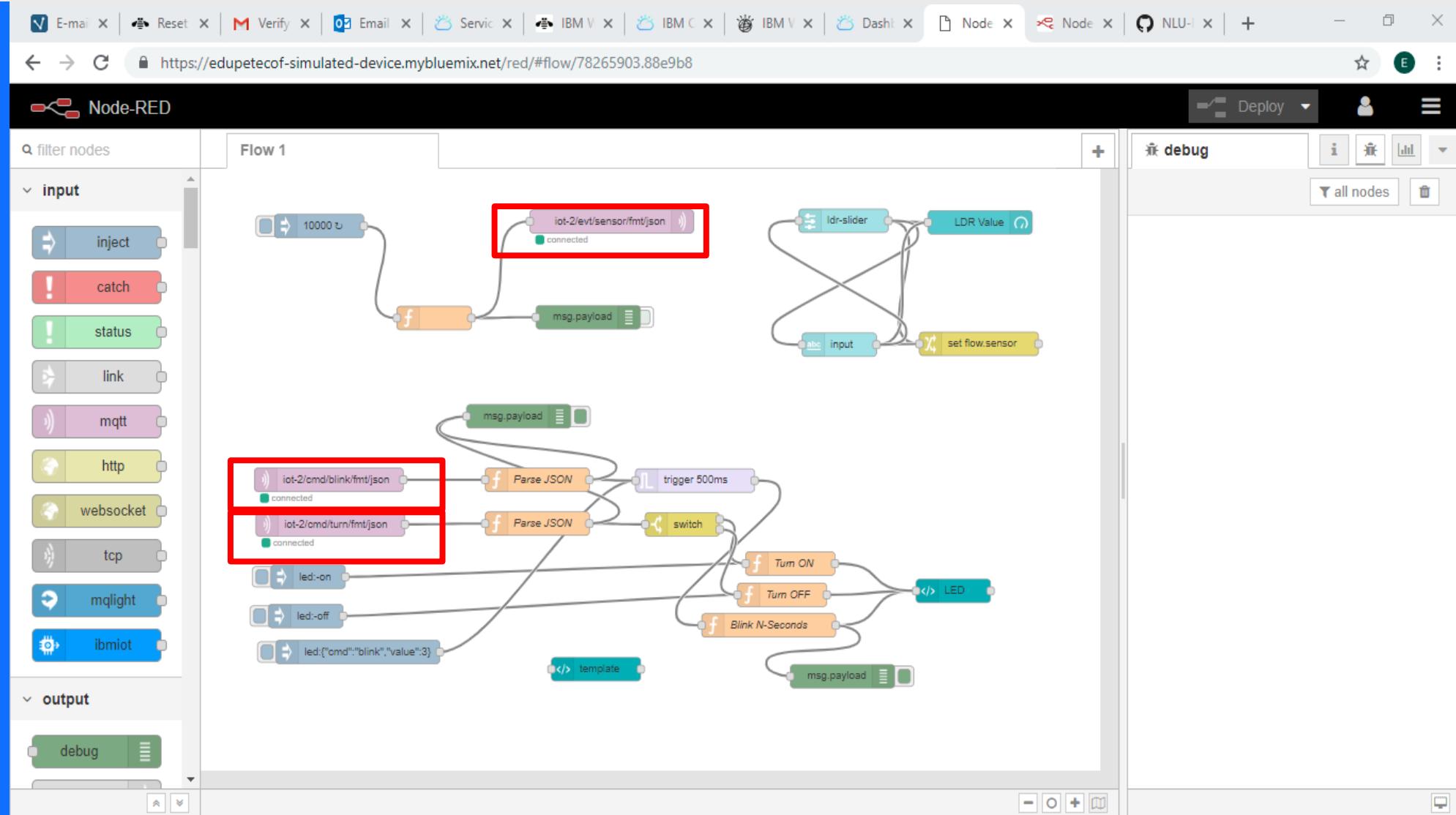


# 3. Criando um dispositivo simulado com o Node-RED

## 3.6. Verificando a conexão.

- Clique em Deploy.

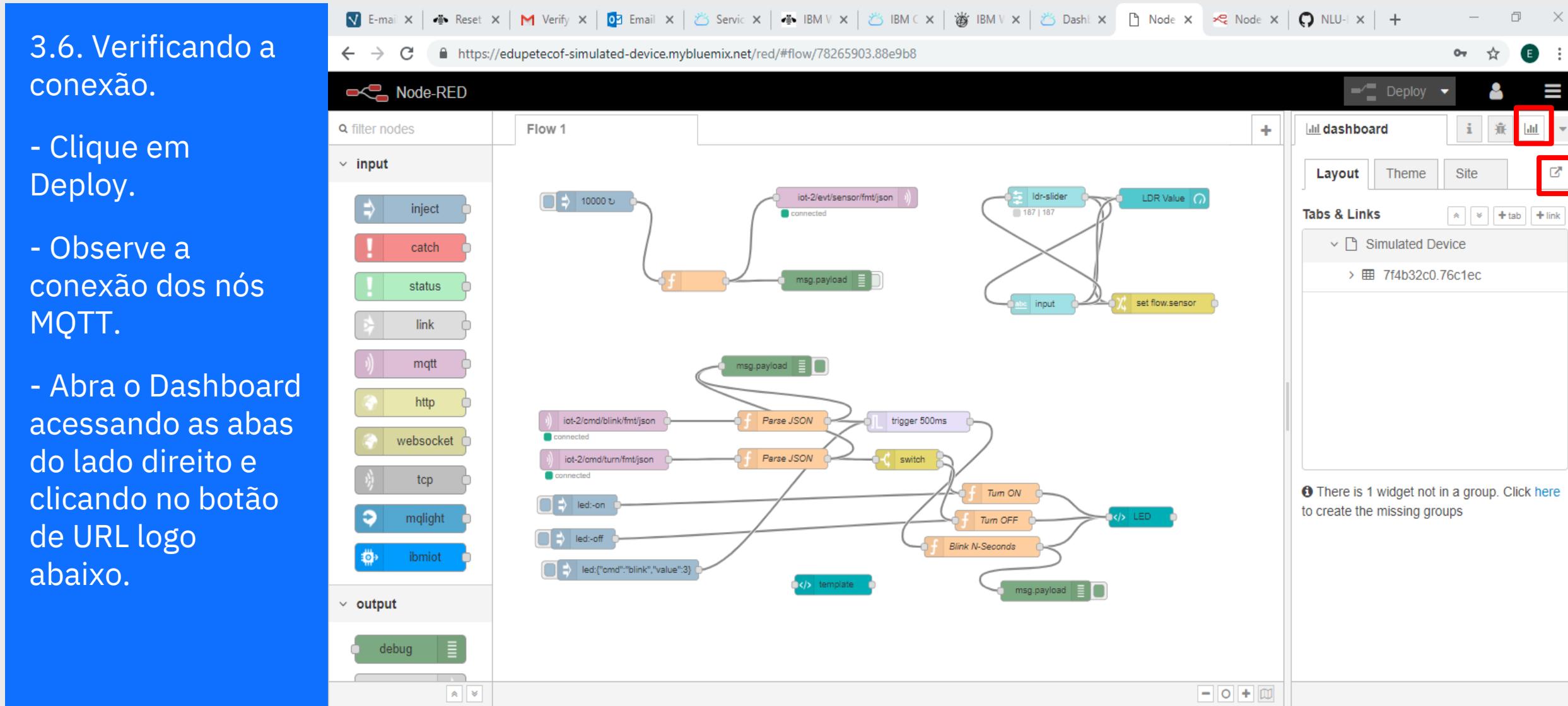
- Observe a conexão dos nós MQTT.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.6. Verificando a conexão.

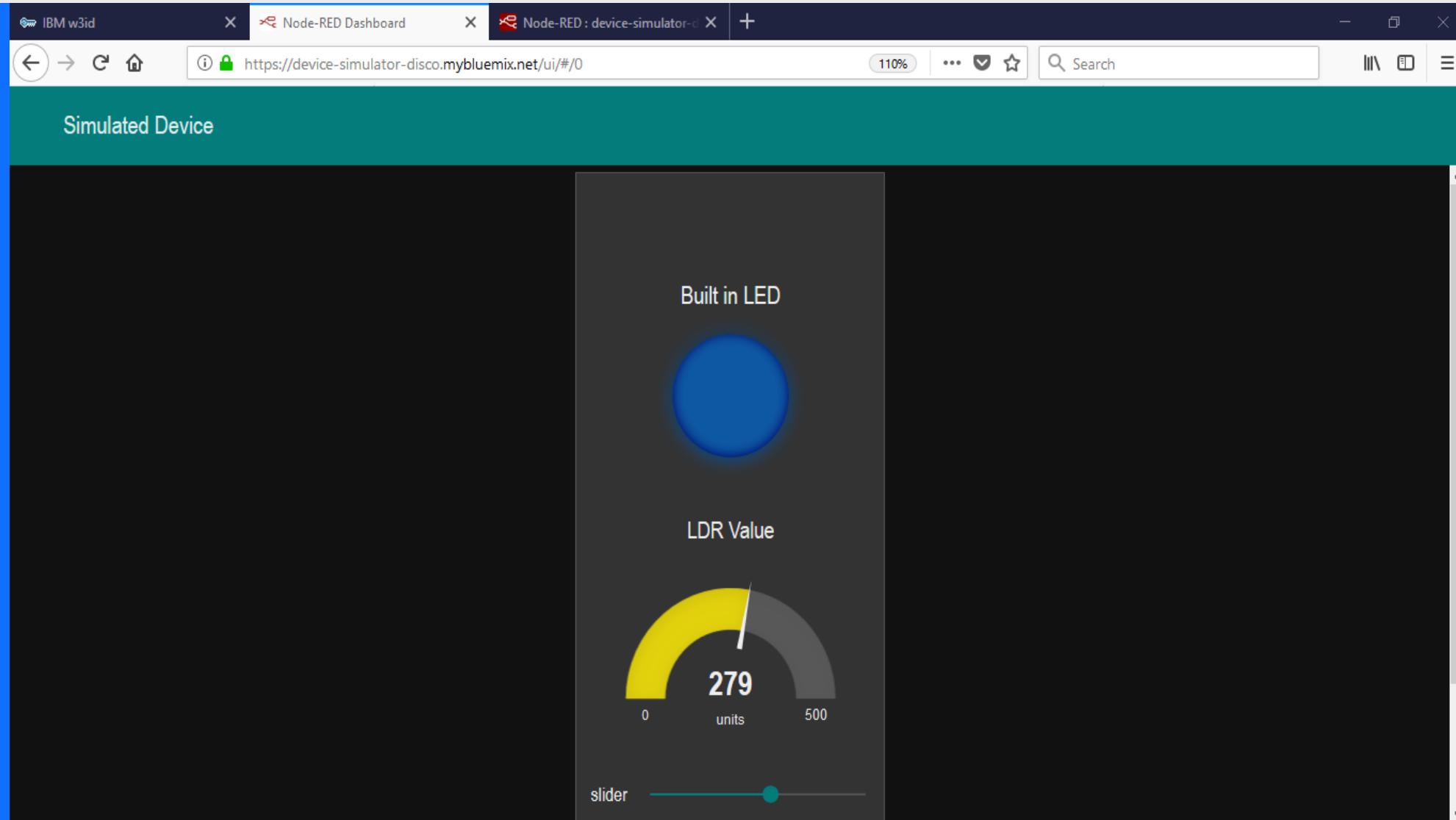
- Clique em Deploy.
- Observe a conexão dos nós MQTT.
- Abra o Dashboard acessando as abas do lado direito e clicando no botão de URL logo abaixo.



### 3. Criando um dispositivo simulado com o Node-RED

#### 3.6. Verificando a conexão.

- Clique em Deploy.
- Observe a conexão dos nós MQTT.
- Abra o Dashboard acessando as abas do lado direito e clicando no botão de URL abaixo.
- Mova o slider para mudar o valor do sensor.



# 3. Criando um dispositivo simulado com o Node-RED

## 3.6. Verificando a conexão.

- Verifique a conexão na plataforma de IoT.

The screenshot shows the 'IBM Watson IoT Platform' dashboard with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2>. The top navigation bar includes links for E-mail, Reset, Verify, Email, Service, IBM V, IBM C, IBM V, IBM V, Dash, Node, Node, NLU-, and a plus sign. The user is logged in as eduardo.petecof@aluno.ufabc.edu.br with ID: 7y60pl. The main page title is 'IBM Watson IoT Platform' and the sub-page title is 'Browse Devices'. On the left, there is a vertical sidebar with icons for Home, Device Types, Device ID, Device Type, Class ID, Date Added, Delete, Filter, Previous, and Next. The main content area displays a table titled 'Browse Devices' with the following columns: Device ID, Device Type, Class ID, and Date Added. A search bar at the top right says 'Type the Device ID to search for'. The table shows one result: 'my-simulated-device' (Device ID), 'Simulated-Device' (Device Type), 'Device' (Class ID), and 'Sep 30, 2018 4:27 PM' (Date Added). A red box highlights the 'Device ID' column header.

Device ID	Device Type	Class ID	Date Added
my-simulated-device	Simulated-Device	Device	Sep 30, 2018 4:27 PM

# 3. Criando um dispositivo simulado com o Node-RED

## 3.6. Verificando a conexão.

- Verifique a conexão na plataforma de IoT.
- Selecione o Device e veja os eventos chegando na aba “Recent Events”.

The screenshot shows the IBM Watson IoT Platform dashboard. At the top, there are four tabs: "Service Details - IBM Cloud", "IBM Watson IoT Platform", "Node-RED Dashboard", and "Node-RED : device-simulator-". The "IBM Watson IoT Platform" tab is active, displaying the "Browse" section. A sidebar on the left contains icons for various device types. The main area shows a table with columns: Device ID, Device Type, Class ID, and Date Added. One row is selected, showing "teste\_sim" as the Device ID, "Simulated" as the Device Type, "Device" as the Class ID, and "Aug 28, 2018 6:19 PM" as the Date Added. Below the table, there are tabs for Identity, Device Information, Recent Events, State, and Logs. The "Recent Events" tab is selected. It displays a table with columns: Event, Value, Format, and Last Received. Two rows are listed, both labeled "sensor". The first row has a value of "[{"d": [{"counter": 360000, "sensor": 279}]}]" and was received "a few seconds ago". The second row has a value of "[{"d": [{"counter": 330000, "sensor": 279}]}]" and was received "a few seconds ago". The "Value" column for the second row is highlighted with a red box. On the right side of the dashboard, there is a small icon of a gear with three dots.

Event	Value	Format	Last Received
sensor	[{"d": [{"counter": 360000, "sensor": 279}]}]	json	a few seconds ago
sensor	[{"d": [{"counter": 330000, "sensor": 279}]}]	json	a few seconds ago

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.

The screenshot shows the IBM Cloud dashboard interface. At the top, there are three tabs: 'Dashboard - IBM Cloud', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area is titled 'Dashboard' and includes filters for 'RESOURCE GROUP', 'CLOUD FOUNDRY ORG', 'CLOUD FOUNDRY SPACE', 'LOCATION', and 'CATEGORY'. A prominent red box highlights the 'Create resource' button in the top right corner. Below this, a message box says 'Learn more about migrating your eligible service instances to resource groups.' Under 'Cloud Foundry Applications', there is one entry: 'Lab-IoT-JPPP' in 'US South' with 'JoaoPedroPP52...' as the CF Org, 'dev' as the CF Space, 256 MB of memory, and a green status icon indicating it's 'Running (1/1)'. The 'Clusters' section shows one cluster named 'mycluster' in 'US South' with 1 node, Kube version 1.10.7\_1520, and a green status icon indicating it's 'Normal'. The bottom section, 'Cloud Foundry Services', is currently empty. On the far right, there is a vertical 'FEEDBACK' button.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.

The screenshot shows the IBM Cloud Catalog interface. At the top, there are three tabs: "Catalog - IBM Cloud", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.myblu". The URL in the address bar is <https://console.bluemix.net/catalog/?search=functions>. The main content area is titled "Catalog" and has a search bar with "functions" typed in. On the left, a sidebar lists categories: All Categories (4), Compute, Analytics, Integration, Internet of Things, Security and Identity, Starter Kits, Web and Mobile, and Web and Application (1). The "Compute" category is expanded, showing sub-categories: Compute (1), Containers, Networking, Storage, AI, Analytics (1), Databases (1), Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits, Web and Mobile, and Web and Application (1). The "Compute" section contains a card for "Functions" which is highlighted with a red box. The card includes the IBM logo, a green "f" icon, the text "Functions", and a brief description: "IBM Cloud Functions is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events." Below this, the "Analytics" section is partially visible, featuring the "Db2 Warehouse" service.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.
- Clique no botão “Start Creating”.

The screenshot shows the IBM Cloud Functions "Getting Started" page. The URL in the browser is [https://console.bluemix.net/openwhisk/?bss\\_account=1dfcc0aea](https://console.bluemix.net/openwhisk/?bss_account=1dfcc0aea). The left sidebar has a "Functions" section with "Overview" selected, and other options like "Pricing", "Concepts", "Integrations", "CLI", "iOS SDK", "API Key", "Documentation", "Actions", "Triggers", "Monitor", "Logs", and "APIs". The main content area features a large "Getting Started with IBM Cloud Functions" heading, a brief description of the service, and two buttons: "Start Creating" (which is highlighted with a red box) and "Download CLI". Below these are sections for "What's New" and "Recent Changes". A large graphic on the right shows a green "f" icon inside a circle with various connectivity symbols like arrows and stars.

Getting Started with  
IBM Cloud Functions

IBM Cloud Functions (based on Apache OpenWhisk) is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events and costs nothing when not in use. [Learn More](#)

[Start Creating](#) [Download CLI](#)

★ What's New:

- Updated Action runtimes: NodeJS 8, Swift 4
- Get started quickly with Templates: [Try it now](#)
- New logging service integration (see the [Logs](#) link in the left navigation)

★ Recent Changes:

- Compliance: New ISO certifications ([ISO 27001](#), [ISO 27017](#), [ISO 27018](#))
- Available in new datacenter: Frankfurt, Germany (see the [Region selector](#))

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.
- Clique no botão “Start Creating”.
- Caso seu Space não seja encontrado. Mude a região.

The screenshot shows a web browser window with three tabs at the top: "IBM Cloud Functions - Create", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.myblu". The main content area is titled "IBM Cloud" and has a "Functions" section. A modal dialog box is open in the center, displaying the message: "No Cloud Foundry Space". Below this, it says "Current Namespace:" and provides instructions: "Before you can proceed with Cloud Functions, you must create a Cloud Foundry space. [Do this now](#)". The URL in the address bar is https://console.bluemix.net/openwhisk/create.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.
- Clique no botão “Start Creating”.
- Caso seu Space não seja encontrado. Mude a região.

The screenshot shows a browser window with several tabs open: 'IBM Cloud Functions - Create', 'IBM Cloud Account', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area is titled 'IBM Cloud' with tabs for 'Catalog', 'Docs', 'Support', and 'Manage'. A search bar says 'Search for resource...'. On the left, there's a sidebar with 'Functions' selected, and options like 'Getting Started', 'Actions', 'Triggers', 'Monitor', 'Logs', and 'APIs'. The main panel has sections for 'REGION', 'CLOUD FOUNDRY ORG', and 'CLOUD FOUNDRY SPACE'. Under 'REGION', 'Germany' is selected. A dropdown menu lists 'Germany', 'United Kingdom', 'US East', and 'US South'. The 'US South' option is highlighted with a red rectangle. At the bottom of the page, there's a footer with 'javascript:void(0)' and a back arrow.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Clique em “Create Action”.

The screenshot shows the 'IBM Cloud Functions - Create' page in a browser. The URL is <https://console.bluemix.net/openwhisk/create>. The page title is 'Create'. On the left, there's a sidebar with 'Functions' selected, and options like 'Getting Started', 'Actions', 'Triggers', 'Monitor', 'Logs', and 'APIs'. The main area has sections for 'Quickstart Templates', 'Create Action', 'Create Trigger', and 'Install Packages'. The 'Create Action' section is highlighted with a red box. It contains text: 'Actions contain your function code and are invoked by events or REST API calls.' To the right, there's another section for 'Create Sequence'. At the bottom, there's a 'Cancel' button.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Clique em “Create Action”.

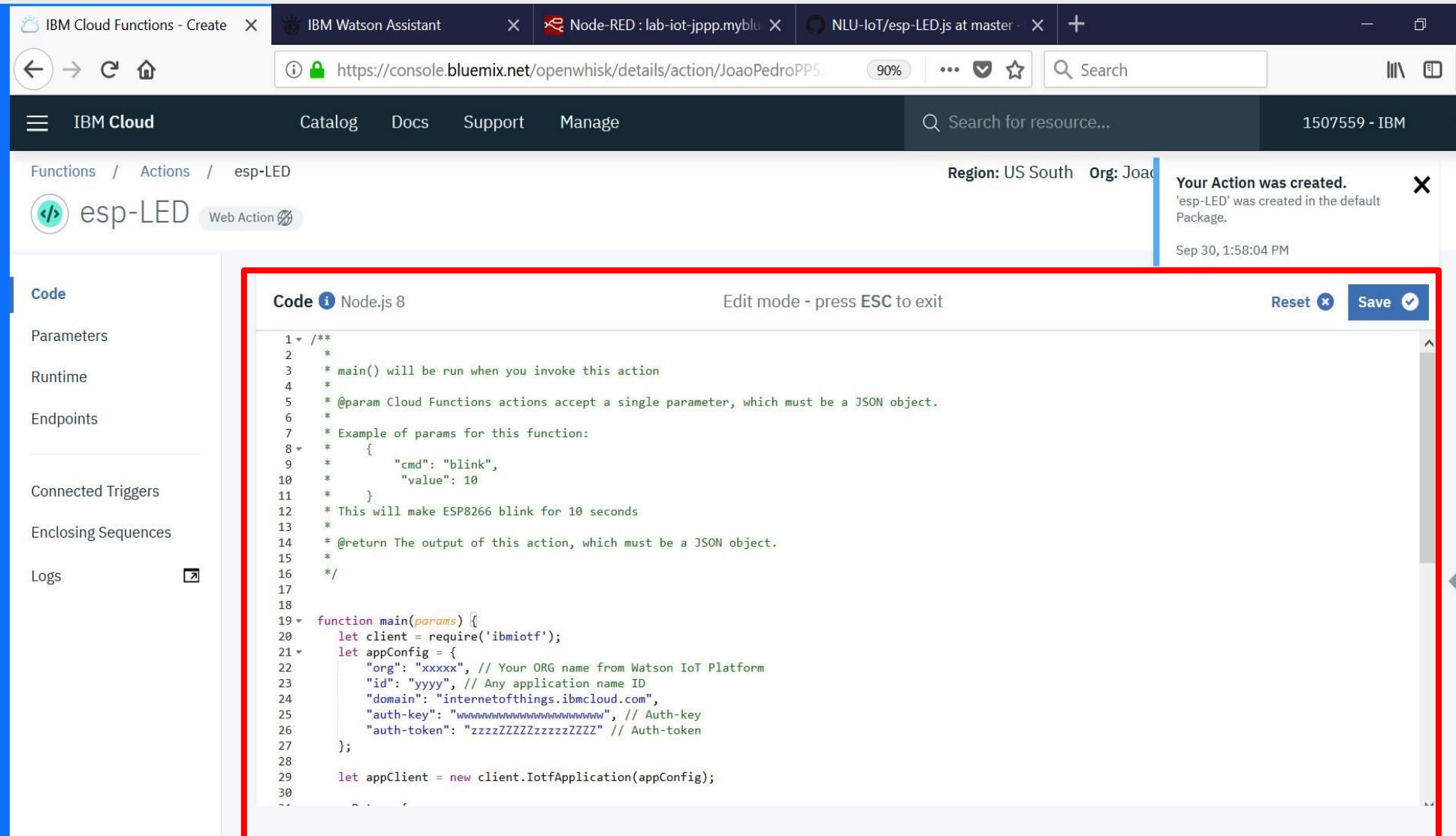
- Dê um nome para a sua primeira Function. Ela será usada para controlar o LED do seu dispositivo.

The screenshot shows the 'Create Action' page in the IBM Cloud Functions interface. The 'Action Name' field is highlighted with a red box and contains the value 'esp-LED'. The 'Enclosing Package' dropdown is set to '(Default Package)'. The 'Runtime' dropdown is set to 'Node.js 8'. A note at the bottom states: 'Looking for Java or Docker? [Java](#) and [Docker](#) Actions can be created with the [CLI](#)'. Navigation buttons at the bottom include 'Cancel', 'Previous', and 'Create'.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Copie o código do git:  
<https://github.com/Bluemix-Estag/NLU-IoT/blob/master/Functions/ESP8266/esp-LED.js> e cole na caixa de código.



The screenshot shows the IBM Cloud Functions interface with a browser tab for 'IBM Cloud Functions - Create' and other tabs like 'IBM Watson Assistant', 'Node-RED', and 'NLU-IoT/esp-LED.js at master'. The main page displays the 'Actions' section under 'Functions / Actions / esp-LED'. A new action named 'esp-LED' is being created, indicated by a message: 'Your Action was created. 'esp-LED' was created in the default Package.' The timestamp is Sep 30, 1:58:04 PM. On the left, a sidebar lists 'Parameters', 'Runtime', 'Endpoints', 'Connected Triggers', 'Enclosing Sequences', and 'Logs'. The 'Code' tab is selected, showing the Node.js 8 code for the 'main' function:

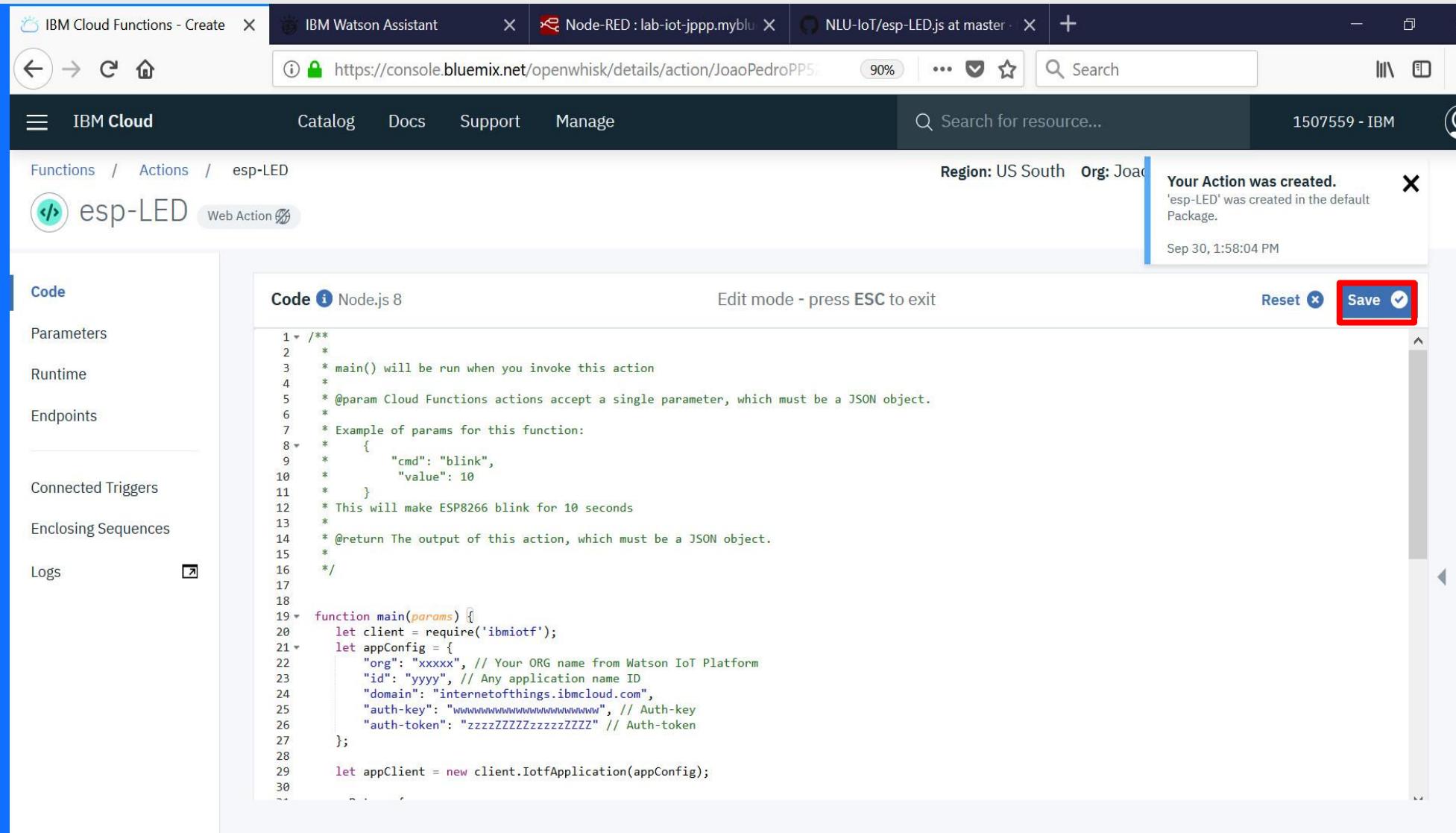
```
1  /**
2  * main() will be run when you invoke this action
3  *
4  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
5  *
6  * Example of params for this function:
7  * {
8  *   "cmd": "blink",
9  *   "value": 10
10 * }
11 *
12 * This will make ESP8266 blink for 10 seconds
13 *
14 * @return The output of this action, which must be a JSON object.
15 *
16 */
17
18
19 function main(params) {
20   let client = require('ibmiotf');
21   let appConfig = {
22     "org": "xxxxx", // Your ORG name from Watson IoT Platform
23     "id": "yyyy", // Any application name ID
24     "domain": "internetofthings.ibmcloud.com",
25     "auth-key": "xxxxxxxxxxxxxxxxxxxx", // Auth-key
26     "auth-token": "zzzzZZZZzzzzZZZ" // Auth-token
27   };
28
29   let appClient = new client.IotfApplication(appConfig);
30 }
```

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.1. Criando as funções.

- Copie o código do git:  
<https://github.com/Bluemix-Estag/NLU-IoT/blob/master/Functions/ESP8266/esp-LED.js> e cole na caixa de código.

- Salve o código.



The screenshot shows the IBM Cloud Functions interface with the following details:

- Header:** IBM Cloud Functions - Create, IBM Watson Assistant, Node-RED : lab-iot-jppp.myblu, NLU-IoT/esp-LED.js at master, +
- Breadcrumbs:** Functions / Actions / esp-LED
- Title:** esp-LED (Web Action)
- Code Editor:** Code (Node.js 8)  
The code is a Node.js script for an ESP8266 device, defining a main function that initializes an IBM IoT client and creates an application configuration object with org, id, domain, auth-key, and auth-token parameters.

```
1  /**
2  * main() will be run when you invoke this action
3  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
4  *
5  * Example of params for this function:
6  * {
7  *   "cmd": "blink",
8  *   "value": 10
9  * }
10 * This will make ESP8266 blink for 10 seconds
11 *
12 * @return The output of this action, which must be a JSON object.
13 *
14 */
15
16
17
18
19 function main(params) {
20   let client = require('ibmiotf');
21   let appConfig = {
22     "org": "xxxxx", // Your ORG name from Watson IoT Platform
23     "id": "yyyy", // Any application name ID
24     "domain": "internetofthings.ibmcloud.com",
25     "auth-key": "xxxxxxxxxxxxxxxxxxxxxx", // Auth-key
26     "auth-token": "zzzzZZZZzzzzzzzz" // Auth-token
27   };
28
29   let appClient = new client.IotfApplication(appConfig);
```
- Buttons:** Reset, Save (highlighted with a red box)
- Message:** Your Action was created. 'esp-LED' was created in the default Package. Sep 30, 1:58:04 PM

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.2. Testando a primeira função.

- Clique no botão “Change Input”.

The screenshot shows a code editor with a file named 'node.js' containing the following code:

```
ode.js 8

in() will be run when you invoke this action
aram Cloud Functions actions accept a single parameter, which must be a JSON object.
return The output of this action, which must be a JSON object.

on main(params) {
  t client = require('ibmiotf');
  t appConfig = {
    "org": "vrreja",
    "id": "app",
    "domain": "internetofthings.ibmcloud.com",
    "auth-key": "a-vrreja-rb8frn4vwb",
    "auth-token": "pPyhnIVL_62BAj0KtH"

  t appClient = new client.IotfApplication(appConfig);

  Data = {
    "d": {
      "fields": [
        {
          "field": params.cmd,
          "value": params.value
        }
      ]
    }
  }
}

module.exports = main;
```

The 'Change Input' button in the toolbar is highlighted with a red box.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.2. Testando a primeira função.

- Clique no botão “Change Input”.
- Escreva na caixa um JSON como o da imagem ao lado. “cmd” indica o comando desejado, nessa caso vamos fazer o LED piscar. “value” indica por quantos segundos ele permanecerá piscando.

Code Node.js 8

```
1 /**
2  * main() will be run every
3  * @param Cloud Function Request
4  * @return The output
5  */
6
7 function main(params) {
8     let client = req;
9     let appConfig =
10        "org": "vrre",
11        "id": "app",
12        "domain": "i",
13        "auth-key": "auth-token"
14    };
15
16    let appClient =
17
18    myData = {
19        "d": {
20            "fields": [
21                {
22                    "field": params.cmd,
23                    "value": params.value
24                }
25            ]
26        }
27    };
28
29}
```

Change Action Input

Edit mode - press ESC to exit

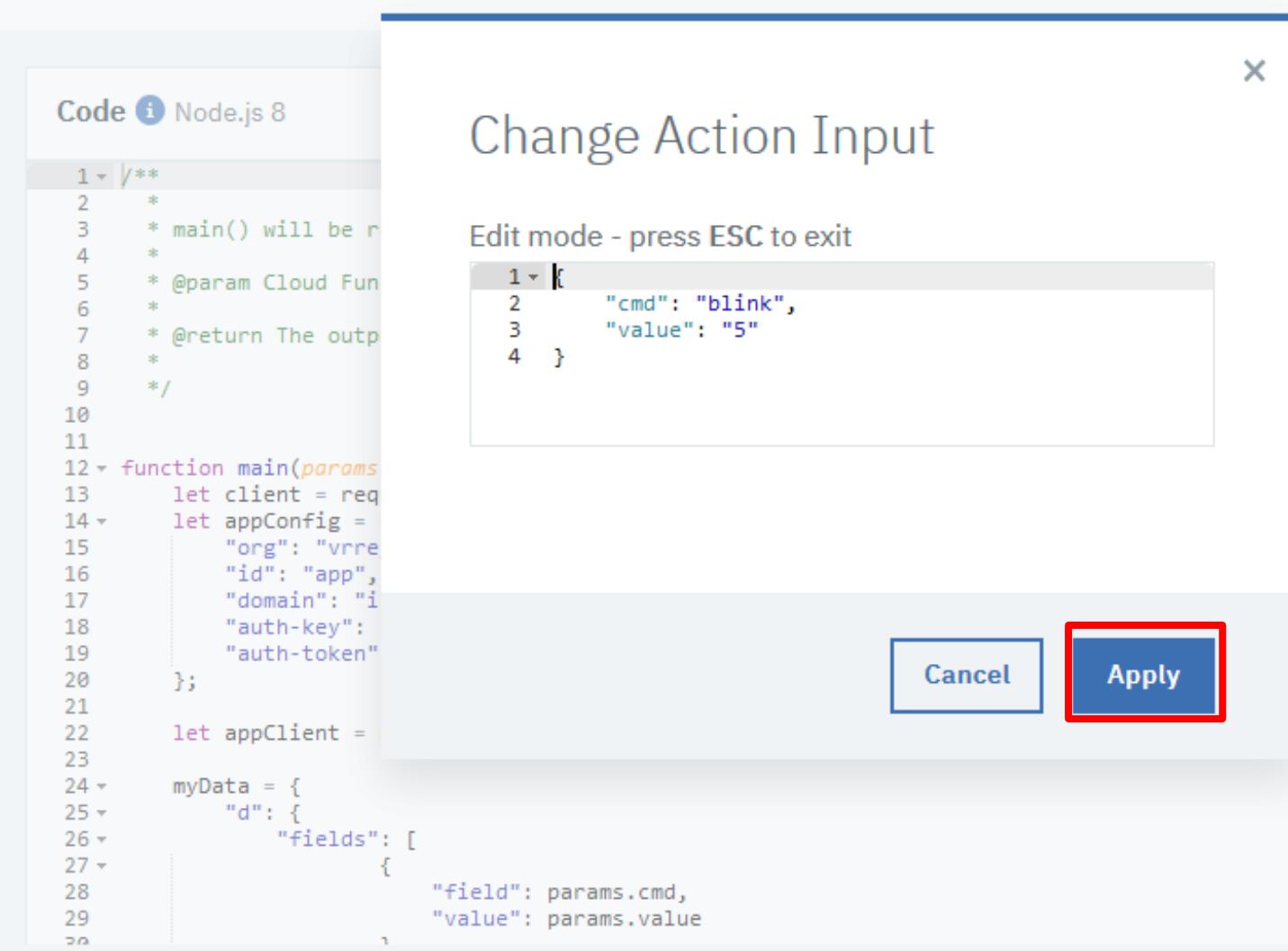
```
{ "cmd": "blink", "value": "5"}
```

Cancel Apply

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.2. Testando a primeira função.

- Clique no botão “Apply”.



# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.2. Testando a primeira função.

- Clique no botão “Apply”.
- Clique no botão “Invoke para executar a sua Action com a entrada definida.



The screenshot shows the IBM Cloud Functions interface. On the right side, there are two buttons: "Change Input" with a pencil icon and "Invoke" with a circular arrow icon. The "Invoke" button is highlighted with a red rectangular box. The main area contains a code editor with the following JavaScript code:

```
const client = require('ibm-watson-iot-client');
const appConfig = require('./appConfig');

let myData = {
    "cmd": "teste",
    "value": "on"
};

let params = {
    "cmd": "teste",
    "value": "on"
};

let ent = new client.IotfApplication(appConfig);

let fields = [
    {
        "field": params.cmd,
        "value": params.value
    }
];

end(callback) {
    pClient.publishDeviceCommand("ESP8266", "teste", params.cmd, "json", myData);
    pClient.publishDeviceCommand("Simulated", "teste_sim", params.cmd, "json", myData);
    callback();
}

let promise = new Promise((resolve, reject) => {
    ent.connect();
    ent.on("connect", function () {
        Data = JSON.stringify(myData);
        end(function () {
            appClient.disconnect();
        });
    });
});
```

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.2. Testando a primeira função.

- Clique no botão “Apply”.
- Clique no botão “Invoke para executar a sua Action com a entrada definida.
- Se a resposta for 200 (HTTP – OK), o LED do seu Device simulado deve piscar.

The screenshot shows the IBM Cloud Functions interface. On the left, there is a code editor window containing Node.js code for sending commands to a device. The code uses the IoT Application Client library to publish commands to devices named "ESP8266" and "Simulated". It also includes logic to handle the connection and disconnect from the device. At the top of the code editor are "Change Input" and "Invoke" buttons. The "Invoke" button is highlighted with a blue border. To the right of the code editor is a panel titled "Activations" which lists two entries:

- triggerBlink** (Success): Activation ID: 6eb979e3bf384cb9b979e3bf382cb92e. Results: { "response": 200 } (This result is highlighted with a red box).
- triggerBlink** (Failure): Activation ID: 5d8c6309f3004ea18c6309f300aea126. Results: (empty array)

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

The screenshot shows a browser window with several tabs open at the top: "IBM Cloud Functions - Create", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.myblu", and "NLU-IoT/esp-LED.js at master". The main content area is the "IBM Cloud" interface, specifically the "Actions" section for a function named "esp-LED". A red box highlights the "Actions" tab in the breadcrumb navigation. On the left, a sidebar lists "Code", "Parameters", "Runtime", "Endpoints", "Connected Triggers", "Enclosing Sequences", and "Logs". The "Code" section displays a Node.js script for the "main" function, which interacts with the Watson IoT Platform. A success message in a modal box states: "Your Action was created. 'esp-LED' was created in the default Package." The timestamp is "Sep 30, 1:58:04 PM".

```
Code i Node.js 8
1 /**
2 *
3 * main() will be run when you invoke this action
4 *
5 * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6 *
7 * Example of params for this function:
8 * {
9 *   "cmd": "blink",
10 *   "value": 10
11 * }
12 * This will make ESP8266 blink for 10 seconds
13 *
14 * @return The output of this action, which must be a JSON object.
15 *
16 */
17
18
19 function main(params) {
20   let client = require('ibmiotf');
21   let appConfig = {
22     "org": "xxxxx", // Your ORG name from Watson IoT Platform
23     "id": "yyyy", // Any application name ID
24     "domain": "internetofthings.ibmcloud.com",
25     "auth-key": "xxxxxxxxxxxxxxxxxxxxxx", // Auth-key
26     "auth-token": "zzzzZZZZzzzzzzzz" // Auth-token
27   };
28
29   let appClient = new client.IotfApplication(appConfig);
30 }
```

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

- Clique no botão “Create” para criar outra Cloud Function.

The screenshot shows the IBM Cloud Functions Actions page. The URL in the browser is https://console.bluemix.net/openwhisk/actions. The page displays two Cloud Functions:

NAME	RUNTIME	WEB ACTION	MEMORY	TIMEOUT
esp-LED	Node.js 8	Not Enabled	256 MB	60 s
omega	Node.js 8	Not Enabled	256 MB	60 s

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

- Clique no botão “Create” para criar outra Cloud Function.

- Clique em “Create Action”

The screenshot shows the 'Create' page for IBM Cloud Functions. The URL in the browser is <https://console.bluemix.net/openwhisk/create>. The page has a sidebar on the left with 'Functions' selected, and a main area with the heading 'Create'. There are several options:

- Quickstart Templates**: Get started quickly using one of the Templates. A number of use cases are available, from a hello world action to invoking functions from Cloudant or Message Hub events.
- Create Action**: Actions contain your function code and are invoked by events or REST API calls. This option is highlighted with a red box.
- Create Sequence**: Sequences invoke Actions in a linear order, passing parameters from one to the next.
- Create Trigger**: Triggers receive events from outside IBM Cloud Functions and invoke all connected Actions.
- Install Packages**: Installing Packages installs reusable Actions into your namespace.

At the bottom of the page is a 'Cancel' button.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

- Clique no botão “Create” para criar outra Cloud Function.

- Clique em “Create Action”

- Dê um nome para sua outra Function.

The screenshot shows the 'Create Action' interface in the IBM Cloud Functions service. The left sidebar is titled 'Functions' and includes links for Getting Started, Actions (which is currently selected), Triggers, Monitor, Logs, and APIs. The main form has the following fields:

- Action Name:** A text input field containing "esp-sensor" is highlighted with a red border.
- Enclosing Package:** A dropdown menu set to "(Default Package)" with a "Create Package" button next to it.
- Runtime:** A dropdown menu set to "Node.js 8".

Below the form, a note states: "Looking for Java or Docker? [Java](#) and [Docker](#) Actions can be created with the [CLI](#)". At the bottom of the page are "Cancel", "Previous", and "Create" buttons.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

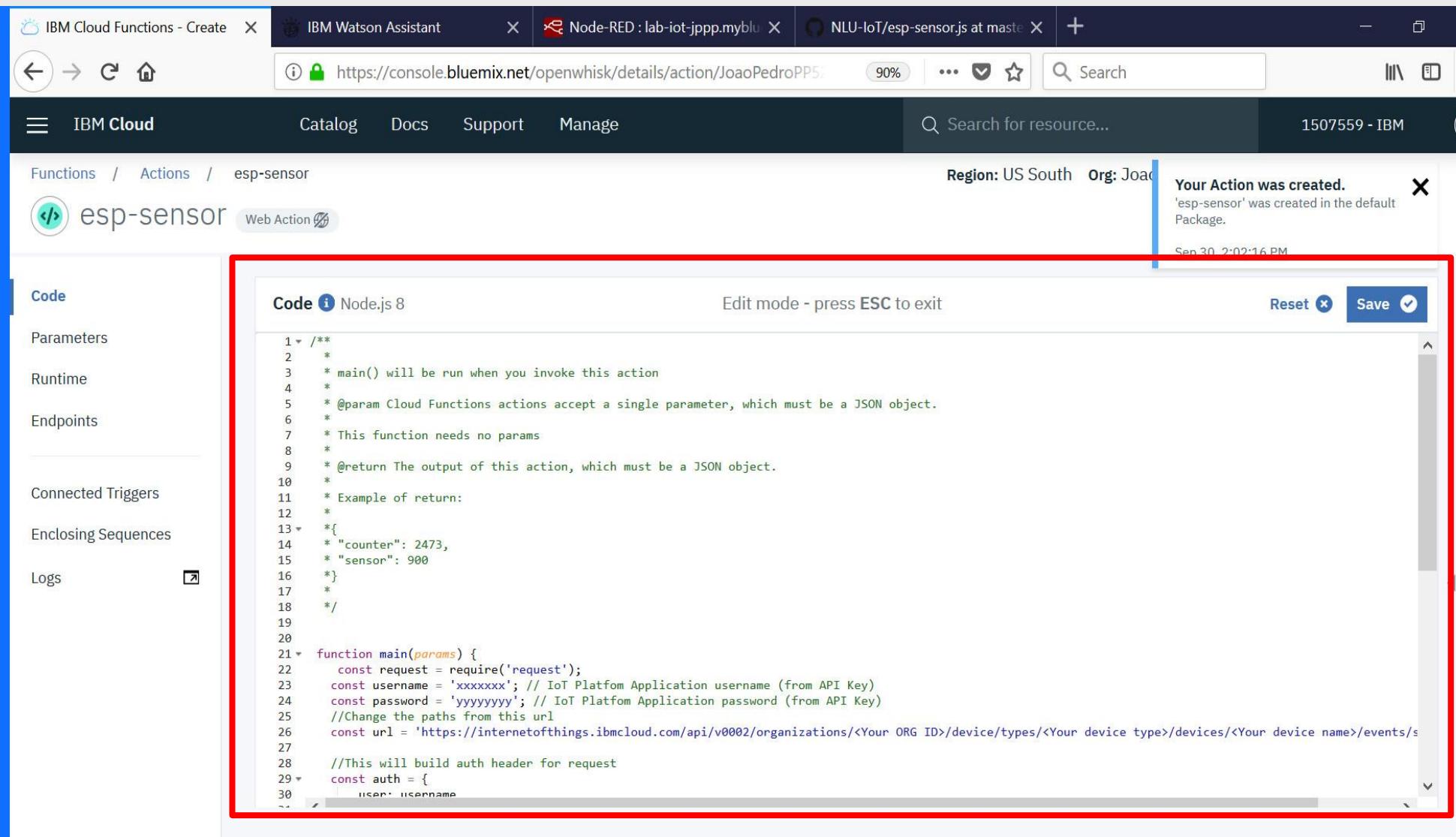
- Clique no botão “Create”.

The screenshot shows the 'Create Action' interface in the IBM Cloud Functions service. The left sidebar lists 'Getting Started', 'Actions', 'Triggers', 'Monitor', 'Logs', and 'APIs'. The main area has tabs for 'REGION' (US South), 'CLOUD FOUNDRY ORG' (JoaoPedroPP52ORG), and 'CLOUD FOUNDRY SPACE' (dev). The title 'Create Action' is centered above the form fields. The 'Action Name' field contains 'esp-sensor'. The 'Enclosing Package' dropdown is set to '(Default Package)' and has a 'Create Package' button next to it. The 'Runtime' dropdown is set to 'Node.js 8'. A note at the bottom says 'Looking for Java or Docker? [Java](#) and [Docker](#) Actions can be created with the [CLI](#)'. At the bottom are 'Cancel', 'Previous', and 'Create' buttons, with 'Create' being highlighted with a red box.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

- Clique no botão “Create”.
- Copie o código do git:  
<https://github.com/Bluemix-Estag/NLU-IoT/blob/master/Functions/ESP8266/esp-sensor.js> e cole na caixa de código.



The screenshot shows the IBM Cloud Functions - Create interface. The browser tabs include 'IBM Cloud Functions - Create', 'IBM Watson Assistant', 'Node-RED : lab-iot-jppp.myblu...', and 'NLU-IoT/esp-sensor.js at master'. The main page displays the 'esp-sensor' action under the 'Actions' category. The 'Code' tab is selected, showing a Node.js script. A red box highlights the code area. The code is as follows:

```
1  /**
2  * main() will be run when you invoke this action
3  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
4  * @return The output of this action, which must be a JSON object.
5  * Example of return:
6  * {
7  *   "counter": 2473,
8  *   "sensor": 900
9  * }
10 */
11
12
13 function main(params) {
14   const request = require('request');
15   const username = 'xxxxxxxx'; // IoT Platform Application username (from API Key)
16   const password = 'yyyyyyyy'; // IoT Platform Application password (from API Key)
17   //Change the paths from this url
18   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/sensor';
19
20   //This will build auth header for request
21   const auth = {
22     user: username
23   }
24
25   request({
26     url: url,
27     method: 'POST',
28     auth: auth
29   }, function(error, response, body) {
30     if (!error && response.statusCode === 200) {
31       console.log('Success');
32     } else {
33       console.error('Error:', error);
34     }
35   });
36 }
```

A message box on the right says 'Your Action was created. 'esp-sensor' was created in the default Package.' with a timestamp of 'Sep 30, 2:02:16 PM'.

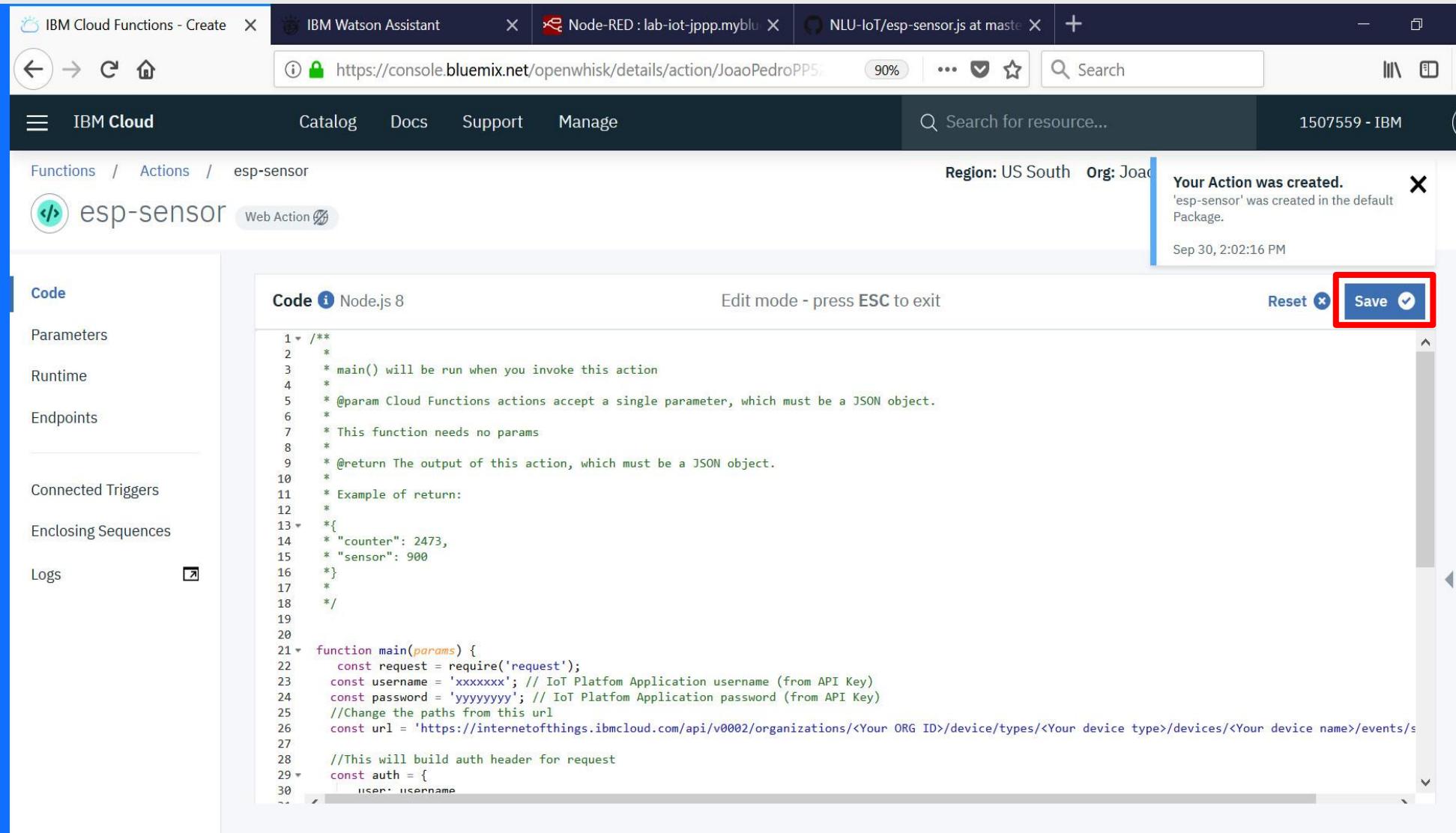
# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.3. Criando a segunda função.

- Clique no botão “Create”.

- Copie o código do git:  
<https://github.com/Bluemix-Estag/NLU-IoT/blob/master/Functions/ESP8266/esp-sensor.js> e cole na caixa de código.

- Salve o código.



The screenshot shows the IBM Cloud Functions - Create interface. The title bar has tabs for 'IBM Cloud Functions - Create', 'IBM Watson Assistant', 'Node-RED : lab-iot-jppp.myblu...', and 'NLU-IoT/esp-sensor.js at master'. The main area shows the 'Actions' section with a 'Functions / Actions / esp-sensor' path. A 'esp-sensor' action is selected, described as a 'Web Action'. On the left, a sidebar lists 'Code', 'Parameters', 'Runtime', 'Endpoints', 'Connected Triggers', 'Enclosing Sequences', and 'Logs'. The 'Code' tab is active, displaying a Node.js script. The script starts with a multi-line comment explaining the function's purpose and parameters. It then defines a 'main' function that requires the 'request' module, sets up authentication variables ('username' and 'password'), and constructs a URL for an IoT API call. A success message 'Your Action was created.' is displayed on the right, stating that 'esp-sensor' was created in the default Package on Sep 30, 2:02:16 PM. The 'Save' button is highlighted with a red box.

```
1 /**
2  *
3  * main() will be run when you invoke this action
4  *
5  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6  *
7  * This function needs no params
8  *
9  * @return The output of this action, which must be a JSON object.
10 *
11 * Example of return:
12 *
13 * {
14 *   "counter": 2473,
15 *   "sensor": 900
16 * }
17 *
18 */
19
20
21 function main(params) {
22   const request = require('request');
23   const username = 'xxxxxxxx'; // IoT Platform Application username (from API Key)
24   const password = 'yyyyyyyy'; // IoT Platform Application password (from API Key)
25   //Change the paths from this url
26   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/s
27
28   //This will build auth header for request
29   const auth = {
30     user: username
31   }
32 }
```

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.4. Obtendo as credenciais para uso das Functions.

- Clique em “Endpoints”.

The screenshot shows the IBM Cloud Functions interface. The URL in the browser is <https://console.bluemix.net/openwhisk/details/action/JoaPedroPP5>. The page title is "IBM Cloud Functions - Create". The left sidebar has a red box around the "Endpoints" tab, which is currently selected. The main content area shows the code for a Node.js action named "esp-sensor". The code is as follows:

```
1  /**
2  *
3  * main() will be run when you invoke this action
4  *
5  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6  *
7  * This function needs no params
8  *
9  * @return The output of this action, which must be a JSON object.
10 *
11 * Example of return:
12 *
13 * {
14 *   "counter": 2473,
15 *   "sensor": 900
16 * }
17 *
18 */
19
20
21 function main(params) {
22   const request = require('request');
23   const username = 'xxxxxxxx'; // IoT Platform Application username (from API Key)
24   const password = 'yyyyyyyy'; // IoT Platform Application password (from API Key)
25   //Change the paths from this url
26   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/sensor';
27
28   //This will build auth header for request
29   const auth = {
30     user: username
31   }
32
33   const options = {
34     url: url,
35     method: 'POST',
36     headers: {
37       'Content-Type': 'application/json'
38     },
39     auth: auth
40   }
41
42   request(options, (error, response, body) => {
43     if (error) {
44       console.error(error);
45       return;
46     }
47     console.log(`Response: ${body}`);
48   });
49 }
```

A message box on the right says "Your Action was created. 'esp-sensor' was created in the default Package." with a timestamp of "Sep 30, 2:02:16 PM". There are "Change Input" and "Invoke" buttons at the bottom right.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.4. Obtendo as credenciais para uso das Functions.

- Clique em “Endpoints”.

- Na seção CURL, revele a autorização utilizada para uso da sua Function clicando no ícone semelhante a um olho.

IBM Cloud Functions - Create X IBM Watson Assistant X Node-RED : lab-iot-jppp.myblu X NLU-IoT/esp-sensor.js at master X +

IBM Cloud Catalog Docs Support Manage Search for resource... 1507559 - IBM

Functions / Actions / esp-sensor Region: US South Org: JoaoPedroPP52ORG Space: dev

esp-sensor Web Action

Code Parameters Runtime Endpoints Connected Triggers Enclosing Sequences Logs

**Web Action**

Enable as Web Action Allow your Cloud Functions actions to handle HTTP events. Learn more about [Web Actions](#).

Raw HTTP handling When enabled your Action receives requests in plain text instead of a JSON body

**REST API**

HTTP METHOD	AUTH	URL
POST	<a href="#">API-KEY</a>	https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaoPedroPP52ORG_dev/actions/esp-sensor

**CURL**

```
curl -u API-KEY -X POST https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaoPedroPP52ORG_dev/actions/esp-sensor?blocking=true
```

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.4. Obtendo as credenciais para uso das Functions.

- Copie para o Note Pad toda a chamada CURL.
- As credenciais estão divididas em duas partes. A primeira (vermelho) é o “username” e depois dos dois pontos temos o “password”.

The screenshot shows the IBM Cloud Functions - Action interface. The URL in the browser is https://console.bluemix.net/openwhisk/details/action/JoaPedroPP52ORG\_dev/esp-sensor. The page displays the configuration for the 'esp-sensor' action, specifically the 'Web Action' and 'REST API' sections. The 'Endpoints' tab is selected in the sidebar. In the 'Web Action' section, there are two checkboxes: 'Enable as Web Action' (unchecked) and 'Raw HTTP handling' (unchecked). In the 'REST API' section, the 'HTTP METHOD' is set to 'POST', 'AUTH' is set to 'API-KEY', and the 'URL' is https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaPedroPP52ORG\_dev/actions/esp-sensor. Below this, a 'CURL' section shows the command: curl -u [REDACTED]:[REDACTED] POST https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaPedroPP52ORG\_dev/actions/esp-sensor?blocking=true. The entire URL part after the colon in the password is highlighted in yellow.

# 4. Enviando comandos ao seu Device usando Cloud Functions

## 4.4. Obtendo as credenciais para uso das Functions.

- Veja que o seu Namespace também aparece na chamada. Vamos usar as credenciais e esse Namespace no nosso próximo serviço.

The screenshot shows the IBM Cloud Functions - Action details page. The URL in the browser is [https://console.bluemix.net/openwhisk/details/action/JoaopPedroPP520RG\\_dev/esp-sensor](https://console.bluemix.net/openwhisk/details/action/JoaopPedroPP520RG_dev/esp-sensor). The page displays the 'esp-sensor' action configuration. On the left, there is a sidebar with tabs: Code, Parameters, Runtime, Endpoints (which is selected), Connected Triggers, Enclosing Sequences, and Logs. Under the 'Endpoints' tab, it says 'Web Action'. There are two checkboxes: 'Enable as Web Action' (unchecked) and 'Raw HTTP handling' (unchecked). Below this is the 'REST API' section, which includes 'HTTP METHOD' (POST), 'AUTH' (API-KEY), and 'URL' ([https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaopPedroPP520RG\\_dev/actions/esp-sensor](https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaopPedroPP520RG_dev/actions/esp-sensor)). At the bottom, there is a 'CURL' section with the command: `curl -u 4582fe5a-b444-4527-8e35-4247ee96a477:wRnjca5HAvq1XBPCcv40PwvaiF1ch7ttdYm9t7B3nFYM74AEFBHQX5G6zVxm2pq4K -X POST https://openwhisk.ng.bluemix.net/api/v1/namespaces JoaoPedroPP520RG dev/actions/esp-sensor blocking=true`. The word 'JoaoPedroPP520RG' is highlighted with a red box.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.1. Criando uma instância do Watson Assistant.

- Volte ao Dashboard e crie um novo recurso.

The screenshot shows the IBM Cloud dashboard interface. At the top, there are tabs for 'Dashboard - IBM Cloud' and 'Node-RED : lab-iot-jppp.mybluemix.net'. The main content area is titled 'Dashboard' and contains three sections: 'Cloud Foundry Applications', 'Clusters', and 'Cloud Foundry Services'. In the top right corner of the dashboard area, there is a blue button labeled 'Create resource' which is highlighted with a red box. The 'Cloud Foundry Applications' section shows one application named 'Lab-IoT-JPPP' with details: Region US South, CF Org JoaoPedroPP52..., CF Space dev, Memory (MB) 256, and Status Running (1/1). The 'Clusters' section shows one cluster named 'mycluster' with details: Location US South, Nodes 1, Kube version 1.10.7\_1520, and Status Normal. The 'Cloud Foundry Services' section shows one service named 'IoT-Lab' with details: Region US South, CF Org JoaoPedroPP52..., Plan iotf-service-free, Service Offering iotf-service. The bottom right corner of the dashboard has a small 'FEEDBACK' button.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.1. Criando uma instância do Watson Assistant.

- Volte ao Dashboard e crie um novo recurso.

- Na aba de busca digite “watson assistant” e ache o serviço “Watson Assistant”.

The screenshot shows the IBM Cloud Catalog interface. At the top, there are two tabs: 'Catalog - IBM Cloud' and 'Node-RED : lab-iot-jppp.mybluemix.net'. The URL in the address bar is <https://console.bluemix.net/catalog/?search=watson%20assistant>. The page title is 'Catalog'. Below the title, there are navigation links for 'IBM Cloud', 'Catalog', 'Docs', 'Support', and 'Manage', along with a search bar that says 'Search for resource...'. On the right side, there is a user profile icon and the text '1507559 - IBM'. A sidebar on the left lists categories: All Categories (2), Compute, Containers, Networking, Storage, AI (1), Analytics, Databases, Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits (1), Web and Mobile, and Web and Application. The 'AI' category is selected. In the main content area, under 'AI', the 'Watson Assistant (formerly Conversation)' service is listed. It has a purple icon of two speech bubbles. The text reads: 'Watson Assistant (formerly Conversation) Lite • IBM Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents'. This entire section is highlighted with a red rectangular box. Below this, there is a 'Starter Kits' section with a card for 'Watson Assistant Basic'.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.1. Criando uma instância do Watson Assistant.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “watson assistant” e ache o serviço “Watson Assistant”.
- Clique no botão Launch Tool.

The screenshot shows the IBM Cloud interface for managing Watson Assistant services. The left sidebar is titled 'Manage' and lists 'Service credentials', 'Plan', and 'Connections'. The main content area shows a service named 'Watson Assistant-Lab' located in 'US South' with 'Org: JoaoPedroPP520RG' and 'Space: dev'. A central callout box contains the text 'Get started by launching the tool.' with a blue 'Launch tool' button. This 'Launch tool' button is highlighted with a red rectangular box. Below it are links to 'Getting started tutorial' and 'API reference'. To the right of the callout, the text 'Plan: Lite' and 'Upgrade' are visible. The bottom section is titled 'Credentials' and shows fields for 'Url' (set to <https://gateway.watsonplatform.net/assistant/api>), 'Username' (redacted), and 'Password' (redacted). A yellow 'Let's talk' button is located in the bottom right corner of this section.

# 5. Criando o Assistente Virtual com o Watson Assistant

5.2. Fazendo o upload do Workspace da demonstração.

- Entre na aba “Workspaces”.

The screenshot shows a web browser window with three tabs: "IBM Watson Service Page", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.mybluemix.net". The "IBM Watson Assistant" tab is active, displaying the "IBM Watson Assistant" landing page. The navigation bar at the top has two items: "Home" and "Workspaces", with "Workspaces" highlighted by a red box. Below the navigation, the text "Introducing IBM Watson Assistant" is displayed, followed by a subtext about Watson Conversation evolving to simplify virtual assistant creation. A large graphic of a speech bubble with three dots is on the right. The main content area is titled "Three easy steps" and lists three steps: 1. Create intents and entities, 2. Build your dialog, and 3. Test your dialog, each with a "Learn more" link.

IBM Watson Assistant

Introducing

## IBM Watson Assistant

Watson Conversation is evolving to simplify how you build and scale virtual assistants. [See what's new](#)

Three easy steps

Follow these steps to create a virtual assistant.

**1** Create intents and entities  
Determine what your virtual assistant will understand by providing training examples so Watson can learn.  
[Learn more](#)

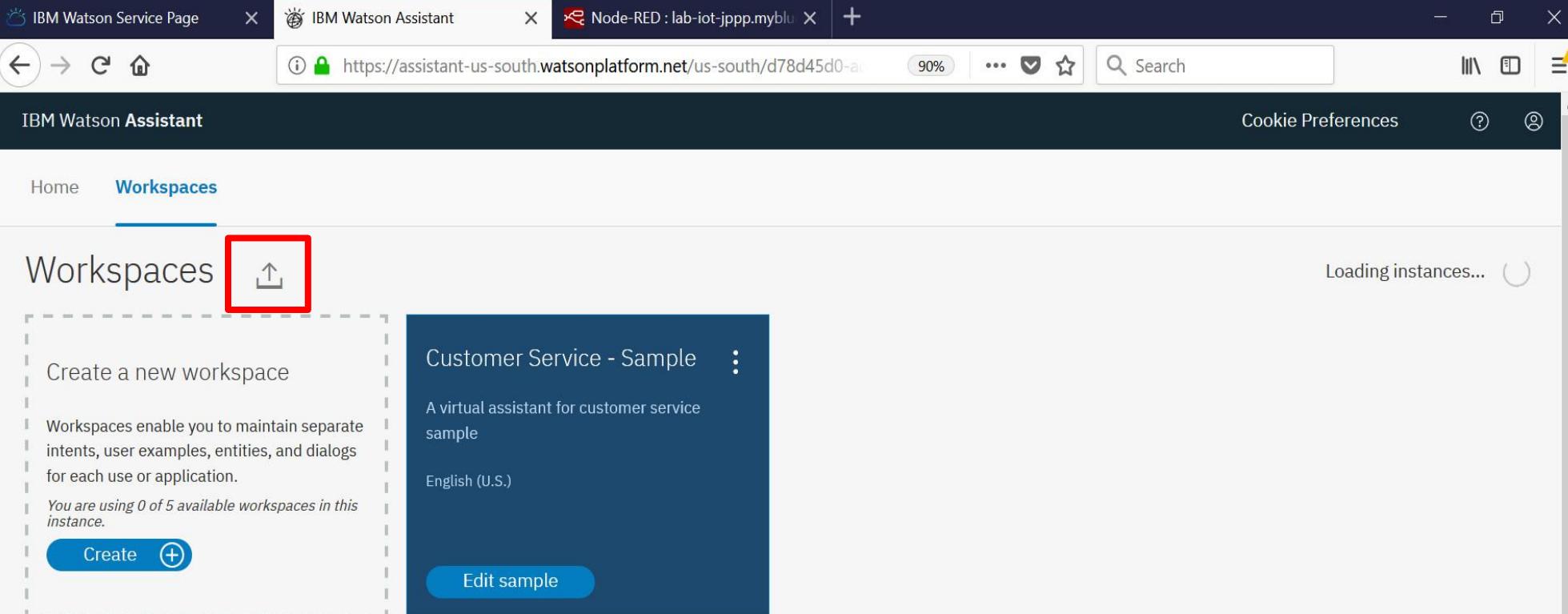
**2** Build your dialog  
Utilize the intents and entities you created, plus context from the application, so your virtual assistant responds appropriately.  
[Learn more](#)

**3** Test your dialog  
Try out the virtual assistant in the tool to see how it recognizes the intents and entities and how it responds firsthand.  
[Learn more](#)

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.2. Fazendo o upload do Workspace da demonstração.

- Salve no seu computador o arquivo JSON desse link:  
<https://github.com/Bluemix-Estag/NLU-IoT/blob/master/Assistant/workspace-9c5cda84-0ce7-4676-aac2-a5a9150eec19.json>



The screenshot shows the IBM Watson Assistant interface in a web browser. The title bar includes tabs for 'IBM Watson Service Page', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The URL in the address bar is <https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a...>. The main content area has a dark header with 'IBM Watson Assistant' and a navigation bar with 'Home' and 'Workspaces' (which is underlined). Below this, there's a section for 'Workspaces' with a large 'Upload' button highlighted by a red box. To the right, a specific workspace is displayed with the title 'Customer Service - Sample', a description 'A virtual assistant for customer service sample', language 'English (U.S.)', and a blue 'Edit sample' button. At the bottom, there's a promotional message about the latest features and links to 'Learn More' and 'Request Beta'.

IBM Watson Assistant

IBM Watson Service Page X IBM Watson Assistant X Node-RED : lab-iot-jppp.myblu X +

https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a... 90% ... Search

Cookie Preferences ? @

Home Workspaces

Workspaces Upload

Create a new workspace

Workspaces enable you to maintain separate intents, user examples, entities, and dialogs for each use or application.

You are using 0 of 5 available workspaces in this instance.

Create +

Customer Service - Sample :

A virtual assistant for customer service sample

English (U.S.)

Edit sample

Try out the latest features

A new way to build virtual assistants is coming. Be one of the first to test drive this new approach in your non-production service instance.

Learn More Request Beta

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.2. Fazendo o upload do Workspace da demonstração.

- Clique no símbolo indicado para fazer um upload.
- Escolha o arquivo JSON baixado do git.

The screenshot shows the IBM Watson Assistant interface in a web browser. The title bar includes tabs for 'IBM Watson Service Page', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area has a dark header with 'IBM Watson Assistant' and a navigation bar with 'Home' and 'Workspaces' (which is underlined). On the left, there's a section for creating a new workspace with a 'Create' button. On the right, there's a detailed view of a workspace named 'Customer Service - Sample' which is described as a 'virtual assistant for customer service sample' in English (U.S.). An 'Edit sample' button is visible. At the bottom, there's a promotional message about trying out beta features, with 'Learn More' and 'Request Beta' buttons.

IBM Watson Assistant

IBM Watson Service Page X IBM Watson Assistant X Node-RED : lab-iot-jppp.myblu X +

https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a... 90% ... Search

Cookie Preferences ? @

Home Workspaces

Workspaces ↑

Create a new workspace

Workspaces enable you to maintain separate intents, user examples, entities, and dialogs for each use or application.

You are using 0 of 5 available workspaces in this instance.

Create +

Customer Service - Sample :

A virtual assistant for customer service sample

English (U.S.)

Edit sample

Try out the latest features

A new way to build virtual assistants is coming. Be one of the first to test drive this new approach in your non-production service instance.

Learn More Request Beta

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Ache o nó com a intenção “#Medicao”.

The screenshot shows the IBM Watson Assistant interface in a web browser. The title bar includes tabs for "IBM Watson Service Page", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.myblu". The URL in the address bar is <https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a>. The main content area is titled "IBM Watson Assistant" and shows the "Dialog" tab selected. Below it, there are buttons for "Add node", "Add child node", "Add folder", and "Settings". The "Lab" workspace is selected. A list of intents is displayed:

- Bem-vindo (welcome)
- #Saudacao
- #Pisca
- #Medicao

The intent "#Medicao" is highlighted with a red rectangular box. Each intent entry shows "1 Response / 0 Context set / Does not return" below its name.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Ache o nó com a intenção “#Medicao”. Clique para editá-lo.

- Clique no ícone com 3 pontos na vertical e selecione a opção “Open JSON editor”

The screenshot shows the IBM Watson Assistant interface in a browser window. The URL is https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-ad15-407e-9e61-036749f38f39/workspaces/9c5cd84-0ce7-4676-aac2-a5a9150eec19/build/dialog#. The interface has tabs for Intents, Entities, Dialog, and Content Catalog, with Dialog selected. On the left, a tree view shows nodes like 'Lab', 'Bem-vindo', '#Saudacao', '#Pisca', and '#Medicao'. The '#Medicao' node is currently selected. To the right, there's a configuration area for this node, including fields for 'Name this node...', 'If bot recognizes:' (containing '#Medicao'), and 'Then respond with:' (set to 'Text'). A context menu is open over the 'Then respond with:' section, with the 'Open JSON editor' option highlighted with a red box. The bottom of the screen shows the browser's address bar and some other tabs.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Preencha as propriedades “user” e “password” com as credenciais usadas pela requisição CURL exemplo de sua Cloud Function.  
Note que “password” possui mais caracteres que “user”.

The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for "IBM Cloud Functions - Action", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.mybluemix.net", "Bluemix-Estag/NLU-IoT", and a "+" button. The main area is titled "IBM Watson Assistant" and shows the "Dialog" tab selected. On the left, there's a sidebar with icons for workspace, intents, entities, and dialog. Below it, a list of nodes is shown under the category "Lab": "Bem-vindo" (welcome), "#Saudacao" (Saudacao), "#Pisca" (Pisca), and "#Medicao" (Medicao). Each node has a "1 Response / 0 Context set / Does not return" status. To the right, a modal window is open for creating a new node. It asks "Name this node..." and "Then respond with:". Below is a code editor containing a JSON-like response template:

```
1 | {
2 |   "context": {
3 |     "private": {
4 |       "my_credential": {
5 |         "user": "REDACTED",
6 |         "password": "REDACTED"
7 |       }
8 |     }
9 |   },
10 |   "output": {
11 |     "generic": [
12 |       {
13 |         "values": [],
14 |         "response_type": "text",
15 |         "selection_policy": "sequential"
16 |       }
17 |     ],
18 |     "actions": []
19 |   }
20 | }
```

Below the code editor, there are "Tips:" and examples for "Array [value1, value2]", "Complex object {"variable1" : value1, "variable2" : value2}", and "String "value1"".

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Preencha as propriedades “Sua\_ORG”, “Seu\_SPACE” com os parâmetros usadas pela requisição CURL exemplo de sua Cloud Function. Adicione, ao final, o nome de sua action que pega a medição do sensor (esp-sensor).

The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for "IBM Cloud Functions - Action", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.mybluemix.net", and "Bluemix-Estag/NLU-IoT". The main area is titled "IBM Watson Assistant" and shows the "Dialog" tab selected. On the left, there's a sidebar with icons for workspace, intents, entities, and dialog. The main workspace is titled "Lab" and contains four nodes: "Bem-vindo" (welcome), "#Saudacao" (Saudacao), "#Pisca" (Pisca), and "#Medicao" (Medicao). Each node has a "1 Response / 0 Context set / Does not return" status. To the right, there's a "Name this node..." input field and a "Then respond with:" section containing a JSON code editor. The JSON code is as follows:

```
11  "generic": [
12    {
13      "values": [],
14      "response_type": "text",
15      "selection_policy": "sequential"
16    }
17  ],
18  "actions": [
19    {
20      "name": "/<Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
21      "type": "server",
22      "parameters": {
23        "cmd": ""
24      },
25      "credentials": "$private.my_credential",
26      "result_variable": "$result"
27    }
28  ]
29 }
30 }
```

A red box highlights the "name" field in the JSON code. Below the code editor, there are "Tips:" and examples for "Array", "Complex object", and "String".

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Siga os mesmos passos para os nós “#apaga”, “#acende” e “#pisca” .

The screenshot shows the IBM Watson Assistant interface in a browser window. The URL is <https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-ad15-407e-9e61-036749f38f39/workspaces/9c5cd84-0ce7-4676-aac2-a5a9150eec19/build/dialog#>. The left sidebar shows 'Workspaces / Lab / Build' with tabs for 'Intents', 'Entities', 'Dialog' (selected), and 'Content Catalog'. The main area lists intents: '#Saudacao', '#Pisca', '#Medicao', '#Ascende', and '#Apaga'. A modal dialog is open for the '#Apaga' intent, titled 'Name this node...'. It contains fields for 'If bot recognizes:' with the value '#Apaga' and 'Then respond with:' which is currently set to 'Text'. A context editor dropdown menu is open, showing options 'Open JSON editor' and 'Open context editor'. The status bar at the bottom shows the same URL.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Siga os mesmos passos para os nós “#apaga”, “#acende” e “#pisca”.

- Preencha as propriedades “user” e “password”.

The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, Node-RED : lab-iot-jp, NLU-IoT/esp-sensor.js, and a plus sign. The main area is titled "IBM Watson Assistant" and shows the "Dialog" tab selected. On the left, there's a sidebar with icons for Workspaces, Lab, and Build. The main workspace displays a list of intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. The "#Apaga" intent is currently selected and highlighted with a blue border. To the right of the intents, a modal dialog is open with the heading "Name this node...". Inside the dialog, the text "#Apaga" is entered into a text input field, followed by a minus sign and a plus sign. Below this, the instruction "Then respond with:" is followed by a code editor containing a JSON-like response template. The code is partially redacted with a large red rectangle, but the beginning of the template is visible:

```
1 {
2   "context": {
3     "private": {
4       "my_credential": {
5         "user": "REDACTED",
6         "password": "REDACTED"
7       }
8     },
9     "output": {
10       "generic": [
11         {
12           "values": [],
13           "response_type": "text",
14           "selection_policy": "sequential"
15         }
16       ]
17     },
18   "actions": [
19 
```

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Siga os mesmos passos para os nós “#apaga”, “#acende” e “#pisca”.

- Preencha as propriedades “user” e “password”.

- Preencha as propriedades “Sua\_ORG”, “Seu\_SPACE”

The screenshot shows the IBM Watson Assistant interface within a browser window. The title bar includes tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, Node-RED : lab-iot-jp, NLU-IoT/esp-sensor.js, and a plus sign. The main area has a dark header with 'IBM Watson Assistant' and a navigation sidebar with icons for Workspaces, Lab, and Build. The 'Dialog' tab is selected, showing a list of intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. Each intent card displays '1 Response / 0 Context set / Does not return'. On the right, a modal window is open for the #Pisca intent, titled 'Name this node...'. Below it, the text 'then respond with:' is followed by a JSON code block:

```
12     "values": [],
13     "response_type": "text",
14     "selection_policy": "sequential"
15   }
16 ],
17 "actions": [
18   {
19     "name": "/<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
20     "type": "server",
21     "parameters": {
22       "cmd": "turn",
23       "value": "off"
24     },
25     "credentials": "$private.my_credential",
26     "result_variable": "$result"
27   }
28 ]
29 }
30 ]
31 }
```

Below the code, there are 'Tips:' and three examples: 'Array [value1, value2]', 'Complex object {"variable1": value1, "variable2": value2}', and 'String "value1"'.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Coloque o nome da Action que controla o LED (no exemplo usamos esp-LED).

The screenshot shows the IBM Watson Assistant interface with the 'Dialog' tab selected. On the left, a list of intents is visible: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. Each intent has a description below it: '1 Response / 0 Context set / Does not return'. On the right, a modal window is open for the #Pisca intent, asking 'Name this node...'. Below the input field is a section labeled 'then respond with:' containing a JSON code editor. The code is as follows:

```
12     "values": [],
13     "response_type": "text",
14     "selection_policy": "sequential"
15   }
16 ],
17 "actions": [
18   {
19     "name": "/<Nome_da_sua_action_ACTION>",
20     "type": "server",
21     "parameters": {
22       "cmd": "turn",
23       "value": "off"
24     },
25     "credentials": "$private.my_credential",
26     "result_variable": "$result"
27   }
28 ]
29 ]
30 }
31 }
```

A red box highlights the action name `/<Nome_da_sua_action_ACTION>` in the JSON code.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.3. Colocando as credenciais da sua Cloud Function.

- Coloque o nome da Action que controla o LED (no exemplo usamos esp-LED).

- Também precisamos mandar os parâmetros da função. Como queremos desligar: comando = “turn”, valor = “OFF”.

The screenshot shows the IBM Watson Assistant interface in a browser window. The URL is https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a... The main area displays a list of intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. The #Apaga intent is selected, and a modal window is open to "Name this node..." with the placeholder "men responda com:". Below it, the "respond with:" section contains the following JSON code:

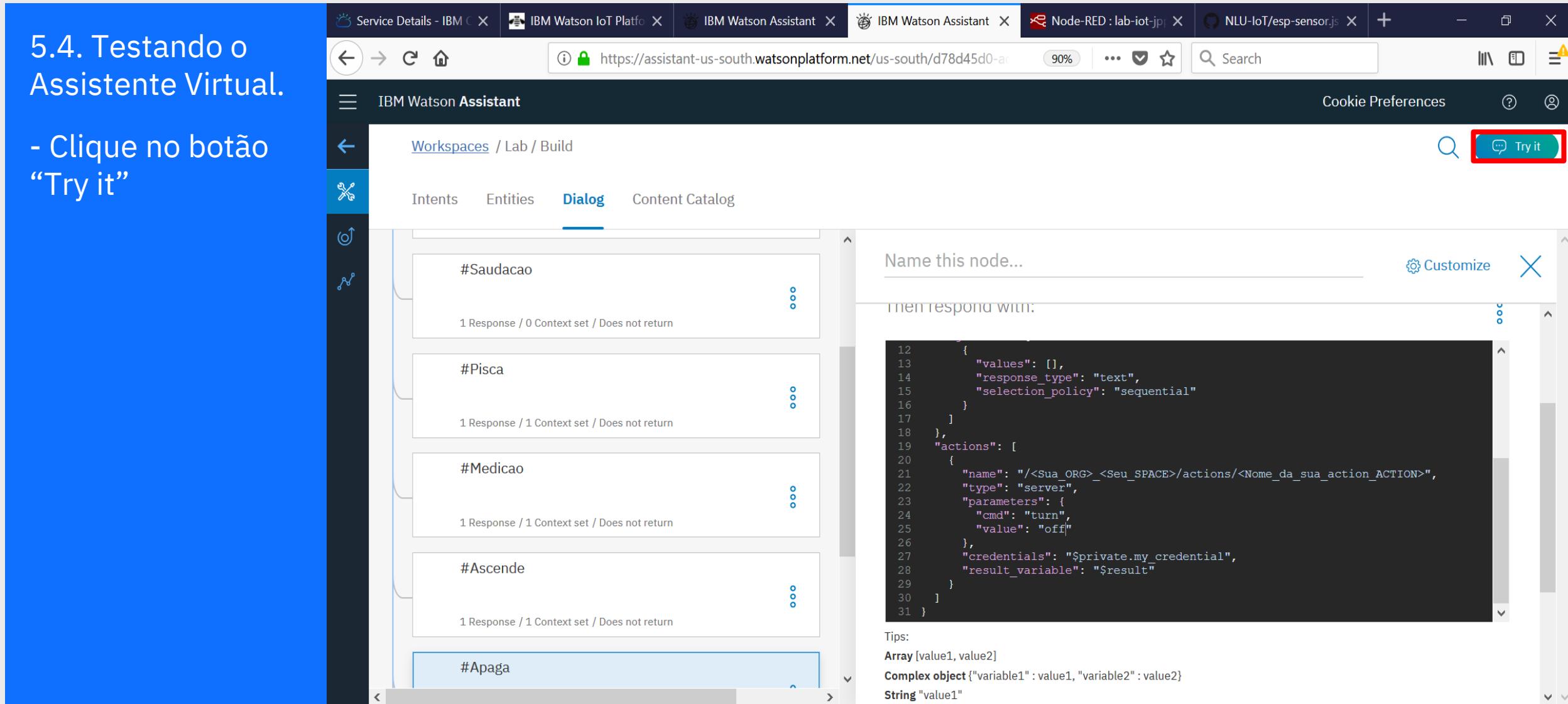
```
12     {
13         "values": [],
14         "response_type": "text",
15         "selection_policy": "sequential"
16     }
17 ],
18 "actions": [
19     {
20         "name": "</Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
21         "type": "server",
22         "parameters": {
23             "cmd": "turn",
24             "value": "off"
25         },
26         "credentials": "$private.my_credential",
27         "result_variable": "$result"
28     }
29 ]
30 }
31 }
```

A red box highlights the "parameters" section of the JSON code, specifically the "cmd" and "value" fields. The modal also includes a "Tips:" section with links to Array, Complex object, and String documentation.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.4. Testando o Assistente Virtual.

- Clique no botão “Try it”



The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, Node-RED : lab-iot-jp, NLU-IoT/esp-sensor.js, and a plus sign. The main navigation bar has links for Workspaces, Lab, and Build, with 'Dialog' selected. On the left, there are icons for Intents, Entities, and Content Catalog. The main content area displays five intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. Each intent card shows '1 Response / 0 Context set / Does not return'. To the right, a modal window is open for the #Apaga intent, asking 'Name this node...' and providing a JSON configuration for the action. The JSON code is as follows:

```
12     "values": [],
13     "response_type": "text",
14     "selection_policy": "sequential"
15   }
16 ],
17 "actions": [
18   {
19     "name": "/<Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
20     "type": "server",
21     "parameters": {
22       "cmd": "turn",
23       "value": "off"
24     },
25     "credentials": "$private.my_credential",
26     "result_variable": "$result"
27   }
28 ]
29 ]
30 }
31 }
```

Below the JSON, there are tips for Array, Complex object, and String values.

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.4. Testando o Assistente Virtual.

- Clique no botão “Try it”

- Faça uma interação com o assistente, exemplo: “Faça o LED do ESP ligar”.

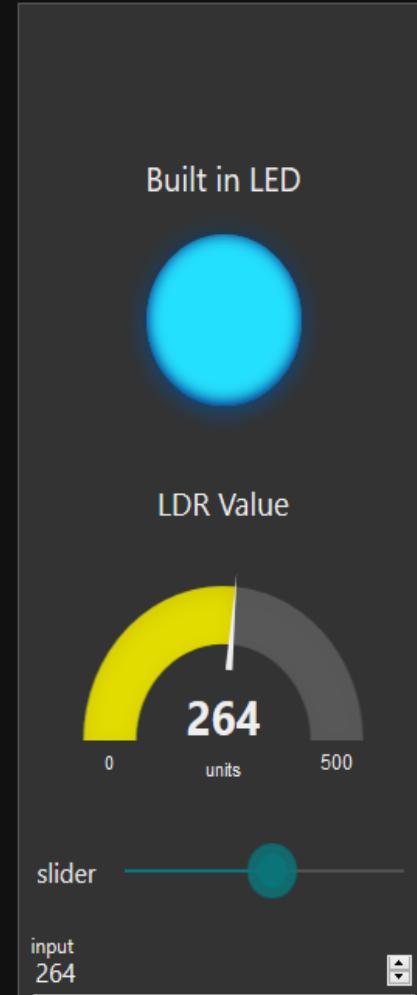
The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for "IBM Cloud Functions - Action", "IBM Watson Assistant", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.mybluemix.net", and "NLU-IoT/esp-LED.js at master". The main area is titled "IBM Watson Assistant" and shows a "Dialog" tab selected. Below it is a "Content Catalog" tab. A toolbar has buttons for "Add node", "Add child node", "Add folder", and "Settings". On the left, there's a sidebar with icons for workspace, lab, build, and settings. The main content area displays a "Lab" dialog tree with nodes like "Bem-vindo", "#Saudacao", "#Pisca", and "#Medicao", each with a response count of 1. To the right, a "Try it out" section contains a text input field with placeholder text "Enter something to test your virtual assistant" and a note "Use the up key for most recent".

# 5. Criando o Assistente Virtual com o Watson Assistant

## 5.4. Testando o Assistente Virtual.

- Clique no botão “Try it”
- Faça uma interação com o assistente, exemplo: “Faça o LED do ESP ligar”.
- O LED do dispositivo simulado deve acender.
- Faça os testes que quiser!

Simulated Device

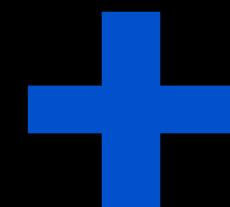
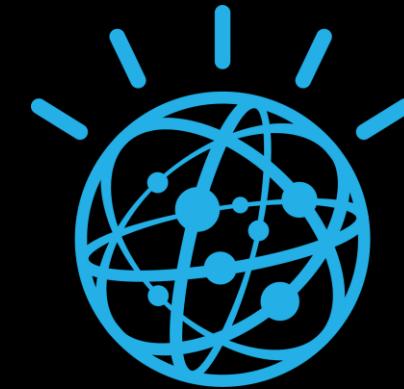


Concluindo:

Digitalizamos o mundo físico.

Interagimos em  
linguagem natural.

**De forma escalável  
e eficiente.**



# Obrigado

**Eduardo Petecof Mattoso**

Watson & Cloud TechSales Intern

[epetecof@br.ibm.com](mailto:epetecof@br.ibm.com)

**João Pedro Poloni Ponce**

Value Creation Intern - BlueDev

[joao.poloni@ibm.com](mailto:joao.poloni@ibm.com)

