DEVOXX™ France

HandsOn Lab
Watson Node-RED on Bluemix

Version: 3.0.0
Last modification date: April 4th, 2017
Owner: Yves LE CLEACH

Twitter : @ylecleach #devoxxfr @IBMFranceLab
Email: yves.lecleach@fr.ibm.com
LinkedIn: https://www.linkedin.com/in/yveslecleach
Meetup: http://ibm.biz/BluemixParisMeetup

Table of Contents

This Lab is designed so it can be done entirely using a Bluemix Trial account.

Copyright IBM Corporation 2017. All rights reserved.

1 Introduction

IBM Watson Developer Cloud (WDC) is a set of micro-services available as REST APIs that applications can use to incorporate cognitive capabilities. Credential access to the APIs is available through Bluemix - IBM's cloud platform as a service. WDC also proposes severals SDK (Java, Node.js, ..) that eases those APIs integration with various languages.

Node-RED is an open-source prototyping tool that uses a drag/drop/wiring paradigm to build applications. Complex applications can be built quickly with little or no programming required, and enable you to connect together hardware devices, APIs and online services in new and interesting ways. It is open source, and can be installed on small devices (e.g. raspberry pi), but a customized version is proposed on Bluemix, with additional open source nodes for Bluemix or Watson. Watson nodes are also usable outside of Bluemix, e.g., on a small computer (raspberry pi) or in any others Cloud Platform provider.

Node-RED offers a large palettes of ready-to-use nodes. The Watson nodes presented in this Lab are a subset of the Cognitive APIs available through the Watson Developer Cloud. The Watson nodes implementation is using the Watson Node.js SDK. (also open source and available on WDC as those nodes).

This document is based on a collection of Node-RED Watson Labs available on Git on WDC:
https://github.com/watson-developer-cloud/node-red-labs

The basic labs are simple, standalone examples developed to show how to call each individual Watson Node-RED node. The advanced labs build on this to bring multiple nodes together to create even more complex applications.

In September 2016, a set of seven very cool "Node-RED Watson Starter Kits" has been added, and are using new recent and interactive nodes (Microphone, Camera, Speaker).
Related article : https://www.linkedin.com/pulse/watson-node-red-residency-sept-2016-paul-read-1

Important: all the presented nodes are open-source, so do not hesitate to send feedbacks through GitHub website (bugs, documentation issue, feature requests), or any suggestion to me.

1.1 Lab Objectives

- Learn Basics on Bluemix concepts: App, Services, Continuous Delivery, Pipeline, Deployment.
- Discover Node-RED and learn essentials development tips as a citizen developer, but also as a Node-RED node contributor
- Discover some of the Watson API through the Watson nodes on Bluemix
- Build your first Cognitive Node-RED applications using Watson nodes
- Advanced (Node.js development): Learn how to build a Node-RED node based on two nodes I have coded : Watson Visual Recognition v3 node and Business Rules node, which are implemented in two flavors : the first one implement the Watson Node.js API SDK, and the second one directly implement the Business Rules service API.

1.2 Prerequisites

For the following Lab exercices, you will need the following:

- a Bluemix Account : cf Annexes section if you do not have an account.

Copyright IBM Corporation 2017. All rights reserved.

- A GitHub account and install Git CLI on your laptop : cf Annexes section if you do not have an account.

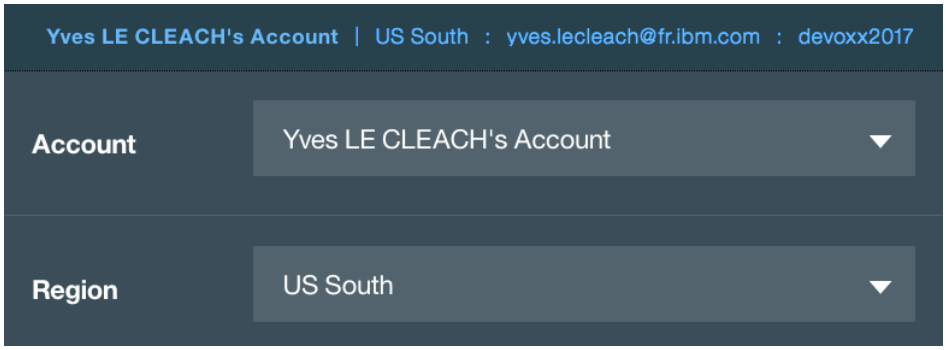
2 Discovering Bluemix and Node-RED

This section is aimed for newcomers to Bluemix and/or Node-RED.

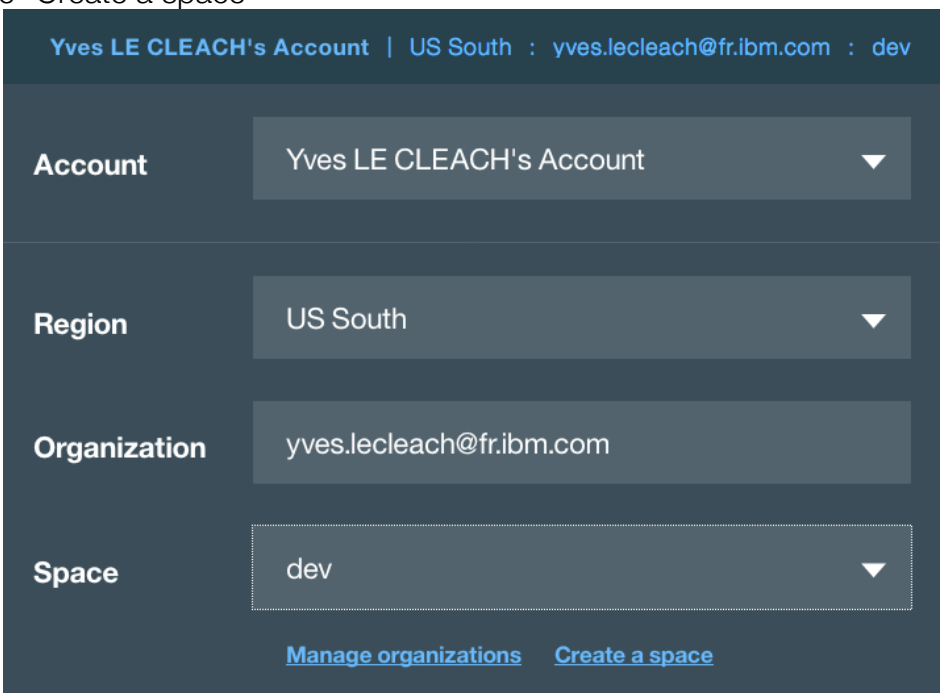
2.1 Create your first Node-RED application on Bluemix

First, log in onto your Bluemix account : https://bluemix.net and creates a Space '**devoxx2017'** in which you will work for this lab session.

First, select in the top right corner, the **Region "US South"** in which we will deploy all this Lab content :

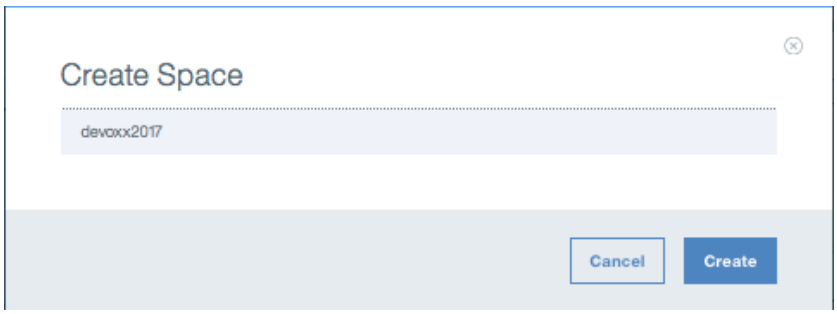


Then Click on the "Create a space"

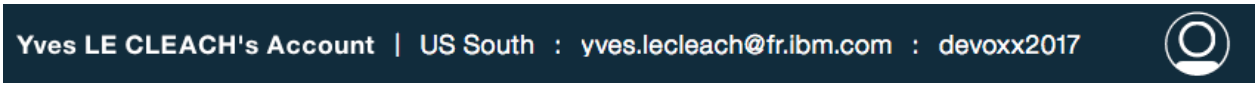


Enter "devoxx2017" and click on the Create button :

Copyright IBM Corporation 2017. All rights reserved.

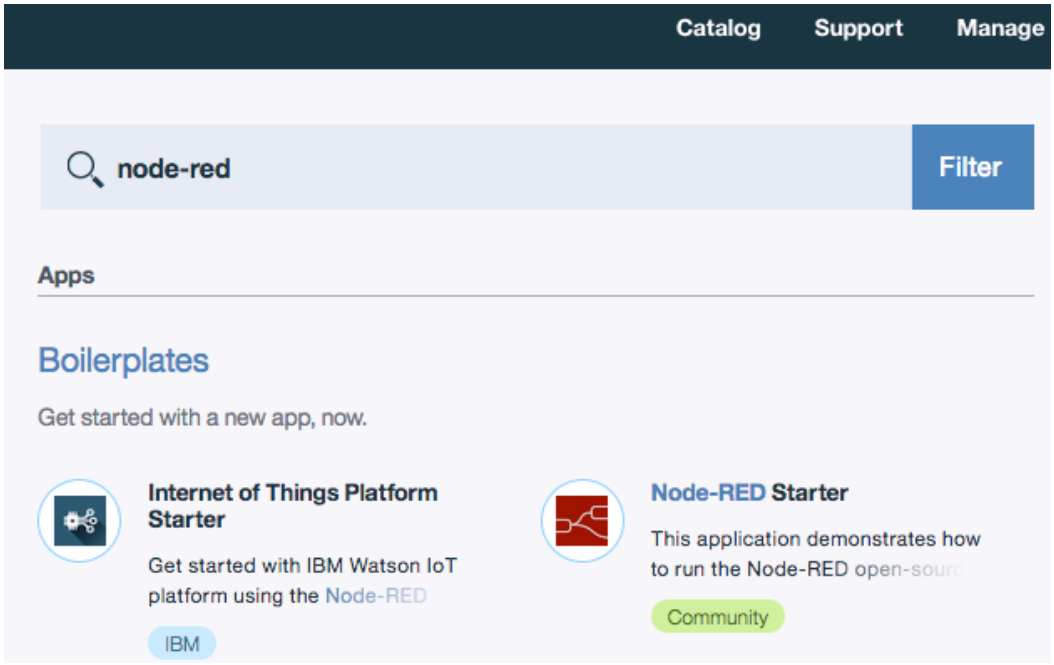


Now you should see as the selected space the newly created “devoxx2017” space as below :



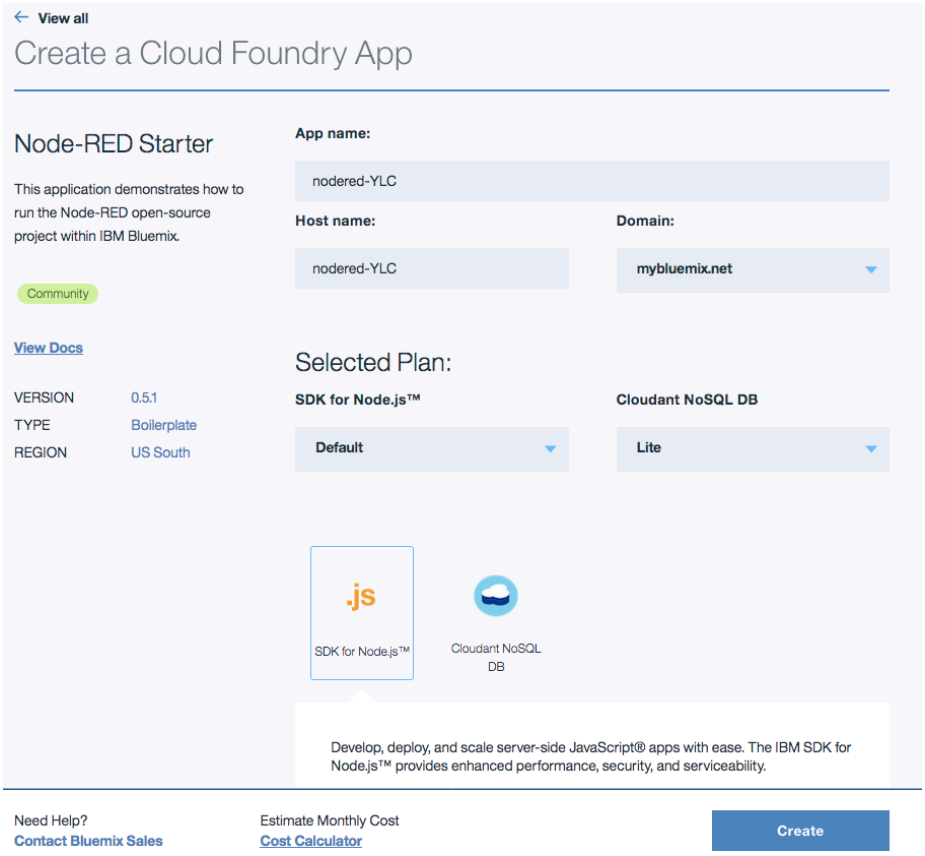
Note : if the **“devoxx2017”** space is not selected, please select it.

Then, go to the **Catalog** in the main menu at the top right, and type “node-red” in the Search field. Select the Node-RED Starter to create a new Node-RED app.
:

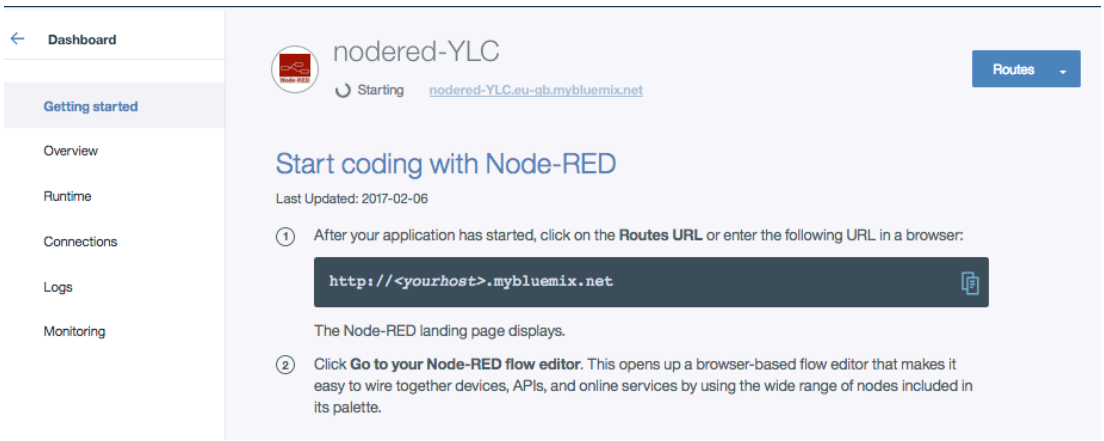


In the “**App name**” field, type “**nodered-XXX**” (replace XXX by your initial, so you will ensure that the hostname is unique) and click on the “**Create**” button.

Copyright IBM Corporation 2017. All rights reserved.



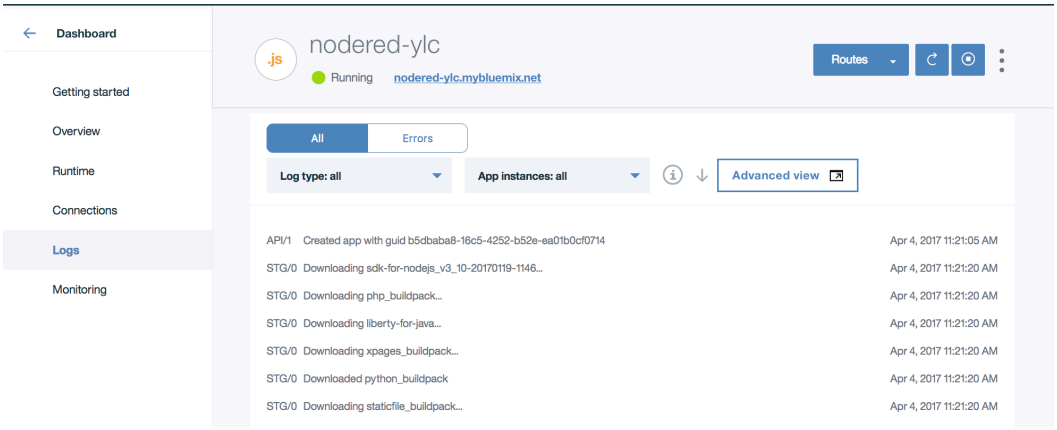
You should see the screen below that correspond to a Getting started which mention some basics Cloud Foundry commands line. (CF CLI).



Note: we will not use the CF CLI in this Lab. (CLI stands for Command Line Interface).

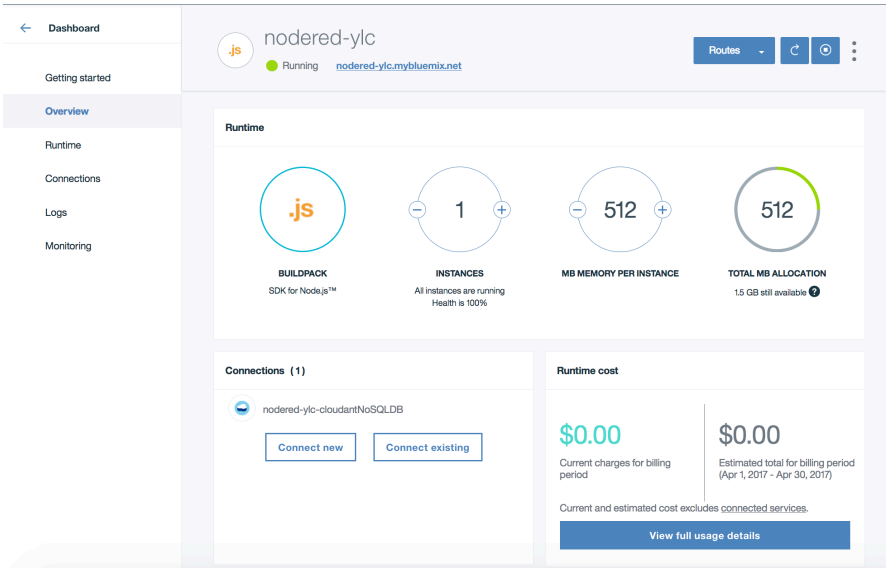
The provisioning of your new node-RED instance is starting. You can go immediately in the Logs Tab of your app to see the details of the provisioning sequence behing the scene. (this also helps one to understand the advantage to use a PaaS technology such as Cloud Foundry).

Copyright IBM Corporation 2017. All rights reserved.



After 2-3 minutes, the application should be available and be online. (green Status)

Click now in the Overview tab of your application.



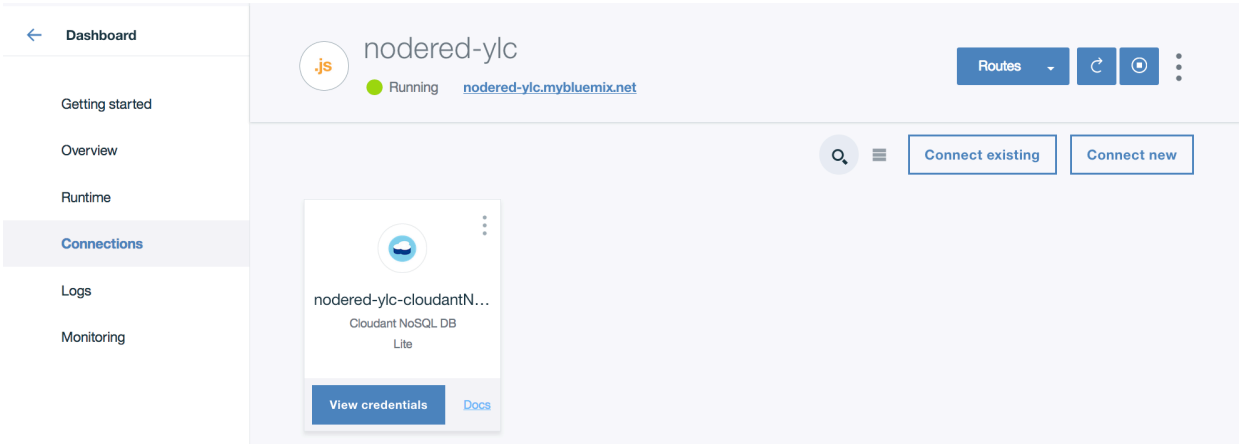
In a Node-RED app, which is run by a Node.js server instance, you can define any numbers of functionality you want, as on any application servers. So a Node-RED app, can host any numbers of applications you desire, as it can be done on a Java EE application server. (e.g: Liberty Java server). You can change the INSTANCES number and the MB MEMORY per instance as required for your(s) application(s).

Notice : Node-js is designed and fits well for Non Blocking I/O applications, and many others usages, and is more and more used worldwide. Node-RED is designed to have a small memory footprint, and can run on small computers (e.g. Raspberry Pi)

Connections Tab

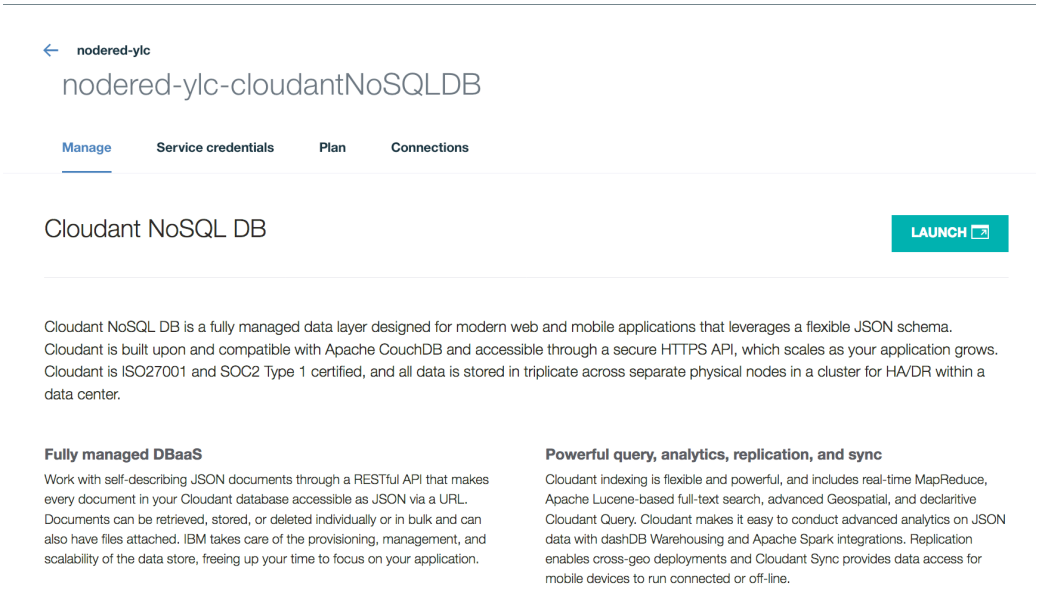
In this tab you will see all Bluemix services instances which are Link (or bound) to your app. One given service instance can be linked to any number of applications inside the same Space. A Cloud Foundry Organization contains typically multiple space to organize apps and service instances. (ex of space name : dev, test, prod, demo). You can invite others Bluemix users in a given Org/Space.

Copyright IBM Corporation 2017. All rights reserved.



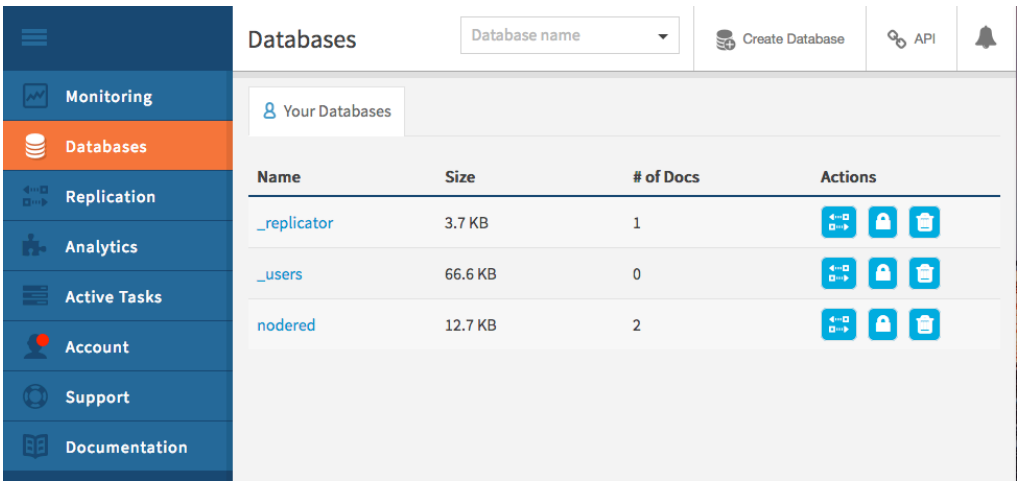
You can observe that the Node-RED app is bound to a Cloudant service instance which has been provisioning at the creation time of the Node-RED app. (this creation time where all resources/hardware/software are allocated is called STAGING).

You can spend 2 min to click on the cloudant service instance and click on the Launch Dashboard.

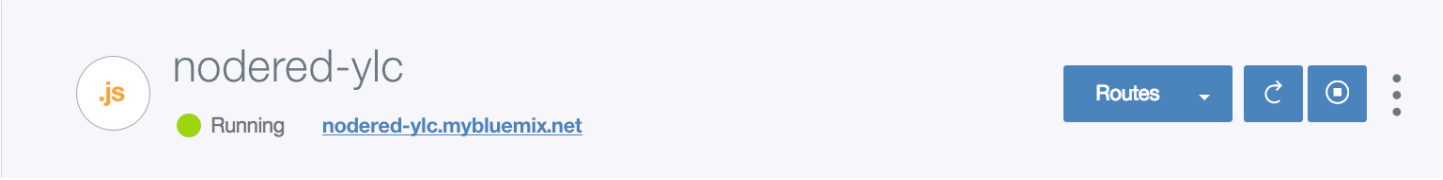


In the Database tab of the Cloudant dashboard, you will find the “nodered” database used by your Node-RED app to save all Node Flows and configuration you will produce. (all is stored as Json documents). You do not need to update directly this “nodered” database as it is used by the Node-RED framework for its own persistence.

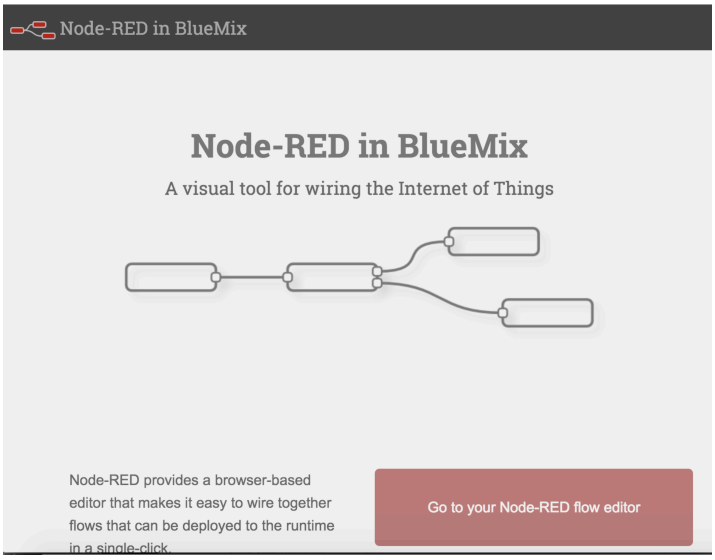
Copyright IBM Corporation 2017. All rights reserved.



Now close the Cloudant dashboard, and return back to your Node-RED Dashboard.

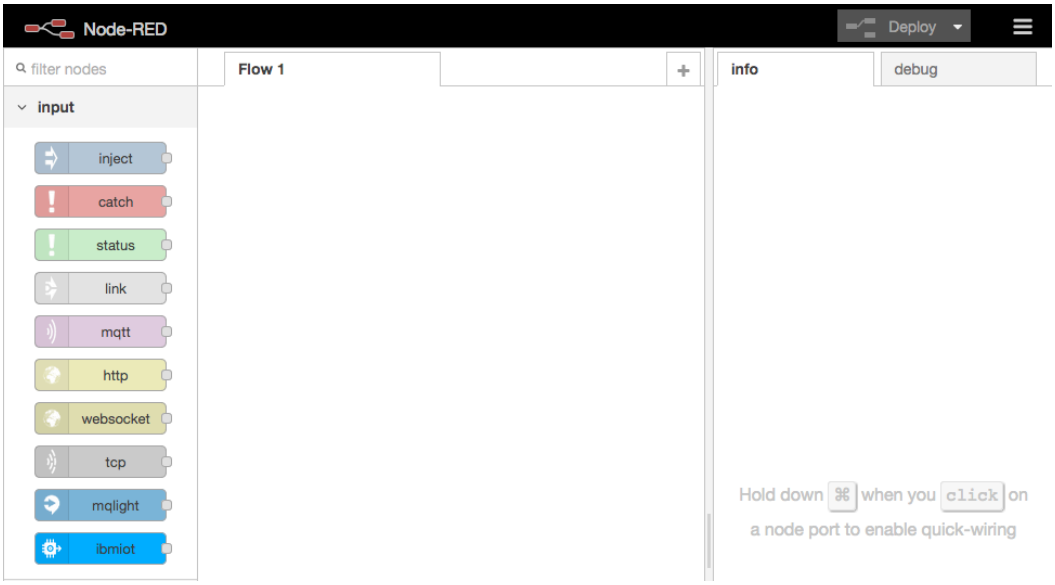


and click on the link beside your App name.You should arrive on the default page of your Node-RED App :



Now click on the RED button to enter the web-based Node-RED Flow Editor:

Copyright IBM Corporation 2017. All rights reserved.



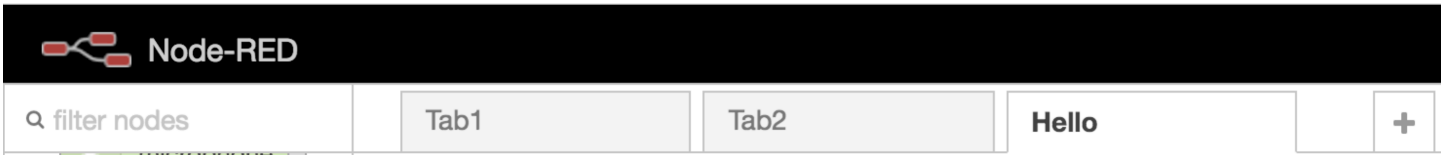
Quickly Browse the Node Palette, and Look at the Info Tab to have some more information on the selected node.

You are now ready to start developing your node-red applications. Lets start by the Node-RED basics.

2.2 Creating your first flow

First create a new Flow tab for this exercice using the + icon:

Click on the added Tab. (Name is Flow2), and modify the name to “Hello”.



Notice : Flows can communicates between Tabs only using the “link in” and “link out” nodes :

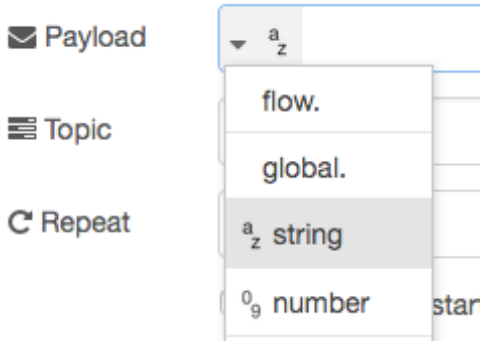
Notice: see OK Watson Starter to see an example of usage of the “link” nodes.

Now, select the Hello Tab, and lets create our first flow.

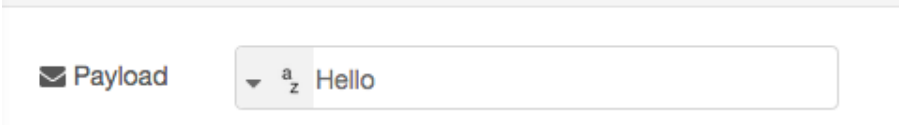
Copyright IBM Corporation 2017. All rights reserved.

This program is a very simple flow that prints the message 'Hello World' on the screen. Here you can see Node-RED's user interface, the colored blocks on the screen are called nodes, which is a visual representation of a piece of JavaScript code to carry out a task. To build this 'Hello World' flow you need to take the following steps:

1. Drag an 'Inject node' to the canvas
2. Double click this node to see the options
3. Use the drop-down, to select **string** for the payload



4. Type 'Hello' on the “Topic” line: this will cause to inject hello into the flow when clicked on the inject node) and click on ok, to save and close this node.



5. Add a 'Function node' to the canvas, open it and place this on the first line into the function:

```
msg.payload += ' World !';
```

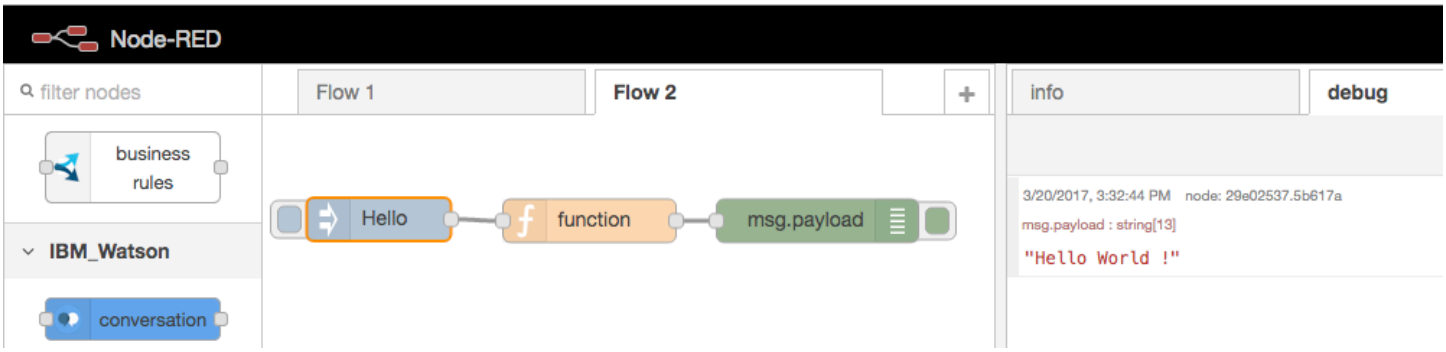
You can optionally set a Label to the node display (Name)

The complete function should look like this:

Copyright IBM Corporation 2017. All rights reserved.

6. Add a 'Debug node' to the canvas.
7. Wire the 'Inject node' to the 'Function node' and the function node to the 'Debug node'. Most nodes have a grey circle on their left side, which is their input port, and on their right side, which is their output port. Left clicking and dragging the output to the input port of the next node connects the two together.
8. Press 'Deploy'.

Now you have build your first Hello World flow. Test it by clicking on the 'Inject node', you will see some output in the debug Tab on the right (click on 'Debug' to change the view from info to debug).



2.3 Importing or Exporting Flows

With Node-RED you can export/import easily the flows you have designed or ones from the node-red community. (http://flows.nodered.org/). The import/export in Node-RED is using an internal JSON representation.

If you want to try this now then select the sample flow below:

https://raw.githubusercontent.com/ylecleach/labs/master/hello-flow.json

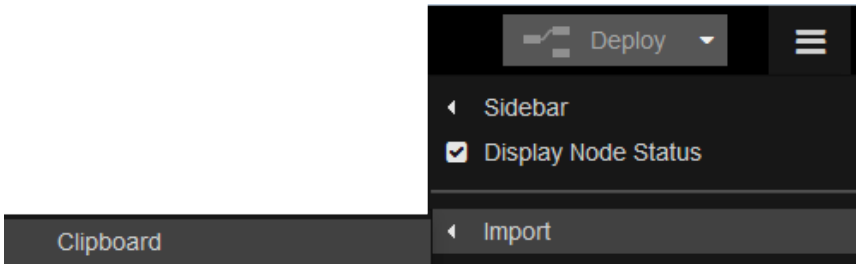
Select the flow and copy it to the clipboard (Ctrl-C).

Copyright IBM Corporation 2017. All rights reserved.

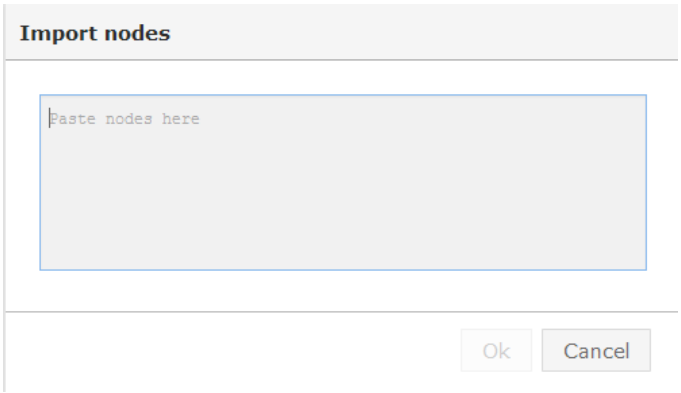
Select the Flow 1 tab in the Node-RED Editor :

Import the flow into Node-RED using by selecting the node-RED menu available in the top right corner:

and select the import from clipboard option.

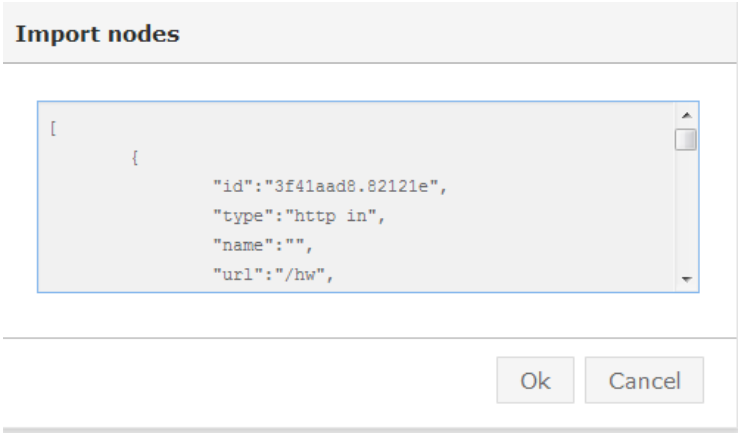


You will be presented with a form in which you create nodes by entering json data.

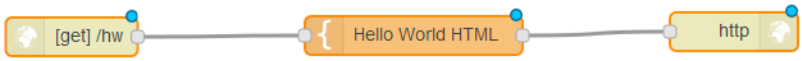


Import your copied flow by pasting (Ctrl-P) from the clipboard into the form.

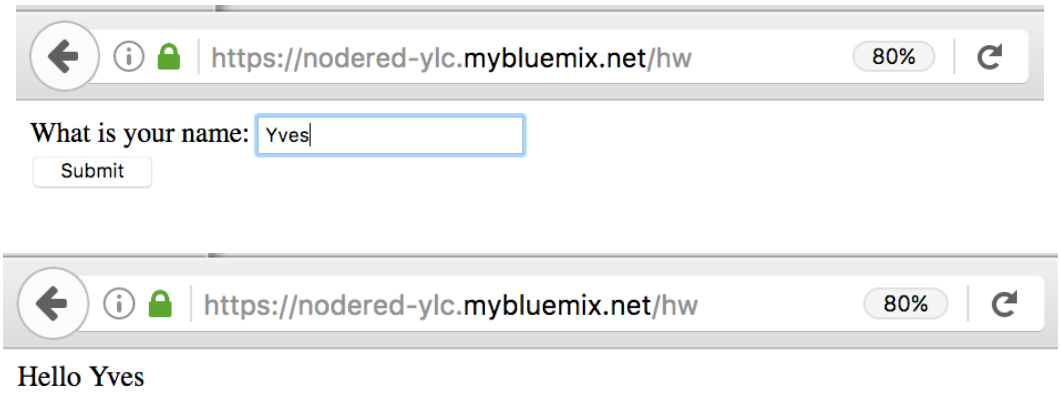
Copyright IBM Corporation 2017. All rights reserved.



Place the imported flow onto your node-RED page and press DEPLOY.



You can test by entering this URL : (replace the hostname by yours)



You can spend a few time by looking at the “template” node code, which is containing basic HTML. The two others nodes “http in” and “http request” nodes are used to create simple web services.

2.4 Securing your Node-RED editor (recommended)

By default, the Node-RED editor is open for anyone to access and modify flows, which is not desirable. Let’s configure the Node-RED security !

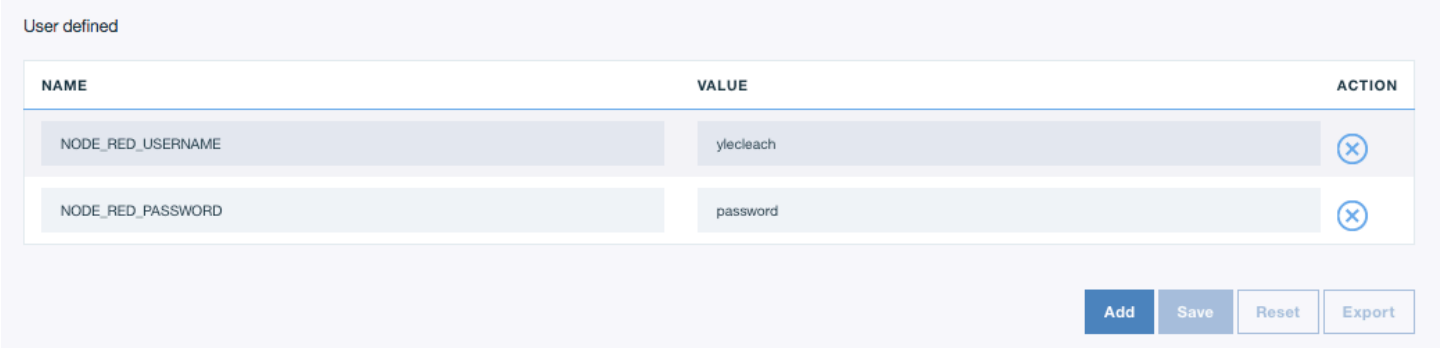
First return back to your Node-RED App details in the Bluemix Console Dashboard, and select the Runtime in the left menu, and select the “Environment variables” tab.

Copyright IBM Corporation 2017. All rights reserved.

Then Environment Variables tab.

Under the section **"User-Defined"** at the bottom of this page, add the following those 2 variables with their corresponding values.

- NODE_RED_USERNAME - *the username to secure the editor with*
- NODE_RED_PASSWORD - *the password to secure the editor with*

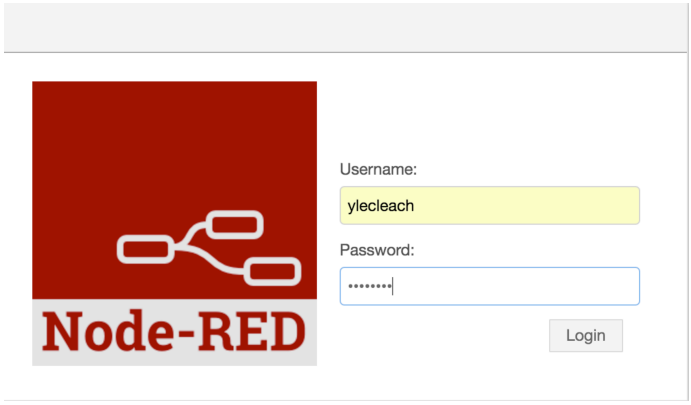


And Click on the SAVE button.

Last step to secure your Node-RED app is to RESTART it through the Bluemix Dashboard :

Now, when someone will click on the button of the Node-RED editor, an authentication dialog box will be prompted.

Copyright IBM Corporation 2017. All rights reserved.



2.5 Enable Continuous Delivery to your app

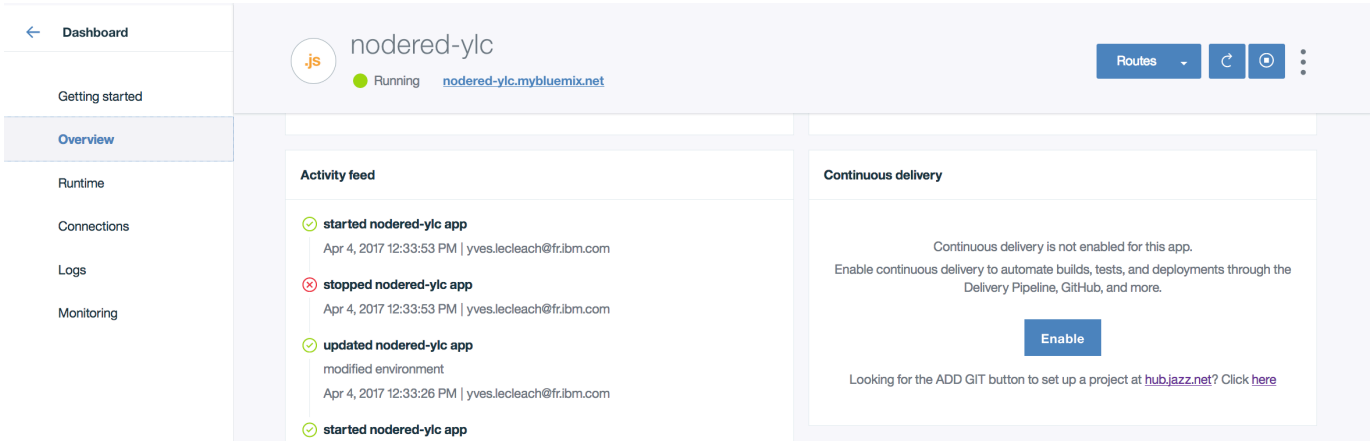
Before you start and to make use of the starter kit applications in section 4, you will need some extra node-RED nodes availables as NPM modules.

To modify your Node-RED app configuration or resources (as HTML static files) you need to edit it in a Project.

To do so, you have to Enable the **Continous Delivery** on your app that will :

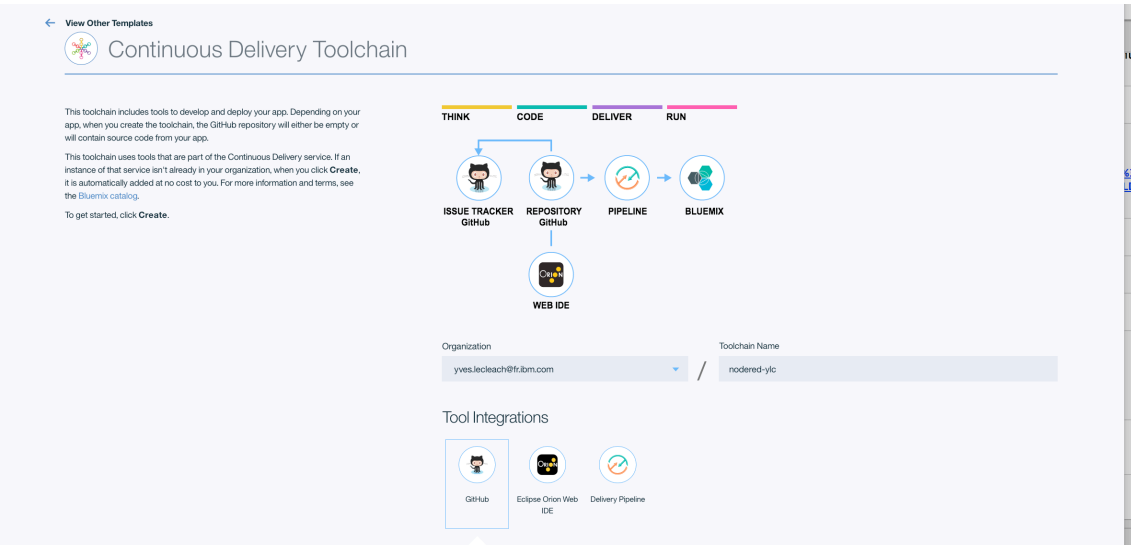
- create a Toolchain for the Node-RED app code (the Project) with
  - a project on your GitHub account. (code and issue tracking links)
  - a default Delivery Pipeline configured to listen to your project updates.
  - A link to Orion Web IDE that enables to edit your project online.

Click on the Overview tab of your application, then click on the **"Enable"** button in **"Continuous delivery"** section at the bottom right of the page.

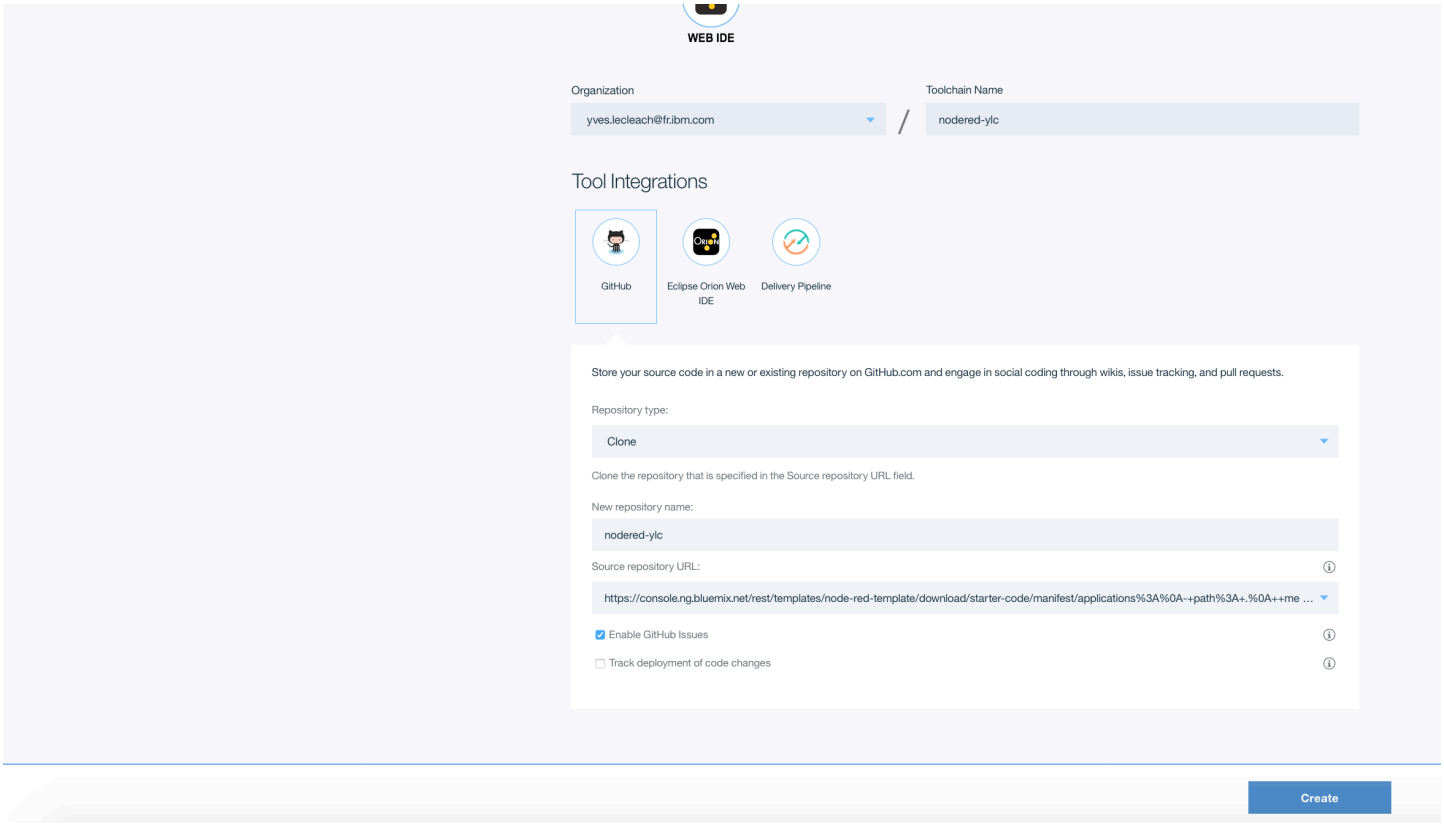


You should land to this page :

Copyright IBM Corporation 2017. All rights reserved.

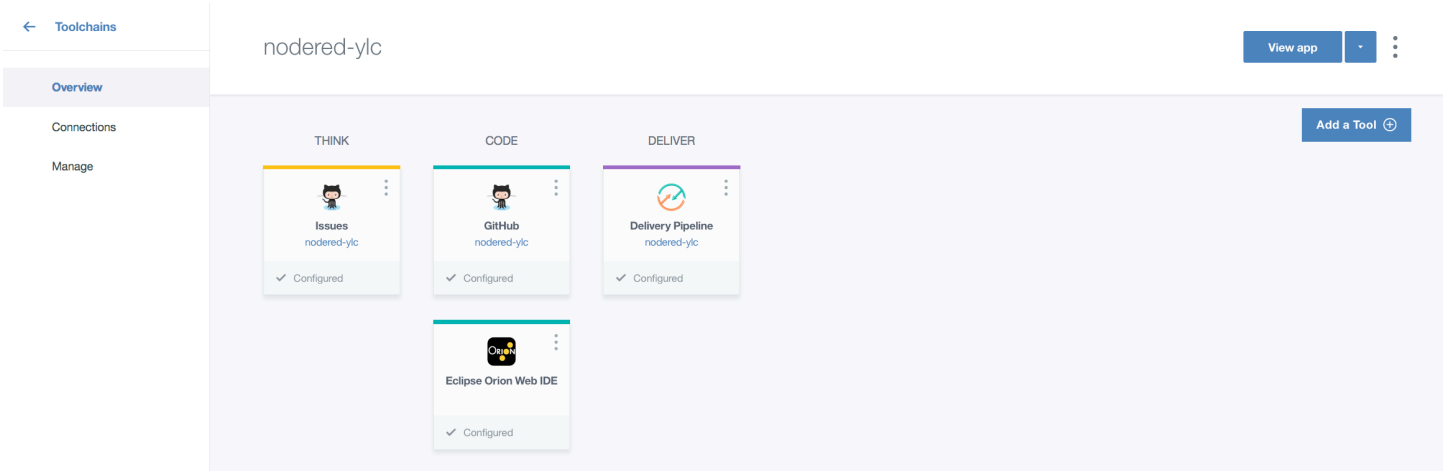


Scroll down, review all settings to have an idea of what is configurable from this point, but leave unchanged all defaults. Finish this step by clicking on the **Create button.**



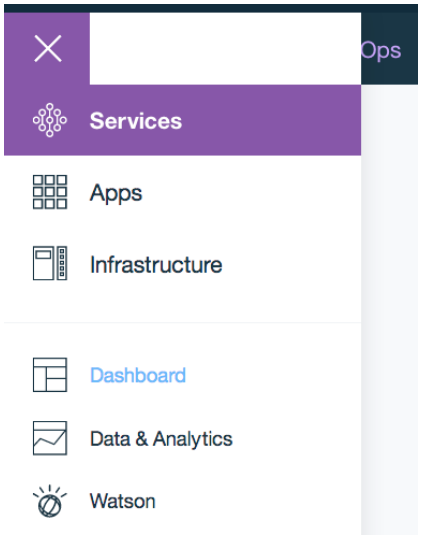
Finally, your project ‘s Tool Chain is created as below :

Copyright IBM Corporation 2017. All rights reserved.

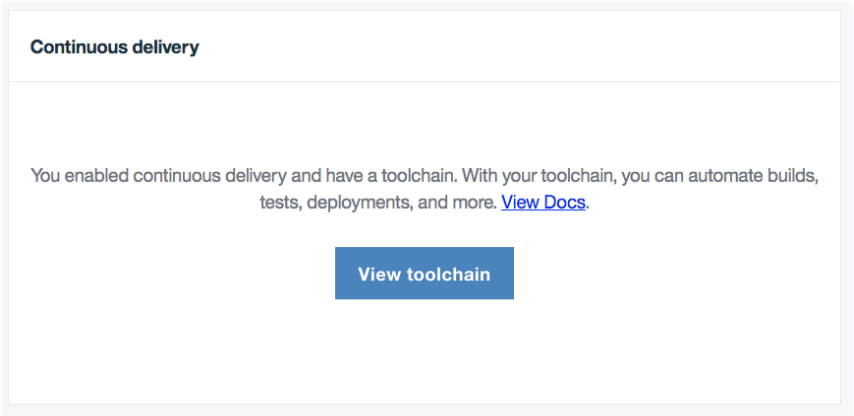


If you click on the Back button you will go to the main Toolchains menu of IBM Bluemix DevOps. https://console.ng.bluemix.net/devops/toolchains

To return back to your application dashboard use the left menu and choose Dashboard.



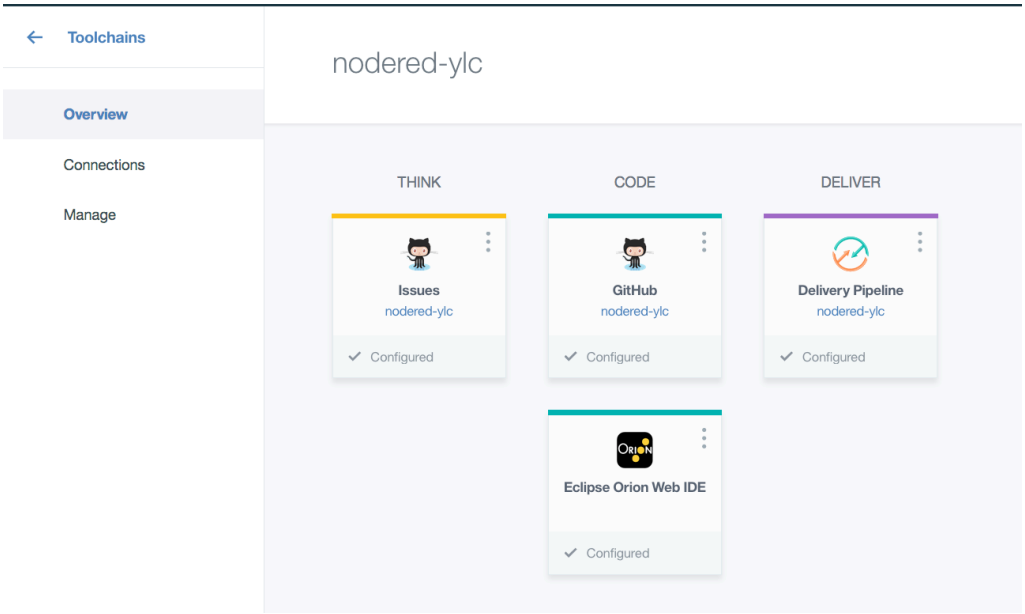
Now, in your application dashboard you can now access directly to the associated ToolChain.



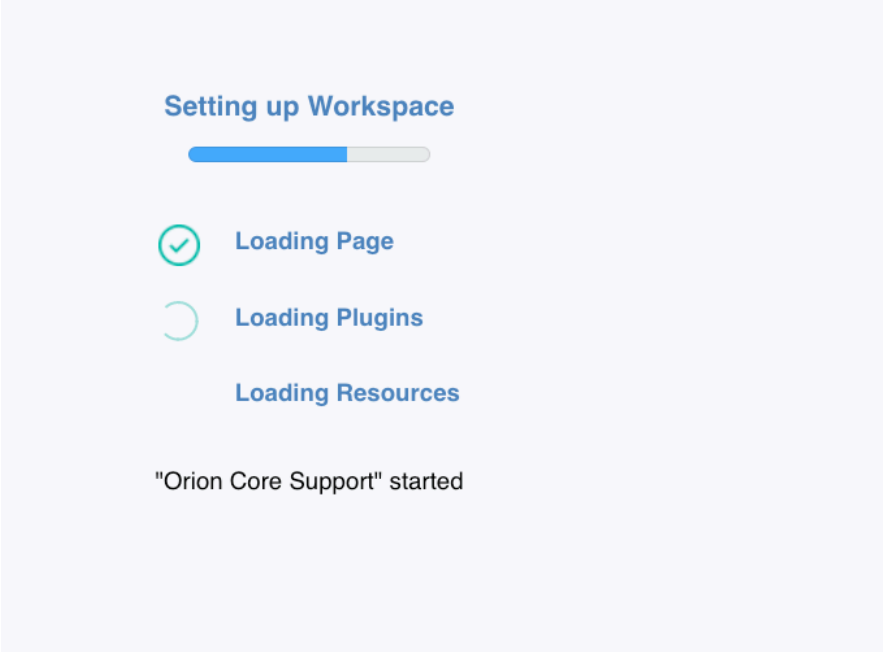
Now lets edit some files.

Click on the View Toolchain to access the the app associate toolchain

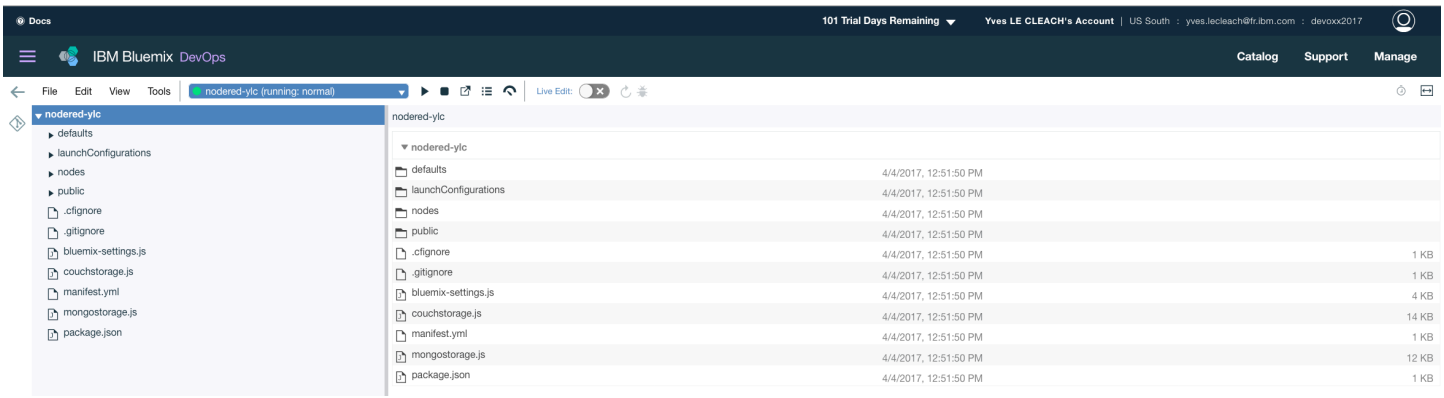
Copyright IBM Corporation 2017. All rights reserved.



Select the Eclipse Orion Web IDE to open a Cloud Editor linked to your GitHub project.

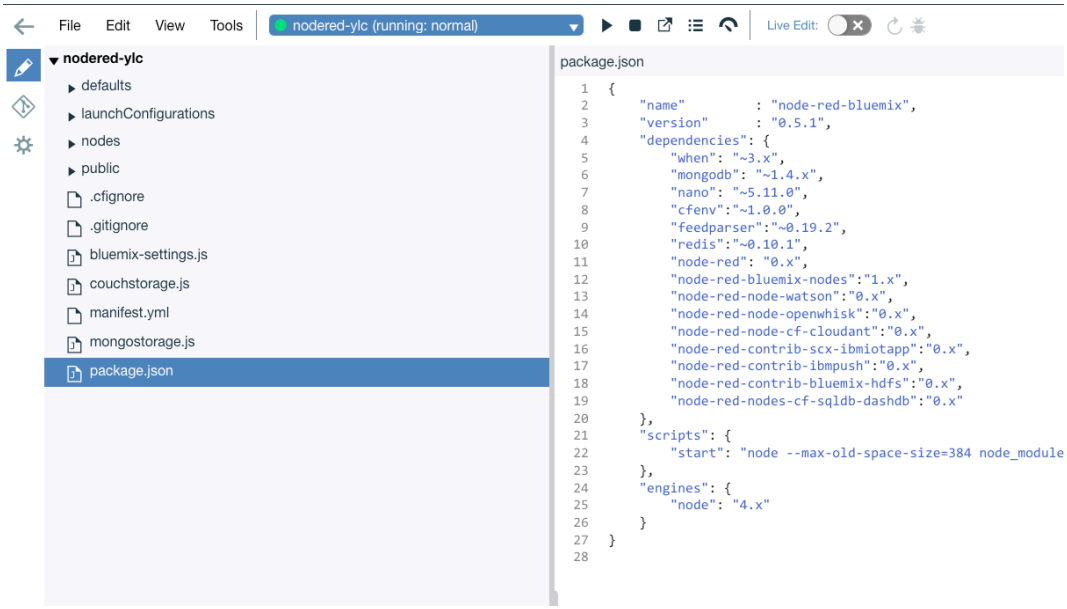


You will land to the Orion Web IDE ：



Copyright IBM Corporation 2017. All rights reserved.

Click on the package.json file to simplify edit it as follow :



In the “dependencies” list, add the following to your **package.json**, to pull in the extra nodes that are required for the exercices. (these nodes are not included by default in the Node-RED in Bluemix.

```
"node-red-contrib-browser-utils":"0.x",
"node-red-contrib-media-utils":"0.x",
"node-red-contrib-play-audio": "2.0.x",
"node-red-dashboard": "2.0.x",
"node-red-node-dropbox": "0.x"
```

Your **package.json** file should looks like to :

Copyright IBM Corporation 2017. All rights reserved.

package.json

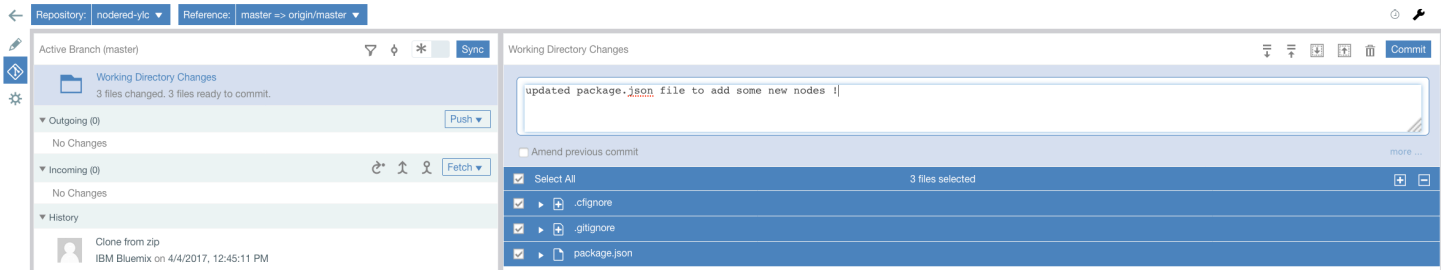
```
{
    "name"         : "node-red-bluemix",
    "version"      : "0.5.1",
    "dependencies": {
        "when": "~3.x",
        "mongodb": "~1.4.x",
        "nano": "~5.11.0",
        "cfenv":"~1.0.0",
        "feedparser":"~0.19.2",
        "redis":"~0.10.1",
        "node-red": "0.x",
        "node-red-bluemix-nodes":"1.x",
        "node-red-node-watson":"0.x",
        "node-red-node-openwhisk":"0.x",
        "node-red-node-cf-cloudant":"0.x",
        "node-red-contrib-scx-ibmiotapp":"0.x",
        "node-red-contrib-ibmpush":"0.x",
        "node-red-contrib-bluemix-hdfs":"0.x",
        "node-red-nodes-cf-sqldb-dashdb":"0.x",
        "node-red-contrib-browser-utils":"0.x",
        "node-red-contrib-media-utils":"0.x",
        "node-red-contrib-play-audio": "2.0.x",
        "node-red-dashboard": "2.0.x",
        "node-red-node-dropbox": "0.x"
    },
    "scripts": {
        "start": "node --max-old-space-size=384 node_modules/node-red/red.js --settings ./bluemix-settings.js -v"
    },
    "engines": {
        "node": "4.x"
    }
}
```

Lets commit this change so the app will be automatically redeployed by the default pipeline.

First, Save your change. (Control + S)

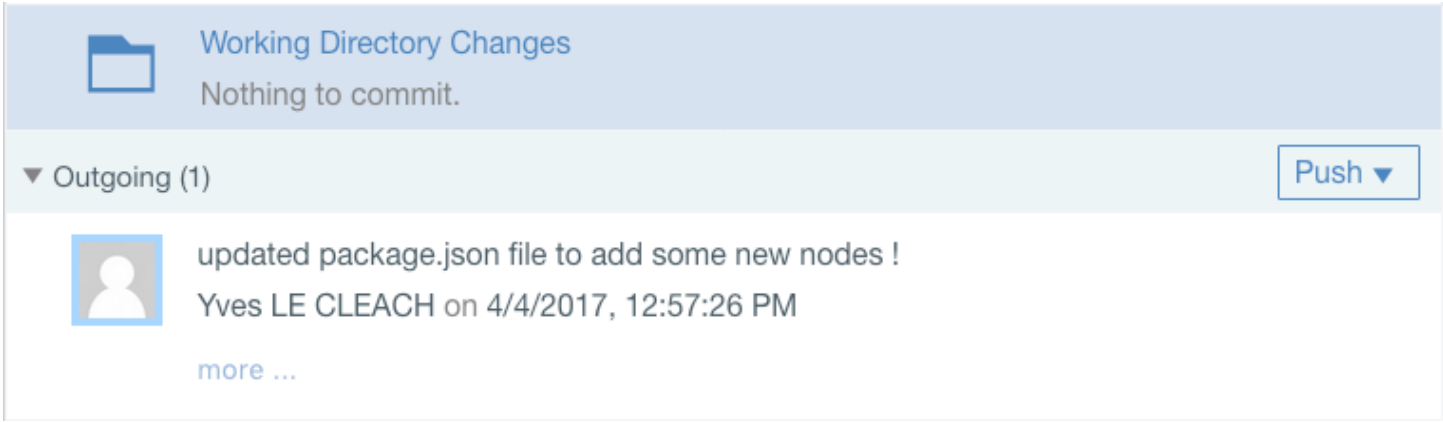
Then click on Git Icon  in the Left bar (icon just below the “Pencil” Icon used for Edit):

Enter a commit message, and click on the **Commit button** at the top right.

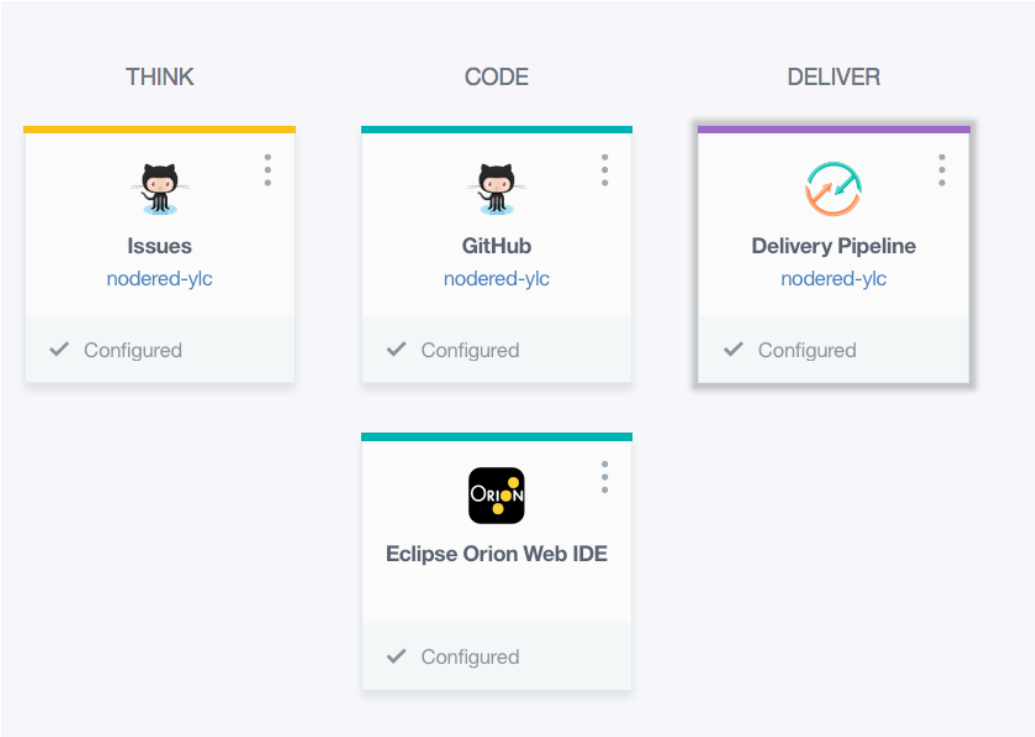


Then Click on the **Push button**

Copyright IBM Corporation 2017. All rights reserved.

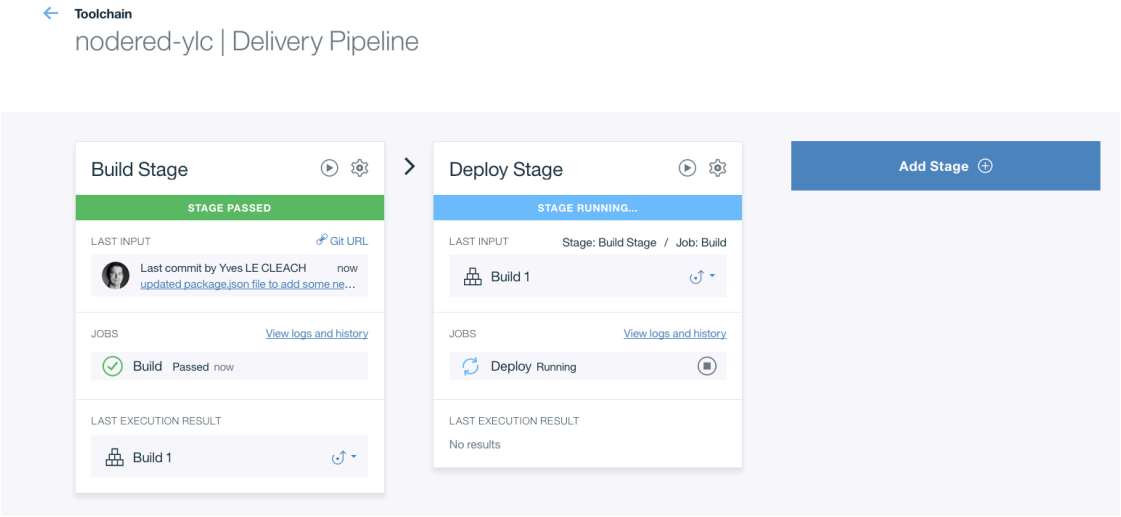


Now Go Back to your ToolChain and select the **Deliver Pipeline** created for you by default.

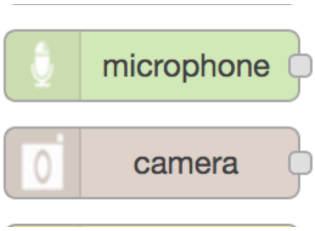


and see what's happened in the Deploy Pipeline: the Git Push has triggered a new Build Stage, then a Deploy Stage. You can have a quick look in logs to see what happened behind the scene.

Copyright IBM Corporation 2017. All rights reserved.



After 2-3 minutes your Node-RED app has been fully restaged, and you can check in your Node-RED Editor that you have well the new nodes.



Now you are ready to start working on Watson Labs on Node-RED … and you can continue on

- **Basic Labs** (section 3): if you are not familiar with Watson APIs or Watson nodes, or
- **Node Red Watson Starter Kits** (section 4): if you are already familiar with Watson APIs or nodes.

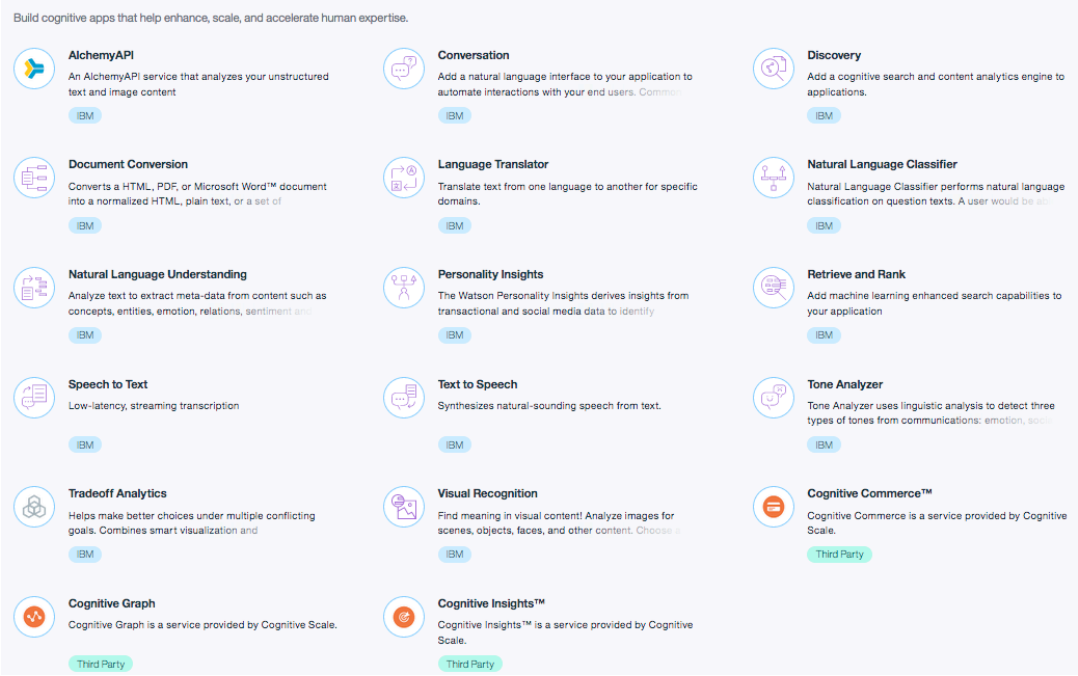
Copyright IBM Corporation 2017. All rights reserved.

3 BASICS Labs

The Watson Node-RED services lab will show you how you can use the Watson nodes in Node-RED.

https://github.com/watson-developer-cloud/node-red-labs/blob/master/basic_examples/README.md

Some Watson APIs are very simple to use, others are more complex with a training mode that must be configured prior using the corresponding API. Some others as Watson Language Translator service offers pre-defined domain specifics domains.



3.1 Watson Visual Recognition

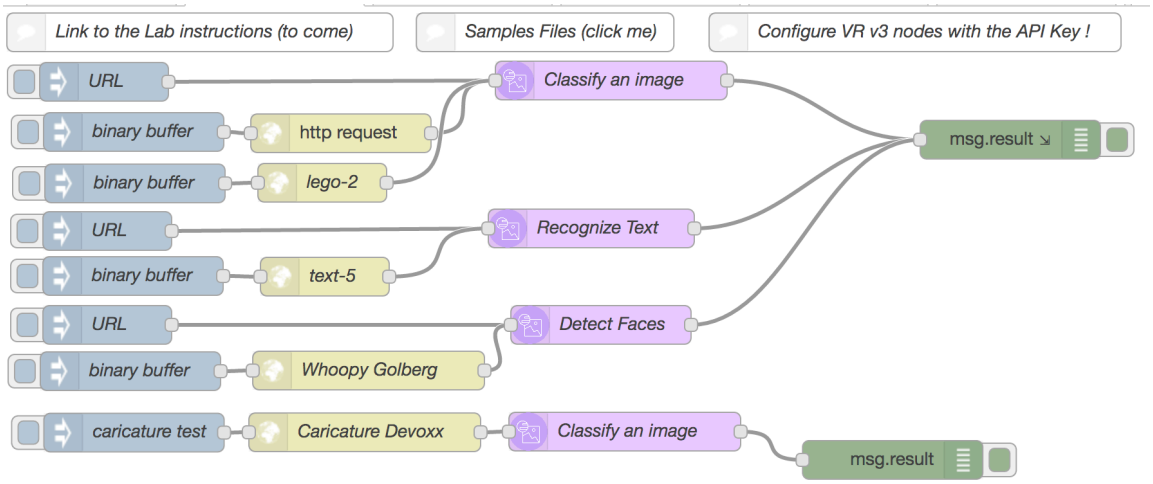
Using the Visual Recognition v3 node, you can use the Watson Visual Recognition service V3 to identify scenes, objects, faces, and text in images you upload to the service. You can create and train a custom classifier to identify subjects that suit your needs.

Firstly, connect a **Watson Visual Recognition** service instance to your app with the 'Free' plan using the "Connect new" button. A restage of the app is required.

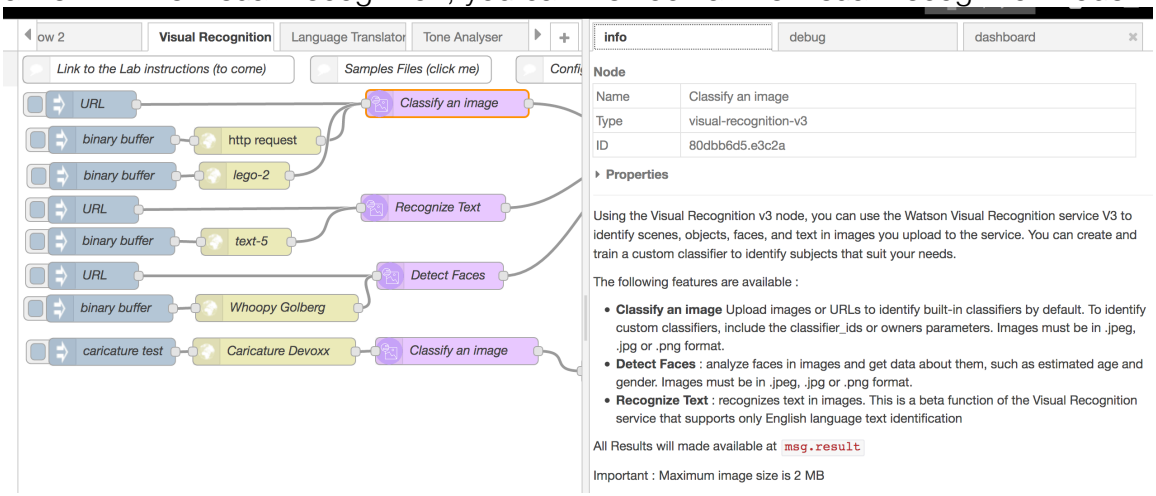
Then in a new Flow, Import the following flow : https://raw.githubusercontent.com/ylecleach/labs/master/node-red/basic-labs/visual-recognition/visual-recognition-sampleflow.json , and click on the Deploy button.

You should see the following Flow :

Copyright IBM Corporation 2017. All rights reserved.



To Familiarize with the Visual Recognition, you can first look at the Visual Recognition node info tab :



To Run the sample, simply click on the various inject button, and inpect the results in the Debug pane.

Documentation : https://www.ibm.com/watson/developercloud/doc/visual-recognition/index.html
API reference : https://www.ibm.com/watson/developercloud/visual-recognition/api/v3/

3.2 Watson Language Translator

This API can be used either without a training mode or with a training mode. As an example, the training mode of this service can be used to define language translation in a particular business domain. (e.g : in a car manufacturing domain, in a medical domain, …). The service is configured by default with a News domain, a Conversational domain (colloquialisms) and a Patent domain. Notice that Watson Language Translation supports translation to some Regional languages.

Translate and Identify language

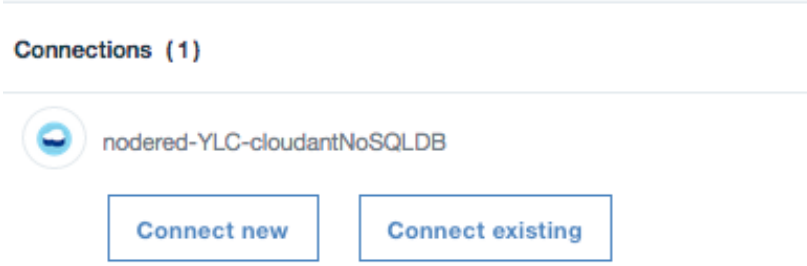
Follow the first part of the Lab, to test translation from EN to FR, and test the language identify node :
https://github.com/watson-developer-cloud/node-red-labs/tree/master/basic_examples/language_translator

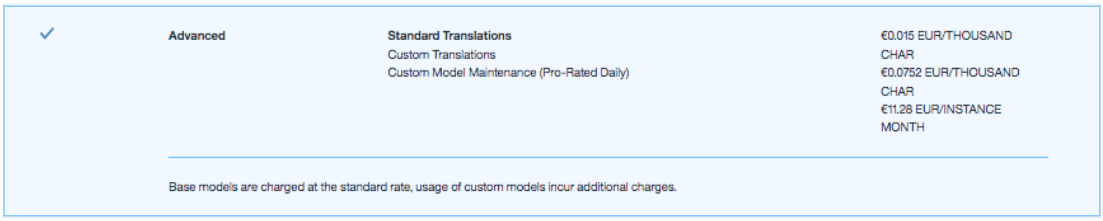
Customizing your domain mode

Follow the second part of the Lab to test the Training mode:
https://github.com/watson-developer-cloud/node-red-labs/tree/master/basic_examples/language_translator#customizing-your-domain

Copyright IBM Corporation 2017. All rights reserved.

Notice : do not forget to Connect a **Watson Language Translator** service instance to your app with the 'Advanced' plan using the "Connect new" button. A restage of the app is required.





Tips:

- You can use the Dropbox nodes as described in their respective documentations : Dropbox setup (do it once for all labs as the configuration is shared amongst Node-RED Dashboard tabs) : https://github.com/watson-developer-cloud/node-red-labs/tree/master/utilities/dropbox_setup

Documentation : https://www.ibm.com/watson/developercloud/doc/language-translator/index.html
API Reference : https://www.ibm.com/watson/developercloud/language-translator/api/v2/

3.3 Watson Tone Analyser service

People show various tones, such as joy, sadness, anger, and agreeableness, in daily communications. Such tones can impact the effectiveness of communication in different contexts. Tone Analyzer leverages cognitive linguistic analysis to identify a variety of tones at both the sentence and document level. This insight can then used to refine and improve communications. It detects three types of tones, including emotion (anger, disgust, fear, joy and sadness), social propensities (openness, conscientiousness, extroversion, agreeableness, and emotional range), and language styles (analytical, confident and tentative) from text.

Follow the following Basic Lab :
https://github.com/watson-developer-cloud/node-red-labs/tree/master/basic_examples/tone_analyser

Notice : do not forget to Connect a new **Watson Tone Analyser** service instance to your app using the "Connect new" button. A restage of the app is required.

Documentation : https://www.ibm.com/watson/developercloud/doc/tone-analyzer/index.html
API Reference : https://www.ibm.com/watson/developercloud/tone-analyzer/api/v3/

Copyright IBM Corporation 2017. All rights reserved.

3.4 TTS : Text to Speech

Designed for streaming low-latency synthesis of audio from written text. The service synthesizes natural-sounding speech from input text in a variety of languages and voices that speak with appropriate cadence and intonation.

Follow the following Basic Lab :
https://github.com/watson-developer-cloud/node-red-labs/tree/master/basic_examples/text_to_speech

Notice : do not forget to Connect a new **Watson Text to Speech** service instance to your app using the "Connect new" button. A restage of the app is required.

Documentation : https://www.ibm.com/watson/developercloud/doc/text-to-speech/
API reference : https://www.ibm.com/watson/developercloud/text-to-speech/api/v1/

Copyright IBM Corporation 2017. All rights reserved.

4 STARTER KITS Labs

Node-RED Starters Kits are prebuilt applications from which to start your own prototypes. The apps are starter kit examples of how to create apps using Node-RED and the Watson Nodes

These materials have already been used in many hackathons, Watson Developer Conferences, and World of Watson conference. The starter apps are fully functioning applications that use the Watson Services and SDKs in Node-RED. The exercises make use of multiple Watson APIs.

Node-RED Watson Starter Kits are apps exploring the capabilities of Watson Developer Cloud in Node-RED. Each kit comes with a video introduction, which shows each application in use.

List of availables Starter Kits:

https://github.com/watson-developer-cloud/node-red-labs/tree/master/starter-kits

For this Hands On Lab session, I propose you to follow one of the two starter kits labs (Ok Watson, or Selfie Training).

4.1 OK Watson

Learn how built a smart bot able to converse with you and able to adapt the conversation based on the tone of the conversation (anger, happy). This starter highlights the Watson Conversation and Tone Analyser APIs

Video Introduction : https://www.youtube.com/watch?v=4L3CjH_f58I

Start by create and connect to your Node-RED app the 4 Watson services :

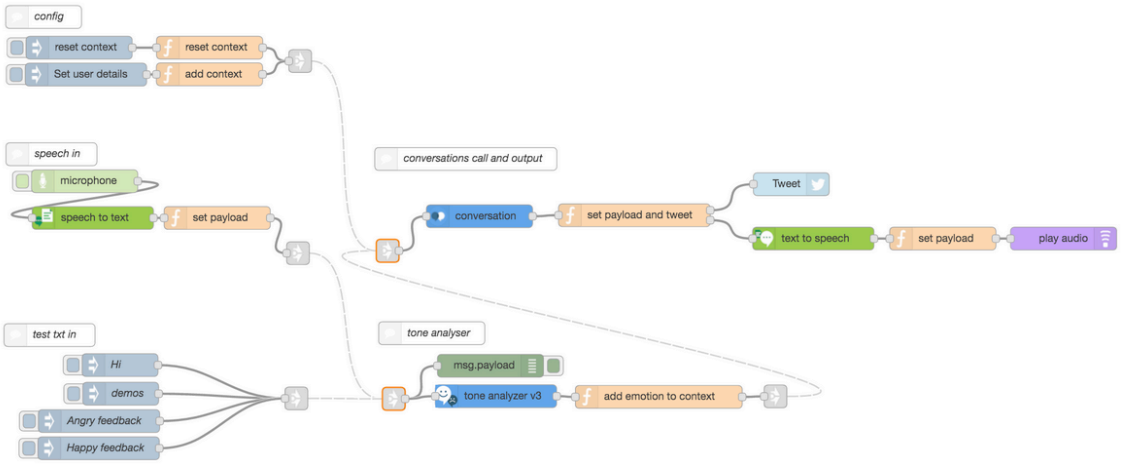
- **Speech to Text**
- **Text to Speech**
- **Conversation (Choose Free plan)**
- **Tone Analyser**

Tips1 : to avoid multiple Restaging, Connect the following Watson services but choose “Cancel” when prompted. Restage only for the last service.

Tips2 : If you use the Twitter node, you would prefer use a different twitter account than your personal one. Just Login to Twitter on the browser with which you use Node-RED with the desired twitter account.

Lab instructions:
https://github.com/watson-developer-cloud/node-red-labs/blob/master/starter-kits/ok_watson/README.md
Notice: you do not need to configure Watson service credentials as mentioned in this lab as the corresponding nodes are auto-discovering them through Connections. (also known as Binding).

Copyright IBM Corporation 2017. All rights reserved.



4.2 Selfie Training

Learn how using the Watson Visual Recognition API to build an application able to detect similarities in faces.

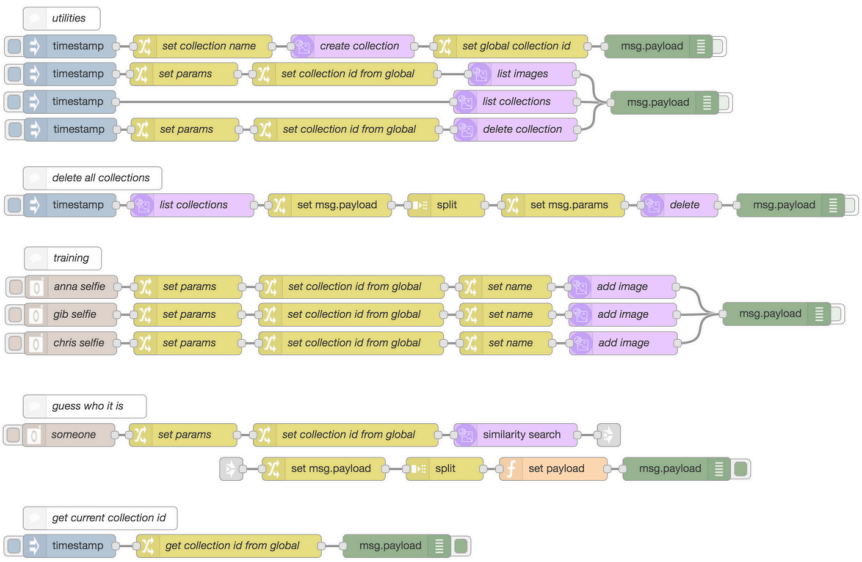
Video Introduction : https://youtu.be/f4atHS0El2k

Start to create and connect the following Watson service and restage :

- **Visual Recognition**

Tips2 : do not forget to activate the Debug nodes to see logs. (dark green nodes)

Lab instructions :
https://github.com/watson-developer-cloud/node-red-labs/tree/master/starter-kits/selfie_training



5 Learn how to build a Node-RED node

Copyright IBM Corporation 2017. All rights reserved.

5.1 Build your first node exercice

The aims of this exercise is to learn the basics on how to build a Node-RED node. The following example defines a simple node that converts a message payload into all lower-case characters.

You can start to read **:** **http://nodered.org/docs/creating-nodes/**

Ready ? then check first setup your development environnment (tested on MacOs X Sierra)

Pre-Requisites (3 min)

- **install a node version manager : https://github.com/creationix/nvm**

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.1/install.sh | bash
```

Tips if error: I had to download the install.sh script, and to replace the first line with : `#!/bin/bash`

Notice: do not forget to open a new shell after the installation of nvm

```
nvm install 4.5  (install of the Node.js 4.5)
nvm use 4.5  (configure nvm to use a given version of node)
node –v   (verify it returns "v4.5.0")
```

- **create a working directory and install Node-RED**

```
mkdir ~/dev/devoxx ; cd ~/dev/devoxx
npm install node-red   (in dev mode, I do not recommand to install something with the option -g (global))
cd node_modules/node-red
npm start
```

- **Open** http://localhost:1880 to verify that Node-RED is starting well from your laptop.

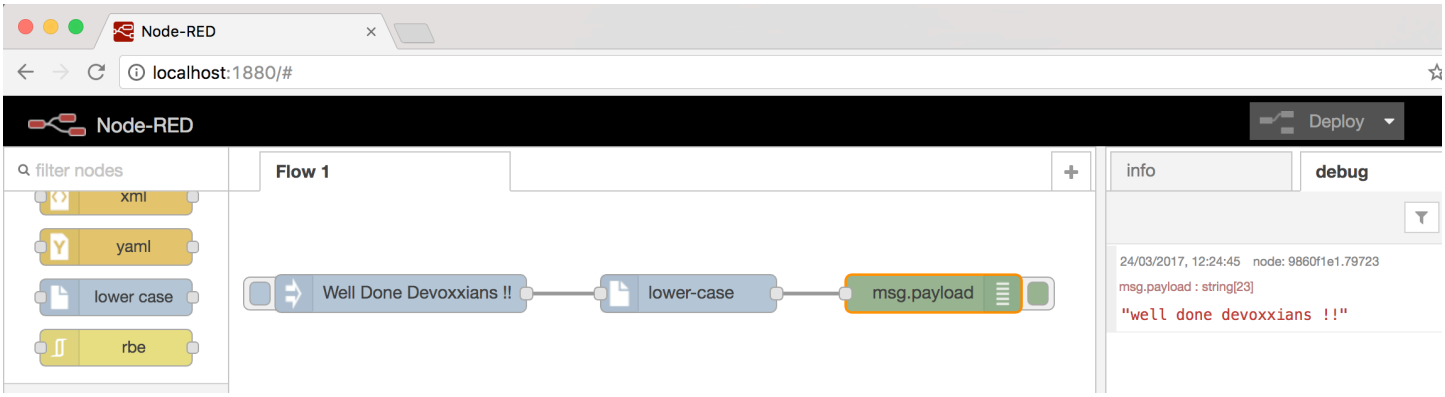
Build your first node step by step (< 1 min)

This step by step is inspired from **:** http://nodered.org/docs/creating-nodes/first-node that I encourage you to read quickly before.

```
cd ~/.node-red   (another option was to create the node in the node_modules/node-red/nodes  directory)
mkdir nodes
cd nodes
wget https://raw.githubusercontent.com/ylecleach/labs/master/devoxx2017/lower-case.html
wget https://raw.githubusercontent.com/ylecleach/labs/master/devoxx2017/lower-case.js
```

- **Restart your node-red app**, and play with your new "lower case" node you just have created. (Look under the "function" category in the Node-RED palette.
- Optional : You can do some minor updates in the html or js files and see the result quickly by restarting

Copyright IBM Corporation 2017. All rights reserved.



CONGRATULATIONS : you have build your first Node-RED node, and you are on the path to be a Watson Node-RED contributor !

Copyright IBM Corporation 2017. All rights reserved.

5.2 Deep dive on a Watson node (Visual Recognition)

Lets continue and now we assume you want to contribute to the Watson nodes.

In this section you will :
- setup your laptop to update the Watson Node-RED open source nodes as done by Contributors.
- discover the Watson Visual Recognition node
- test locally the node, using the API key of a Watson Visual Recognition service instance
- learn about the source code of the Watson Visual Recognition (v3) nodes.

How update / contribute to the Watson nodes

This lab is re-using instructions in the Watson Node-RED project , in the CONTRIBUTING.md file.

Here are the steps to successfully setup your development environment to contribute to this project

- Fork the main Node-RED Watson project using your GitHub account (ex: @ylecleach)
- Create a work directory that will contains the source code of your fork

```
mkdir ~/dev/watson_nodes ; cd ~/dev/watson_nodes
git clone https://github.com/ylecleach/node-red-node-watson
```

Notice : replace ylecleach by your own GitHub ID

- create a npm link to your forked project. This will also build this project dependencies.

```
cd ~/dev/watson_nodes/node-red-node-watson
npm link
```

From this point, we assume your already installed Node-RED on your laptop (see section 5.1)

- **Install your fork project into local Node-RED using npm link**:

```
cd ~/dev/devoxx          (cd to where you have installed your local Node-RED version)
npm link node-red-node-watson
```

- **Restart your local Node-RED and verify you have Watson nodes in the palette** :

```
cd ~/dev/devoxx/node_modules/node-red
npm start
```

- **Open** http://localhost:1880 **and verify that Watson nodes are available in the Node-RED palette**

Anatomy of a node : the Watson Visual Recognition node

This watson node is referenced in the following file :

```
~/dev/watson_nodes/node-red-node-watson/package.json
```

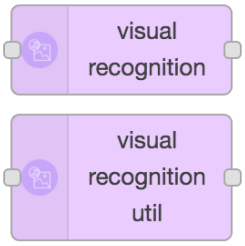
and is imported through this simple line :

Copyright IBM Corporation 2017. All rights reserved.

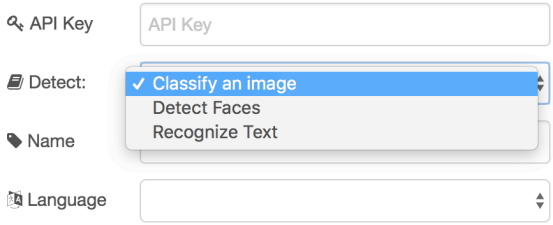
```
"watson-visual-recognition-v3": "services/visual_recognition/v3.js",
```

This module defines 2 nodes. (visual recognition & visual recognition util nodes).

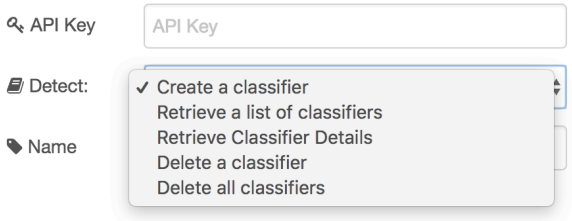
Lets first discover those 2 nodes ! In the Node-RED Editor, create a new Flow, and drag and drop the 2 followings node :



- The **"visual recognition" node** : offers 3 functions (classify an image, detect faces, recognize text)



- The **"visual recognition util" node** : it offers all "Training" functionalities that allow to create specific Classifiers for Image Recognition.



The code of the Watson Visual Recognition node (and its Util node) are under the following directory :

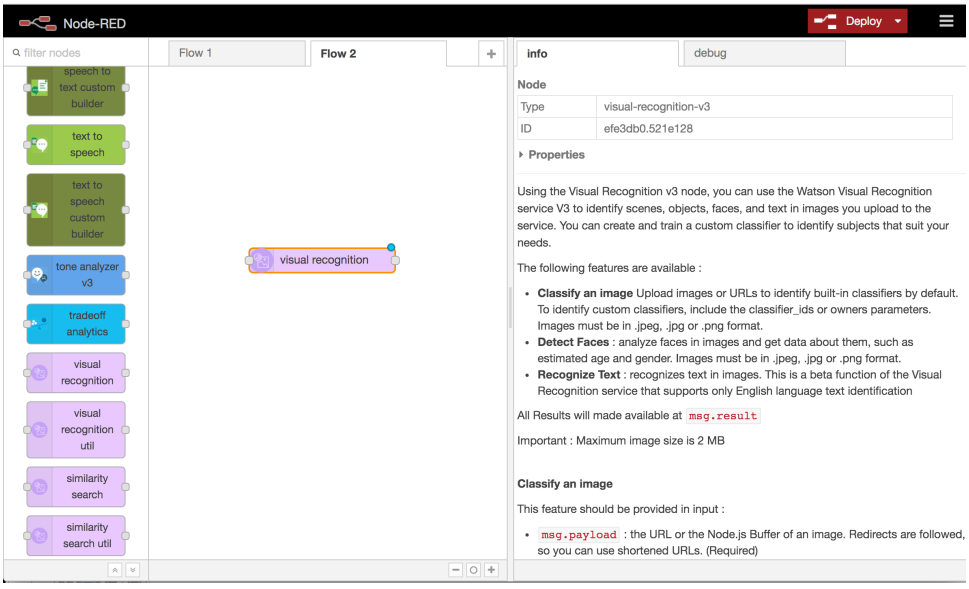
```
ls ~/dev/watson_nodes/node-red-node-watson/services/visual_recognition

total 112
drwxr-xr-x  6 lecleach  staff    204 Mar 24 15:57 icons
-rw-r--r--  1 lecleach  staff  15389 Mar 24 15:57 v3.html
-rw-r--r--  1 lecleach  staff  12256 Mar 24 15:57 v3.js
```

The icons directory is the default name to hold the node icon. The icon filename is referenced in the HTML file of the node. (e.g. v3.html).

Now open in your favorite editor, and open the files: `v3.html` and `v3.js`

In parallel in your Flow Editor, click on the info tab of those two nodes and read quickly theirs functionalities in order to be able to understand the following source code explanations.

Copyright IBM Corporation 2017. All rights reserved.



V3.HTML

Now you can read across the code of the `v3.html` file and you will find all the necessary HTML + Javascript code that were necessary for the UI part of the node such as :

- node definition (label, icon)
- node UI dynamic control. (hide/show credentials).
- Node information (description, inputs and output format for each functionality

V3.JS

This is the server-side code of the 2 nodes which is written in Node.js code as all Node-RED nodes.

- L17 : The node is wrapped as a node module. The module exports a function that gets called when the runtime loads the node on start-up. The function is called with a single argument, RED, that provides the module access to the Node-RED runtime api.
- L366 + L372 : declaration of the 2 nodes in Node-RED registers, using the same “WatsonVisualRecognitionV3Node” function.
- L338 : WatsonVisualRecognitionV3Node : this is the main point where you define the Node. In this definition you will find two important function calls :
- L340 : the RED.nodes.createNode function which initialize the feature shared by all ‘WatsonVisualRecognitionV3Node’ nodes, and which is used to instantiate the node every time a Flow is Deployed. A node instance is deleted and re-created each time the next Flow is deployed.
- L343 : this line register a ‘input’ event on the node instance, and is called when a message arrive to the node. From this line, several others functions are defined. (verifyPayload, verifyInputs, verifyServiceCredentials, execute).

Watson nodes are using Watson Node.js SDK

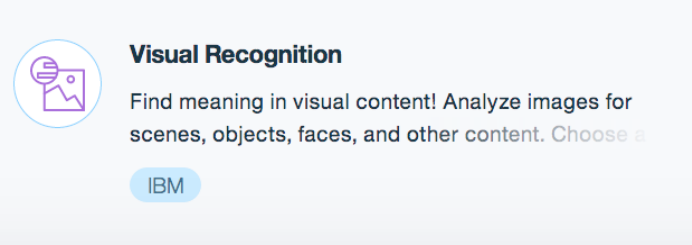
All Watson Node-RED nodes are using the **Watson Node.js SDK** also available on GitHub for your Node.js project. This library is available as a NPM module, and it simplifies drastically the way the Watson Node-RED nodes are written using less code, with no direct call to a given API.

Watson Node.js SDK on GitHub : https://github.com/watson-developer-cloud/node-sdk

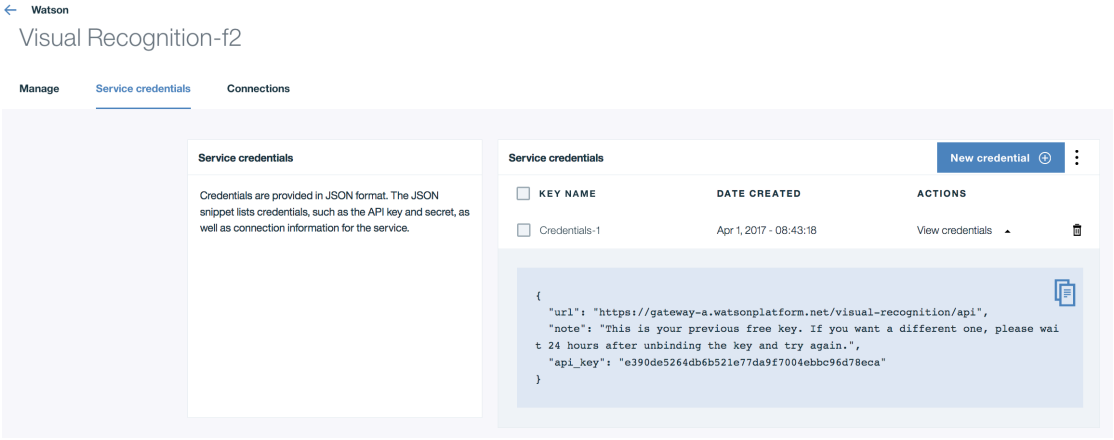
Copyright IBM Corporation 2017. All rights reserved.

How Test locally this node

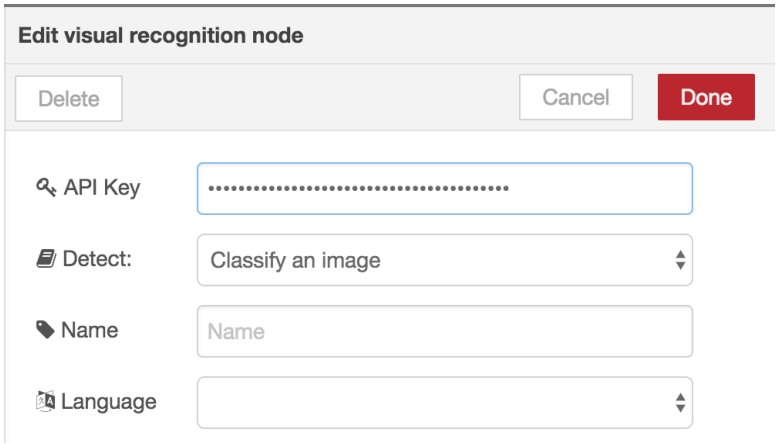
- create a Watson Visual Recognition service instance in your Bluemix Dashboard



- After the service is created, click on the Service Details, and create a new Credential to access to the API key.



- Enter in the “visual recognition node” the API key, click on Done, then click on the red Deploy button in the Node-RED Editor.



Now you are able to use the all Watson Node-RED nodes on http://localhost:1880

Notice : the Watson nodes are currently designed to be runnable in the 2 modes :

- on node-RED on Bluemix : the api key is automatically detected by the node code.
- on node-RED on any other platform : small devices (e.g. Raspberry) or any other cloud. This approach entails you have to configure yourself nodes by entering their API Key.

Reference Watson Visual Recognition

Introduction: https://www.ibm.com/watson/developercloud/visual-recognition.html
Documentation : https://www.ibm.com/watson/developercloud/doc/visual-recognition/index.html
API : https://www.ibm.com/watson/developercloud/visual-recognition/api/v3/

Reference Watson Node.js SDK

GitHub : https://github.com/watson-developer-cloud/node-sdk

Copyright IBM Corporation 2017. All rights reserved.

How to contribute to Watson Node-RED nodes

- You can develop sample flows that make use of the Watson services in innovative solutions. You can publish them as either basic examples, advanced examples, or starter kits on our documentation repository. This will allow you to share your samples to the community building Watson-powered applications on Node-RED.
- You can write new nodes for new services. Visit our GitHub repository is to find the current code and make contributions : https://github.com/watson-developer-cloud/node-red-node-watson
- Follow instructions in https://github.com/watson-developer-cloud/node-red-node-watson/blob/master/CONTRIBUTING.md

Copyright IBM Corporation 2017. All rights reserved.

5.3 Deep dive on a Bluemix node (Business Rules)

Why are we talking of this Business Rules service in a Watson Node-RED Lab ?

1) I have contributed to the code of the Business Rules Node-RED node ;-) This node implementation is using a different approach than in the Watson Node-RED nodes, as it is using directly the REST API proposed by the Business Rules service.
2) The Business Rules service offers you to inject intelligence in your applications, by enabling you to automate your critital business decisions, leveraging enterprise-grade, performant, reliable, technologies used by IBM customers for more than a decade. Business users can author and design the business rules using a natural langage. From an AI perspective, Business Rules adresses the Decision discipline which is needed for decision-making.

Thus, from this perspective the Business Rules service is a complementary tool with Watson Cognitive APIs to build an application that is able to learn from structured and non structured data, to interact with human languages, and to automate any frequent and repeatable business decisions

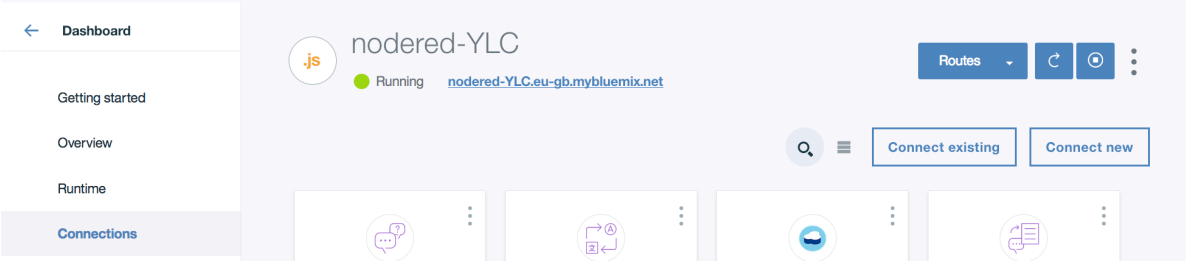
Lets Discover the Business Rules Node-RED node

The Business Rules node is part of the Bluemix nodes which are available only on Node-RED on Bluemix, and their code is also hosted on github.

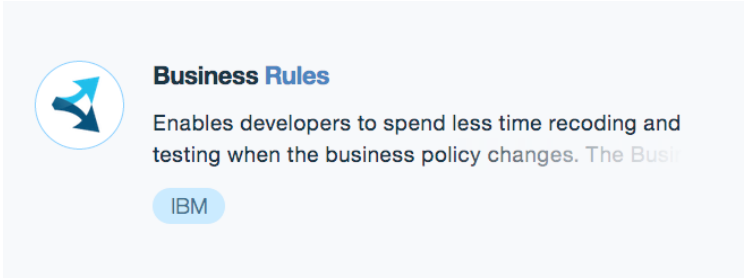
Bluemix Node-RED nodes : https://www.npmjs.com/package/node-red-bluemix-nodes
GitHub Source Code : https://github.com/ibmets/node-red-bluemix-nodes

In the Bluemix Dashboard, select your Node-RED app, and click on it to access your app details

In the Connections tab in the left menu, click on the “Connect new” button

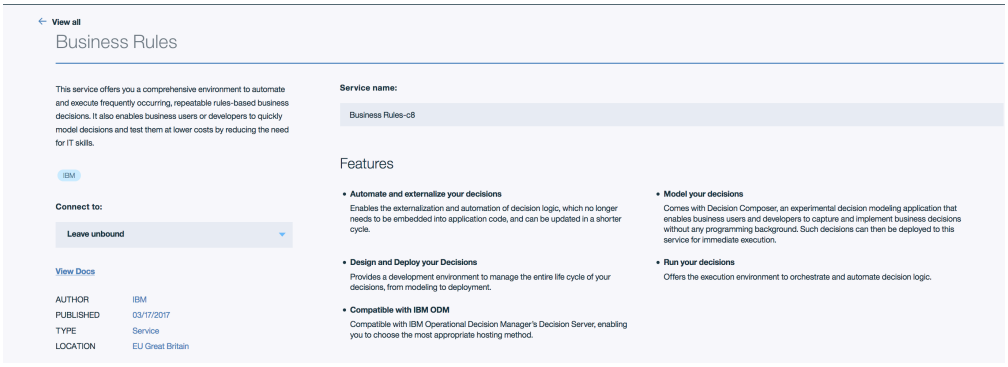


In the displayed Catalog, type “rules” in the search field, and click on the Business Rules tile :



Copyright IBM Corporation 2017. All rights reserved.

You can read the Business Rules service Feature and watch the short videos provided in the Images section.



Click on the Create button, and restage your Node-RED app when asked.

How update / contribute to the Bluemix nodes (including the Business Rules node)

Here are the steps to successfully setup your development environment to contribute to this project

- Fork the following project using your GitHub account (ex: @ylecleach) : https://github.com/ibmets/node-red-bluemix-nodes
- Create a work directory that will contains the source code of your fork

```
mkdir ~/dev/bluemix_nodes ; cd ~/dev/bluemix_nodes
git clone https://github.com/ylecleach/node-red-bluemix-nodes
```

Notice : replace ylecleach by your own GitHub ID

The code of the Business Rules node is under the following directory :

```
ls ~/dev/bluemix_nodes/node-red-bluemix-nodes/businessrules

total 40
drwxr-xr-x  3 lecleach  staff   102 Apr  3 16:25 icons
-rw-r--r--  1 lecleach  staff  8076 Apr  3 16:25 v1.html
-rw-r--r--  1 lecleach  staff  9576 Apr  3 22:23 v1.js
```

You can have a quick look at v1.js and v1.html. You can observe the main difference with the previous node : **Business Rules node is using some of the Business Rules service REST APIs. (no Node.js SDK available), and it integrates the Decision Services test tool. (Run button on the Test feature of the node).**

- create a npm link to your forked project. This will also build this project dependencies.

```
cd ~/dev/bluemix_nodes/node-red-bluemix-nodes
npm link
```

From this point, we assume your already installed Node-RED on your laptop (see section 5.1)

- **Install your fork project into local Node-RED using npm link**:

Copyright IBM Corporation 2017. All rights reserved.

```
cd ~/dev/devoxx          (cd to where you have installed your local Node-RED version)
npm link node-red-bluemix-nodes
```

- **Restart your local Node-RED and verify you have Bluemix nodes in the palette (e.g mongodb, cloudant, Business Rules node)**:

```
cd ~/dev/devoxx/node_modules/node-red
npm start
```

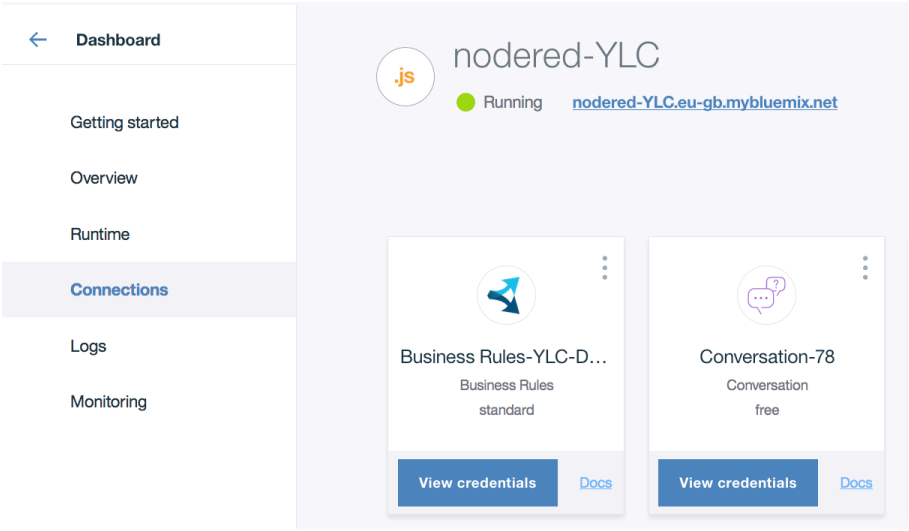
- **Open** http://localhost:1880 **and verify that Bluemix nodes are available in the Node-RED palette, such as the Business Rules node.**

How Test locally the Business Rules node

The business rules nodes has been designed to run only in the context of Bluemix, as on the opposite of Watson APIs that can be used outside of the context of Bluemix, the Business Rules on Bluemix service only exists on … Bluemix ;-)

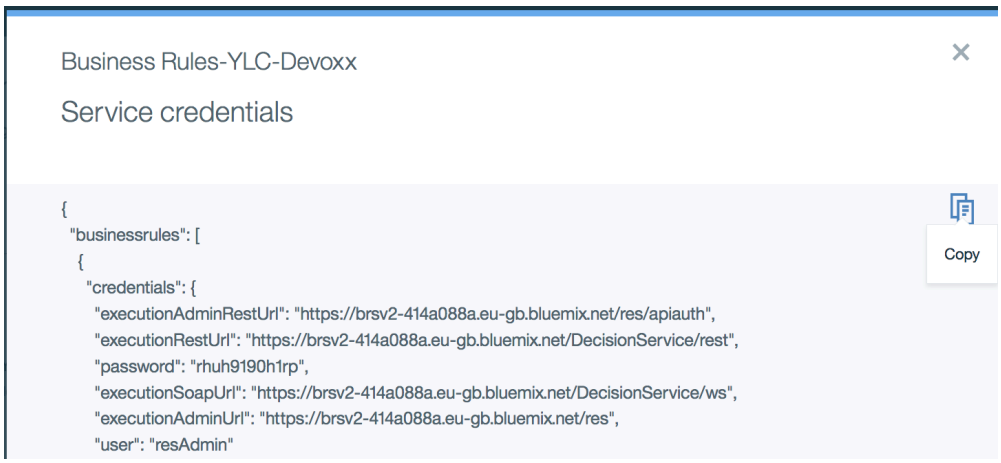
Lets see the trick I use regularly to develop nodes locally to simulate a bluemix environment.

First, open the Bluemix Dashboard, and select your Node-RED app, and your Business Rules instances details :



Click on the "View credentials" button of your Business Rules service instance which is connected to your Node-RED app. You should see :

Copyright IBM Corporation 2017. All rights reserved.



Now click on the copy button at the right to put the service credentials in the clipboard.
As this credentials is in Json format, we have to transform it into a string representation.

Here is a very useful hint to learn for your node.js developments. Now, enters in a shell the following command.

```
node    (this open a node interactive shell in which you can run Node.js code)
>JSON.stringify(CMD+V) (then RETURN)
```

You should see a string version of the service credentials, copy it with a CMD+C. (with enclosing simple quotes)

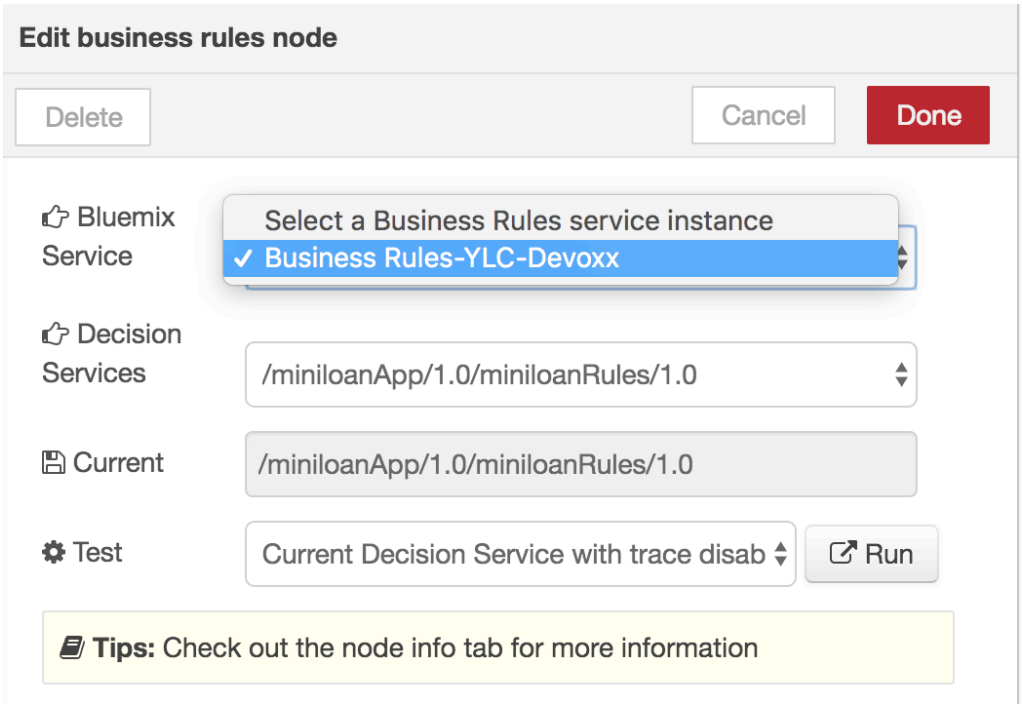
```
... })
'{"businessrules":[{"credentials":{"executionAdminRestUrl":"https://brsv2-414a088a.eu-gb.bluemix.net/res/apiauth","executionR
estUrl":"https://brsv2-414a088a.eu-gb.bluemix.net/DecisionService/rest","password":"rhuh9190h1rp","executionSoapUrl":"https:/
/brsv2-414a088a.eu-gb.bluemix.net/DecisionService/ws","executionAdminUrl":"https://brsv2-414a088a.eu-gb.bluemix.net/res","use
r":"resAdmin"},"syslog_drain_url":null,"label":"businessrules","provider":null,"plan":"standard","name":"Business Rules-YLC-D
evoxx","tags":["web_and_app","ibm_created","ibm_dedicated_public"]}]}'
```

Now update the v1.js file of the business rules node as follow by replacing the line L21 by :

```
//vcap = JSON.parse(process.env.VCAP_SERVICES || "{}"),
 vcap = JSON.parse(CMD+V),
```

Save the v1.js file, and restart your Node-RED server and verify you can access from your localhost the distant Business Rules service instance hosted on Bluemix from the Configure node panel of the Business Rules node :

Copyright IBM Corporation 2017. All rights reserved.

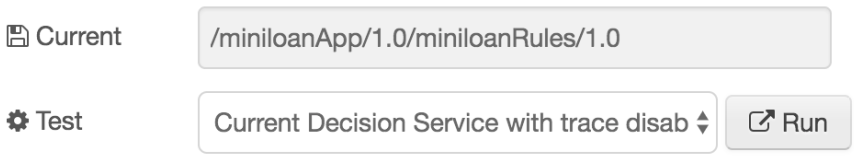


To See how deploy Decision Services on your Business Rules service instance, please consult the Reference documentation provided below.

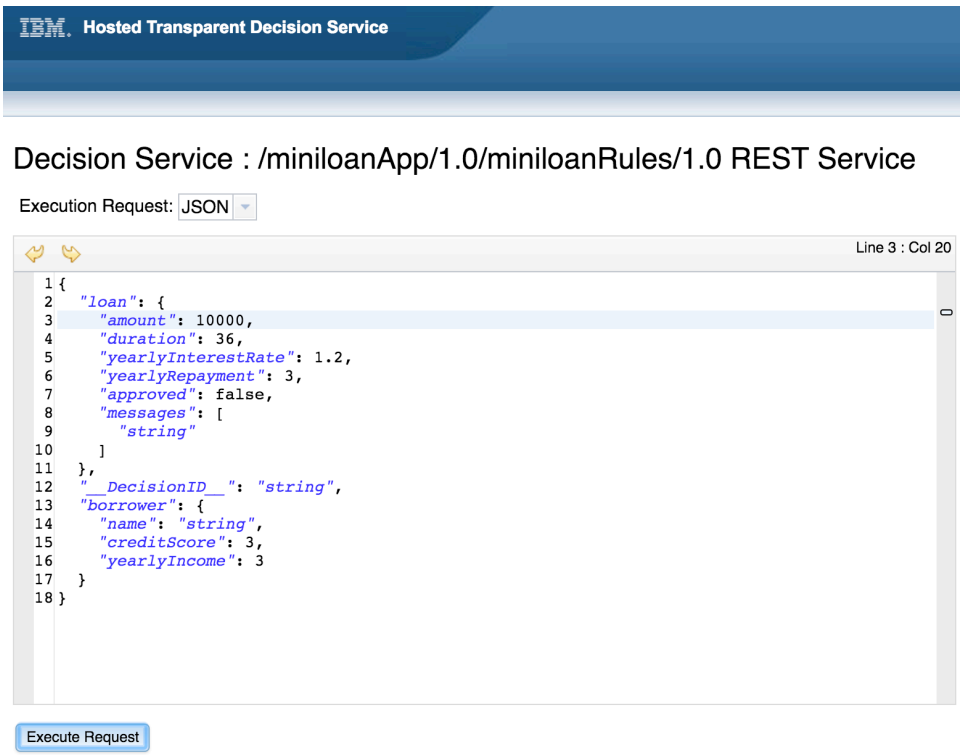
Alternatively, and only for Devoxx France 2017 event, you can re-use for this lab this configuration to speed up, and be able to play quickly with some pre-installed rulesets :

```
vcap =
JSON.parse('{"businessrules":[{"credentials":{"executionAdminRestUrl":"https://brsv2-
414a088a.eu-gb.bluemix.net/res/apiauth","executionRestUrl":"https://brsv2-414a088a.eu-
gb.bluemix.net/DecisionService/rest","password":"rhuh9190h1rp","executionSoapUrl":"http
s://brsv2-414a088a.eu-
gb.bluemix.net/DecisionService/ws","executionAdminUrl":"https://brsv2-414a088a.eu-
gb.bluemix.net/res","user":"resAdmin"},"syslog_drain_url":null,"label":"businessrules",
"provider":null,"plan":"standard","name":"Business Rules-YLC-
Devoxx","tags":["web_and_app","ibm_created","ibm_dedicated_public"]}]}'),
```

You can now test the installed Decision services on the Business Rules service instance.
Click on the Run button in the Business Rules node Configure panel.



Copyright IBM Corporation 2017. All rights reserved.



Reference Documentation on Business Rules :

Business Rules for Bluemix service documentation :
https://console.ng.bluemix.net/docs/services/rules/rules.html#rules

Business Rules for Bluemix service documentation (MiniLoan sample) :
https://console.ng.bluemix.net/docs/services/rules/index-gentopic5.html#ruleov006

Business Rules Node-RED node documentation :
https://github.com/ylecleach/labs/tree/master/node-red/basic-labs/business-rules

Business Rules Node-RED node tutorial :
https://github.com/ylecleach/labs/blob/master/node-red/basic-labs/business-rules/tv-series-finder.md

Try the new Decision Composer (experimental) : http://ibm.biz/DecisionComposer to author your Decisions Services in the Cloud. This is a new (cloud) way to author Decision Services, additionally to the Rule Designer (Eclipse based tool).

Copyright IBM Corporation 2017. All rights reserved.

6 Additional Resources

Bluemix :
- Login / Doc / Pricing **: https://bluemix.net**
- Bluemix Help **: https://ibm.biz/bluemixhelp** (Main, Support, Sales, Technical, Security, Training)

Watson on Node-RED :
DeveloperWorks Open : https://developer.ibm.com/open/openprojects/watson-on-node-red/
Watson Node-RED Nodes: https://github.com/watson-developer-cloud/node-red-node-watson
Watson Node-RED Labs: https://github.com/watson-developer-cloud/node-red-labs
Watson Node-RED Starter Kits : https://github.com/watson-developer-cloud/node-red-labs/tree/master/starter-kits
Watson Node-RED Bluemix Boilerplate: https://github.com/watson-developer-cloud/node-red-bluemix-starter

Node-RED (open source and project under Node.js foundation) :
- Introduction to Node-RED: https://ibm.biz/BdHvfc
- Node-RED documentation : http://nodered.org/docs/
- Node-RED forum : https://groups.google.com/forum/#!forum/node-red

You can find additional resources on Watson :
- Watson Developer Cloud : main documentation on Watson APIs
- Watson API Explorer : useful for developers to discover Watson REST API
- Watson Starter Kits: many watson nice demos
- Watson Developer Cloud Node.js SDK: the Node.js SDK for Watson – used by Watson nodes.

Want to Meetup People on Paris, and share your Bluemix or Watson Projects experiences in France

Bluemix/Watson communities in France :
- Bluemix Paris Meetup : http://ibm.biz/BluemixParisMeetup - (2700+ members)

Follow me on :

- Twitter : https://twitter.com/ylecleach
- GitHub : https://github.com/ylecleach
- LinkedIn : https://www.linkedin.com/in/yveslecleach

Follow IBM France Lab :
- Twitter : https://twitter.com/IBMFranceLab
- Slideshare : http://fr.slideshare.net/IBMFranceLab (all french Bluemix Meetup slides)

Copyright IBM Corporation 2017. All rights reserved.

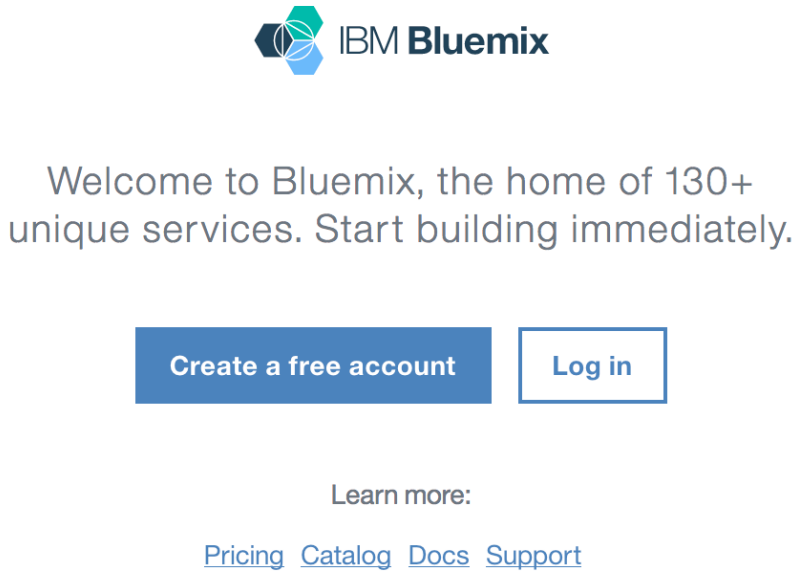
7 Annexes

7.1 Create your Bluemix account

IBM Bluemix is a cloud Platform as a service (PaaS) which supports several programming languages (Java, Node.js, Go, PHP, Python, Ruby...) and services as well as an integrated software development method called **DevOps toolchain** to build, run, deploy and manage applications easily on the cloud. Bluemix is based on Cloud Foundry open technology and runs on a SoftLayer infrastructure.

*If you already have a Bluemix account, connect with your **IBM ID** and your Bluemix password. If not, please follow the instructions.*

1. _ Connect to https://bluemix.net and click on "Create a free account"



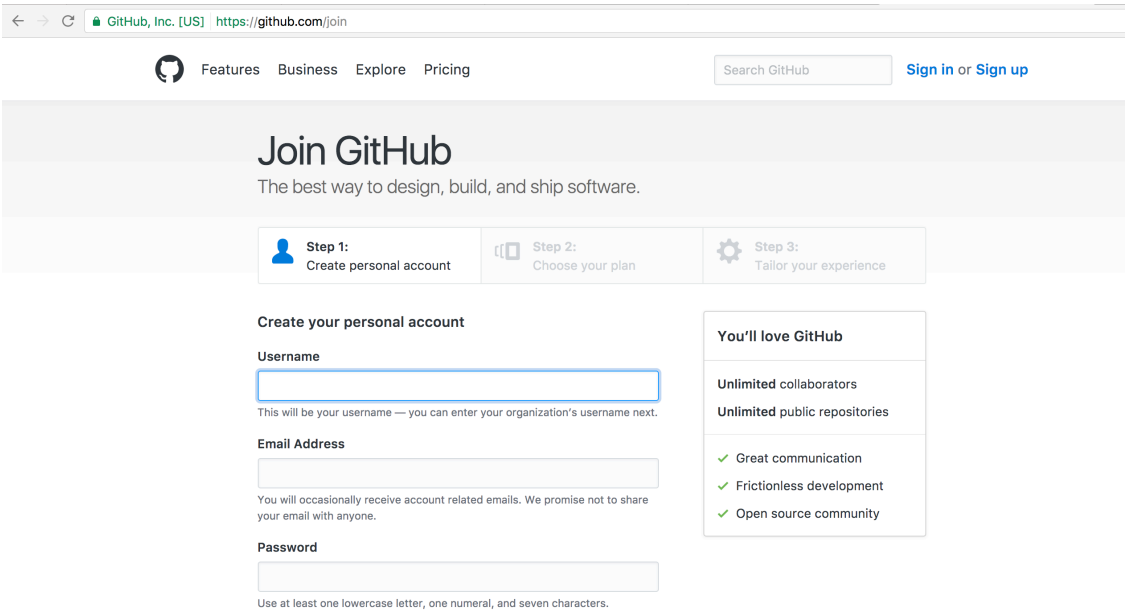
2. _ If you don't have an **IBM ID** yet, please fill in the form and click on "Create account", it will create an **IBM ID** and a Bluemix account. By default the Bluemix account will provide you free 30-day trial access with no credit card required. You get access to:

- 2 GB of runtime and container memory to run your applications,
- unlimited IBM services and APIs (max 10 services instances)
- and complimentary support limited to Severity 4

7.2 Create your GitHub account

Please follow the procedure : https://github.com/join

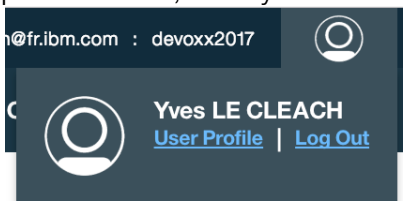
Copyright IBM Corporation 2017. All rights reserved.



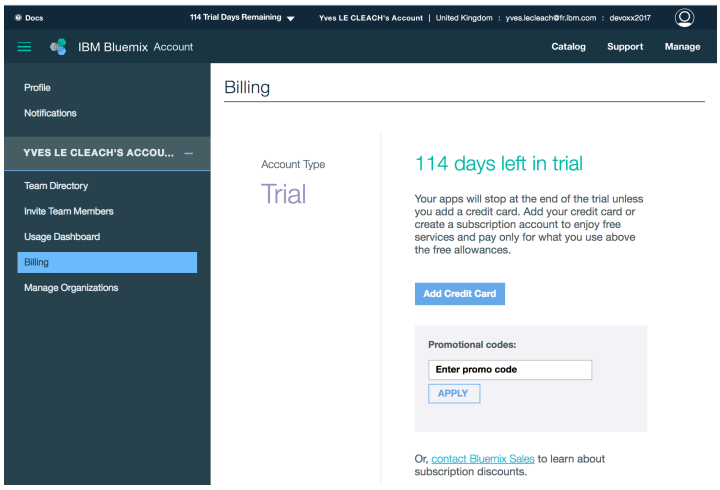
To install the Git Command Line, please follow :
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

7.3 Applying a Promo Code

If you received from the instructor a promo-code, then you have to click on the User Profile link



Then just enter the promo-code.



Note: each code is personal and usable once, and the account extension is variable. (e.g. could be +1 / +2 / +3 months extension of the Free Trial. (For this Devoxx France 2017 event, those promo-codes are +3 months extension of a one month trial, limited to 50 promo-codes).

Copyright IBM Corporation 2017. All rights reserved.