HandsOn Lab
Watson Node-RED on Bluemix

Version: 3
Last modification date: 19th October 2016
Owner: Yves LE CLEACH

Twitter : @ylecleach #IBMBizco @IBMFranceLab
Mail : yves.lecleach@fr.ibm.com

Table of Contents

This Lab is designed so it can be done entirely using a Bluemix Trial account. (1 app 512 Mb +7 services instances)

Yves, Watson Node-RED OSS team.

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.

More recently (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
- Discover some of the Watson API through the Watson nodes on Bluemix
- Build your first Cognitive Node-RED applications using Watson nodes

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

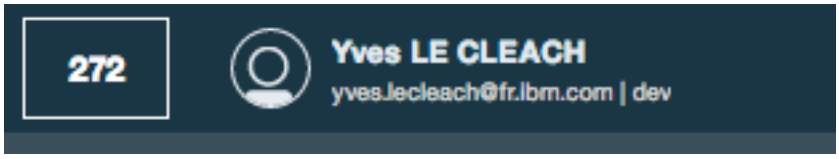
2 Discovering Bluemix and Node-RED

We assume in this Labs that you are using the New Bluemix Console : https://new-console.ng.bluemix.net

2.1 Create your first Node-RED application on Bluemix

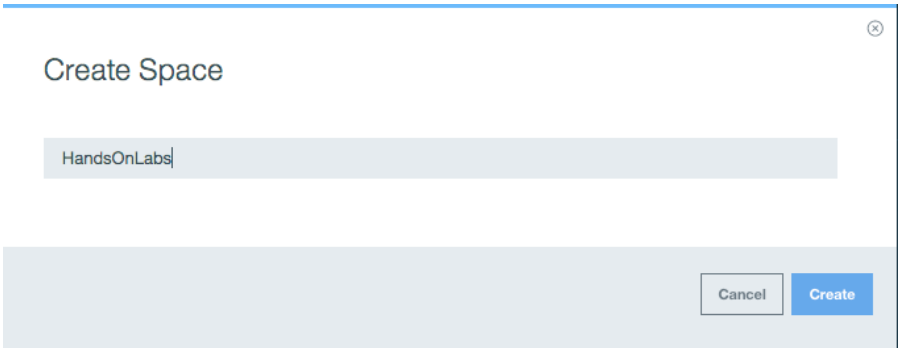
First, log in onto your Bluemix account : https://bluemix.net and creates a Space '**HandsOnLabs'** in which you will work for this lab session.

Click on your Name

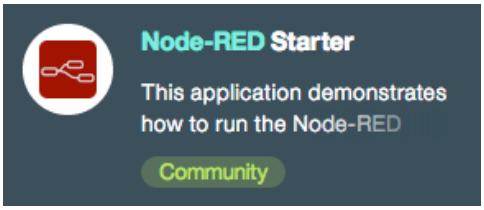


Click on the Create a Space Link

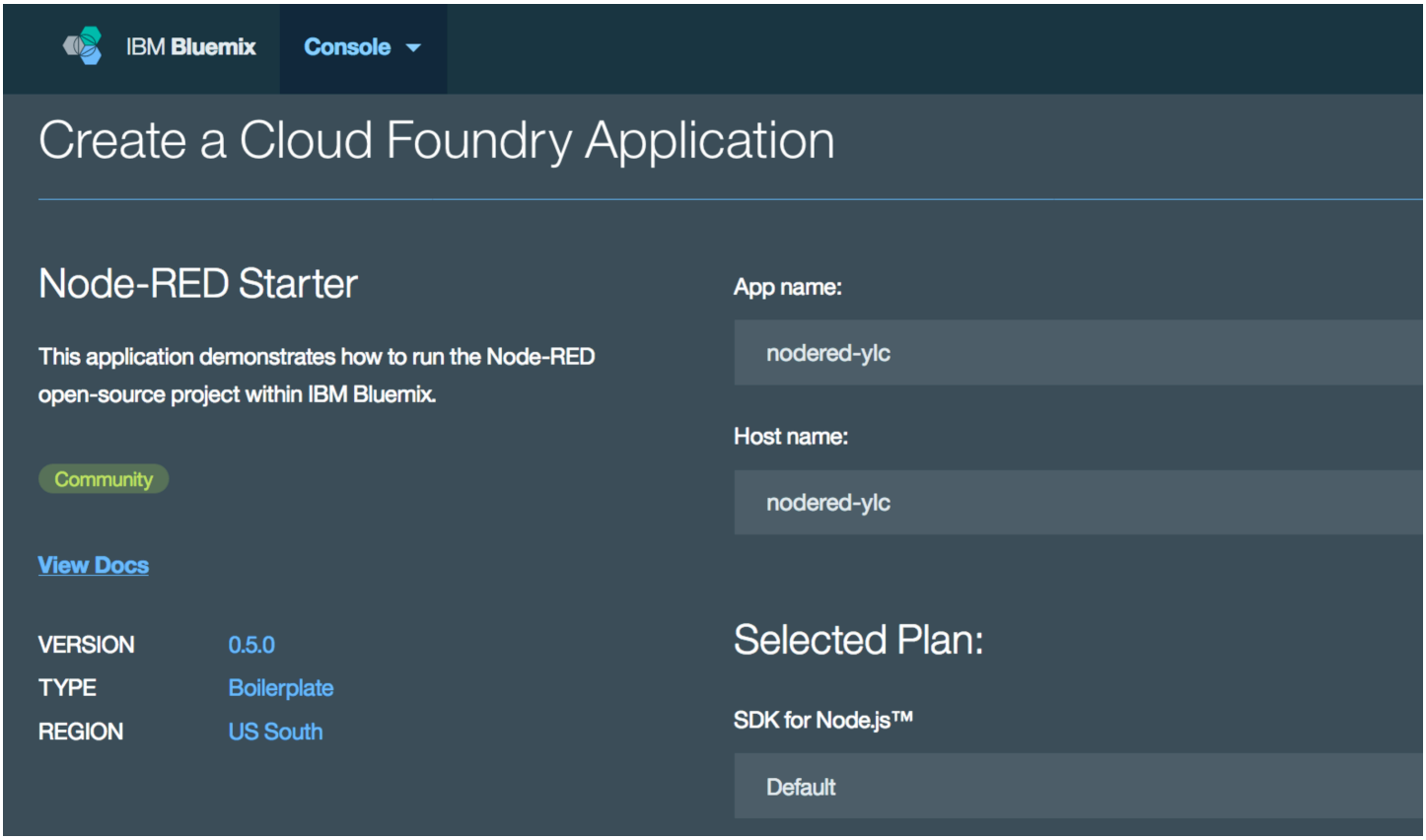
Enter "HandsOnLab" and click on the Create button :



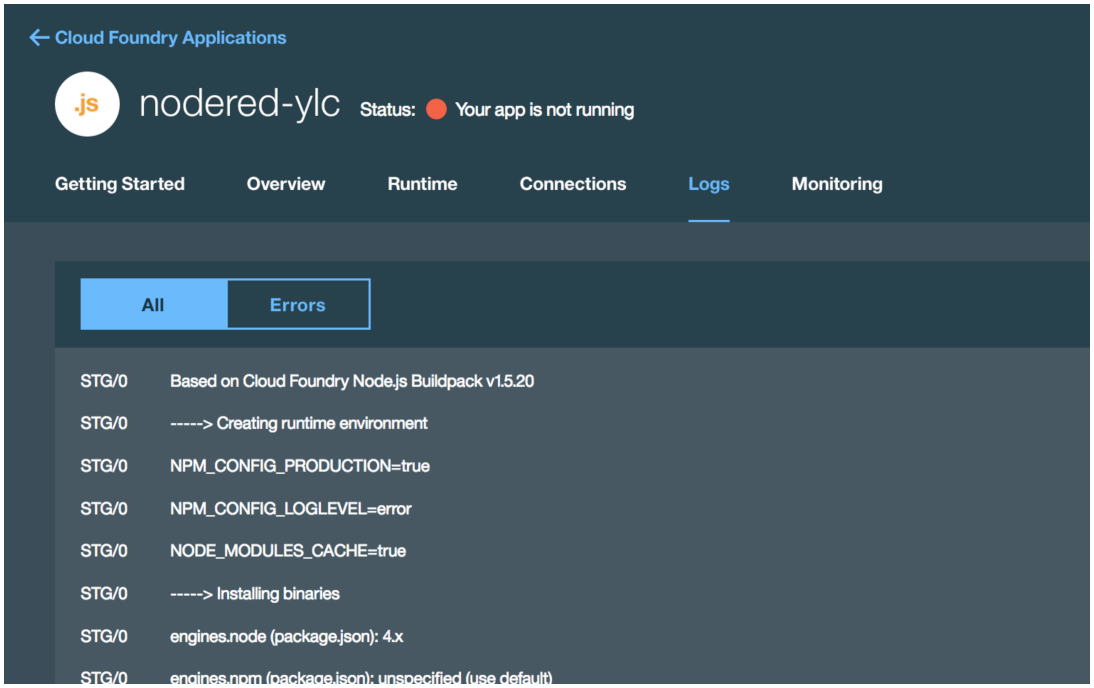
Then, go to the **Catalog** and type “node-red” in the Search field.
Then Click on the Node-RED Starter in the Boilerplates category:



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

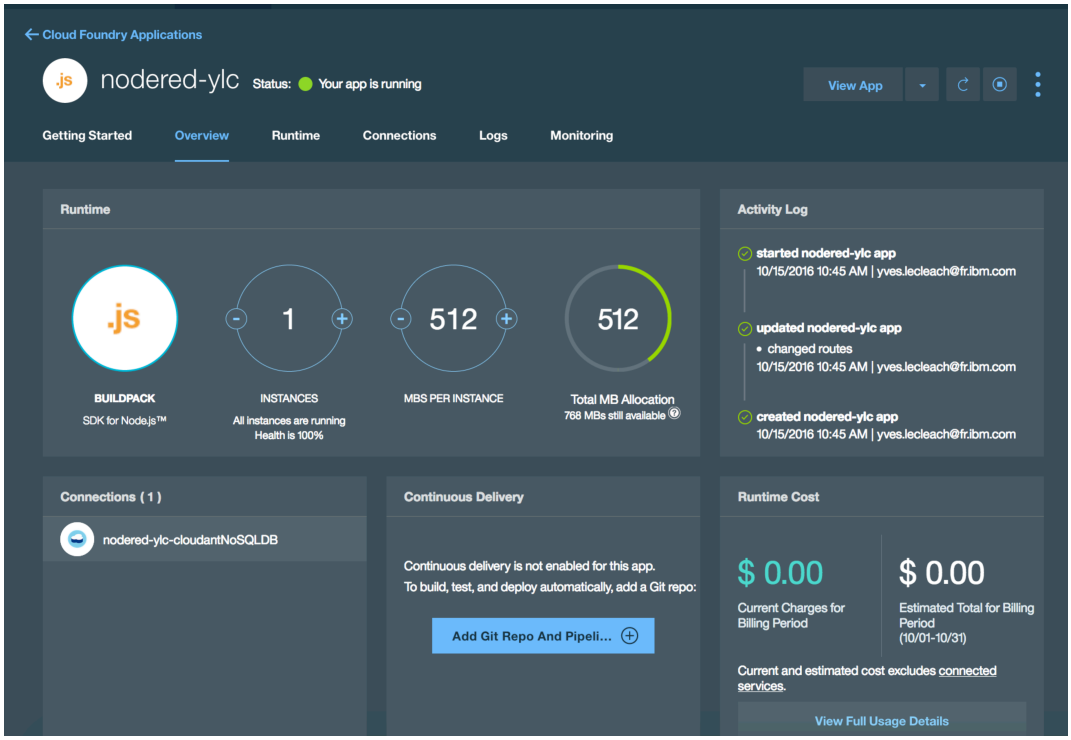


The provisioning of your new node-RED instance is starting. You can go immediately in the Logs Tab of your app to see the details of the provisioning sequence behing the scene. (this also helps one to understand the advantage to use a PaaS technology such as Cloud Foundry).



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

Click now in the Overview tab of your application.

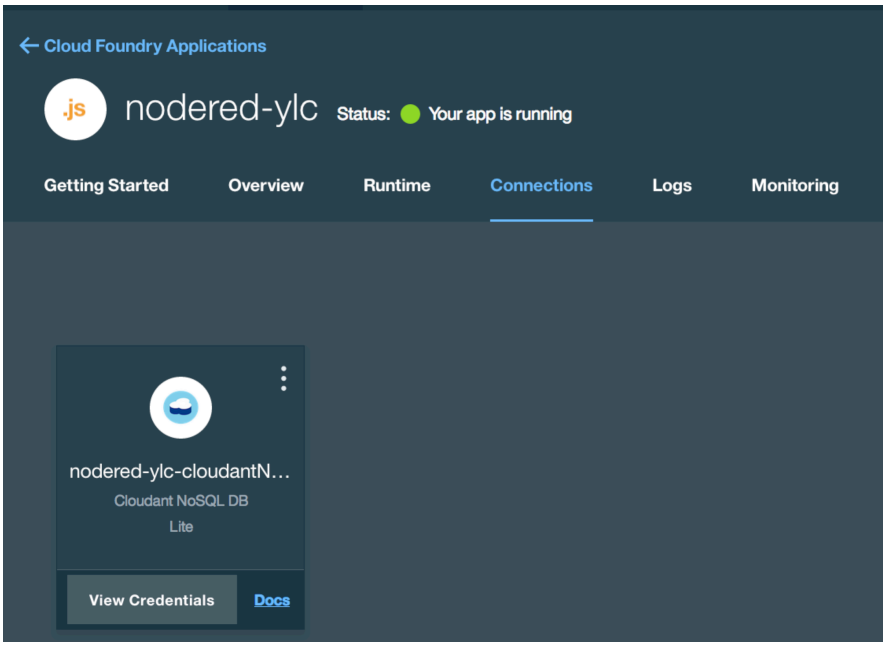


In a Node-RED app, which is run by a Node.js server instance, you can define any numbers of functionality you want, as on any application servers. So a Node-RED app, can host any numbers of applications you desire, as it can be done on a Java EE application server. (e.g: Liberty Java server).

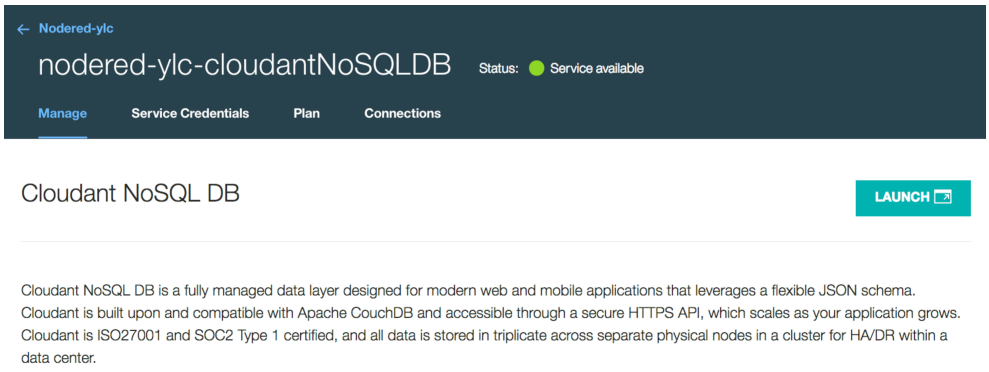
Notice : Node-js is designed and fits well for Non Blocking I/O applications, and many others usages, and is more and more used worldwide.

Connections Tab

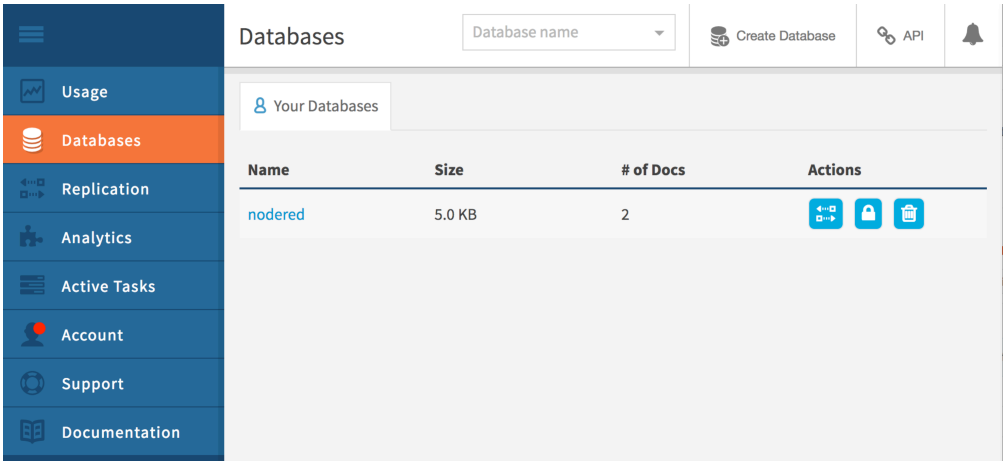
In this tab you will see all Bluemix services instances which are Link (or bound) to your app. One given service instance can be linked to any number of applications inside the same Space. A Cloud Foundry Organization contains typically multiple space to organize apps and service instances. (ex of space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



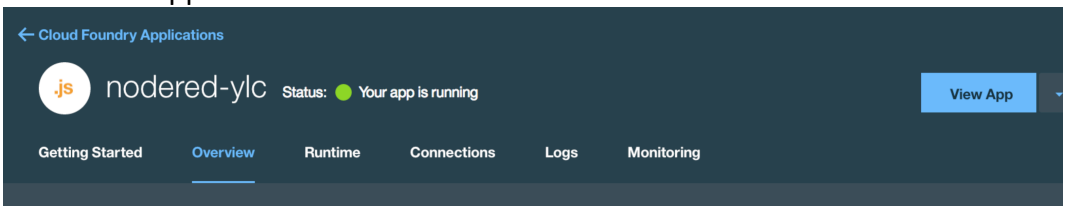
You can observe that the Node-RED app is bound to a Cloudant service instance which has been provisioning at creation time of the Node-RED app.
You can spend 2 min to click on the cloudant service instance and click on the Launch Dashboard.



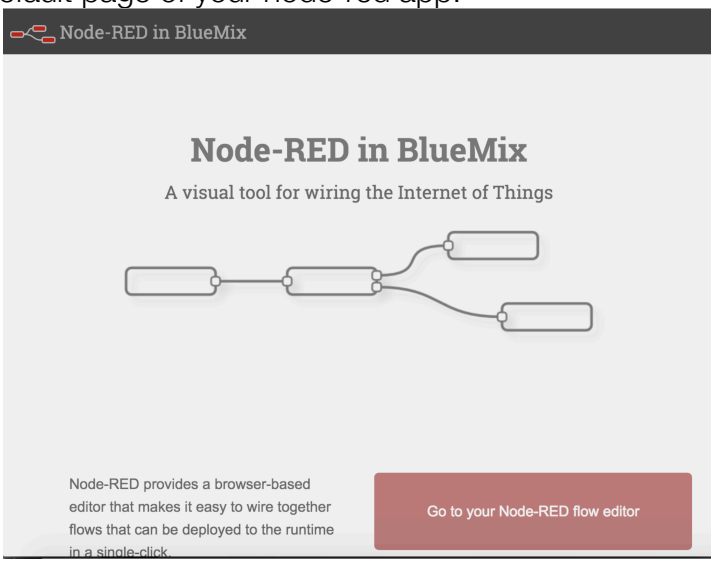
In the Database tab of the Cloudant dashboard, you will find the “nodered” database used by your Node-RED app to save all Node Flows and configuration you will produce. (all is stored as Json documents). You do not need to update directly this “nodered” database as it is used by the Node-RED framework.



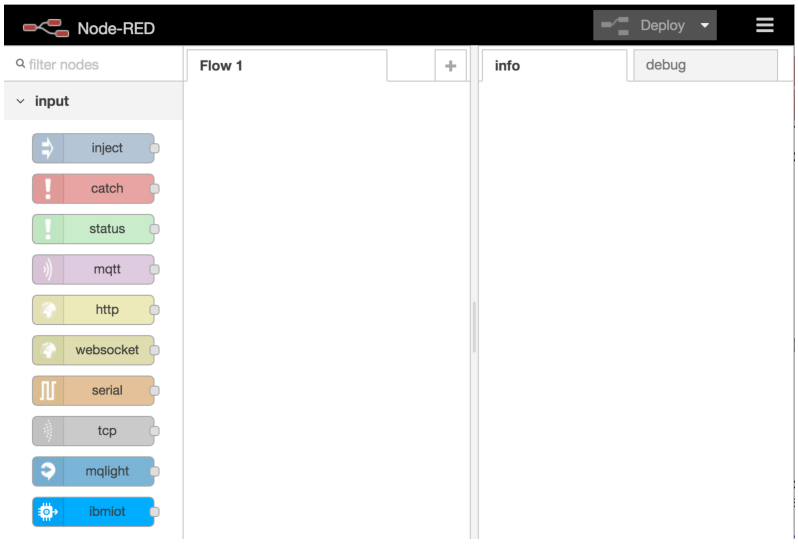
Now, Click on the "**View App**" button



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:



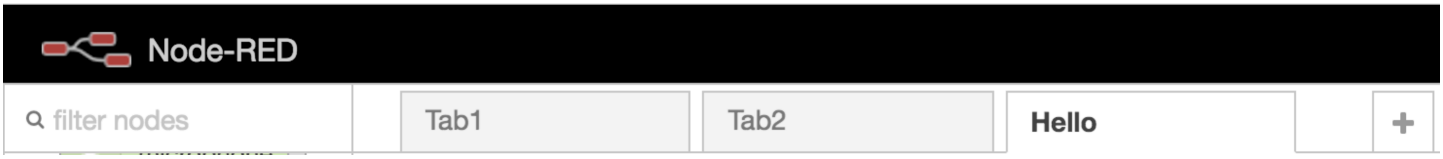
Quickly Browse the Node Palette, and Look at the Info Tab to have some more information on the selected node.

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 Tab for your Flow using the + icon:

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



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

(see OK Watson to see an example of usage)

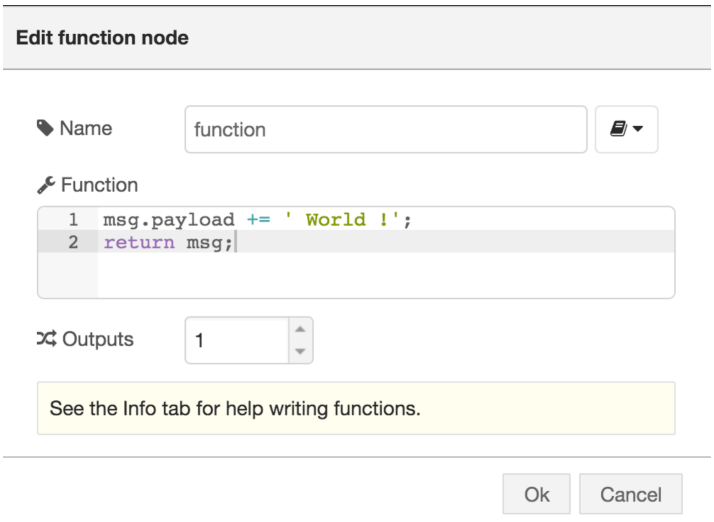
Now, select the Hello Tab, and lets create our first flow.

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 second 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:

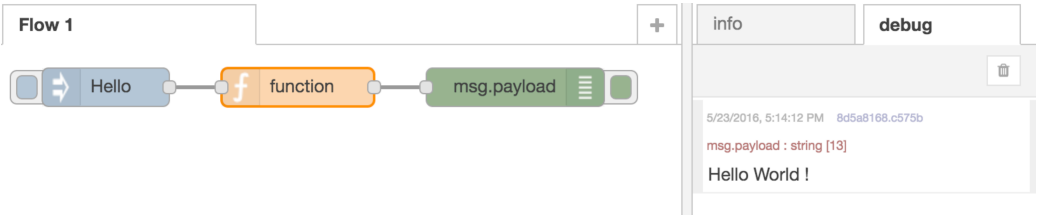
   ```
   msg.payload += ' World !';
   ```

   This will add 'World !' to the payload. The complete function should look like this:



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 window on the right (click on 'Debug' to change the view from info to debug).



2.3 Importing or Exporting Flows

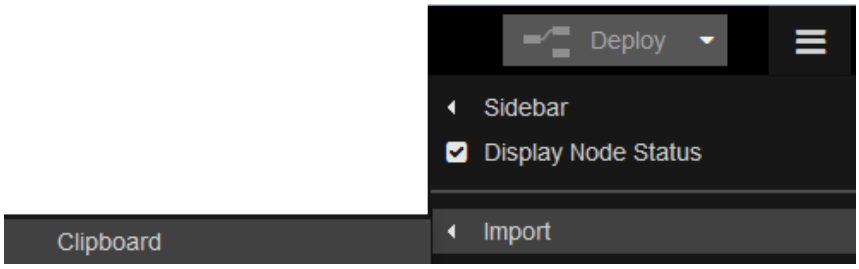
With Node-RED you can export/import easily the flows you have designed or ones from the node-red community. (http://flows.nodered.org/)

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

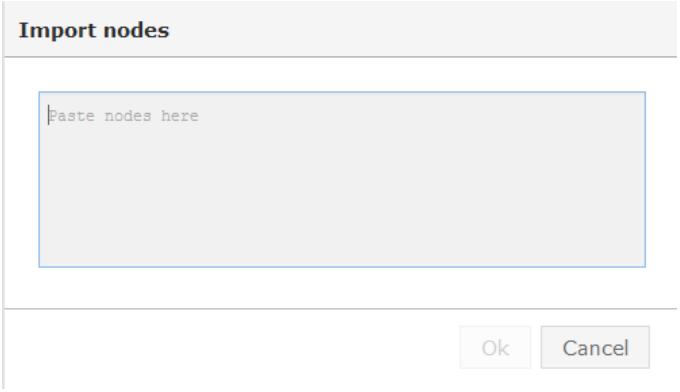
https://raw.githubusercontent.com/ylecleach/labs/master/hello-flow.json

Select the flow and copy it to the clipboard (Ctrl-C). Import the flow into Node-RED using by selecting the node-RED menu

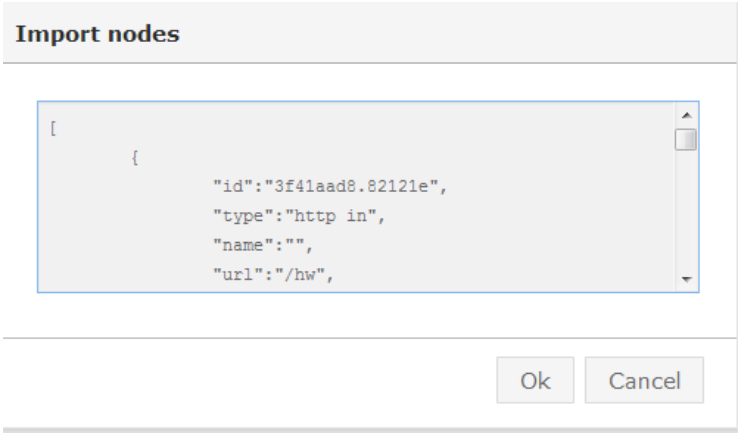
and select the import from clipboard option.



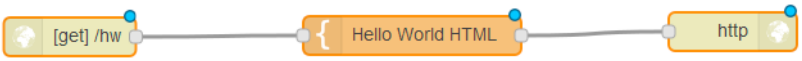
You will be presented with a form in which you create nodes by entering json data.



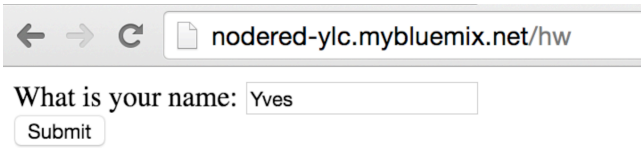
Import your copied flow by pasting (Ctrl-P) from the clipboard into the form.

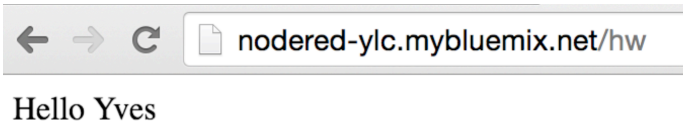


Place the imported flow onto your node-RED page and press DEPLOY.



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

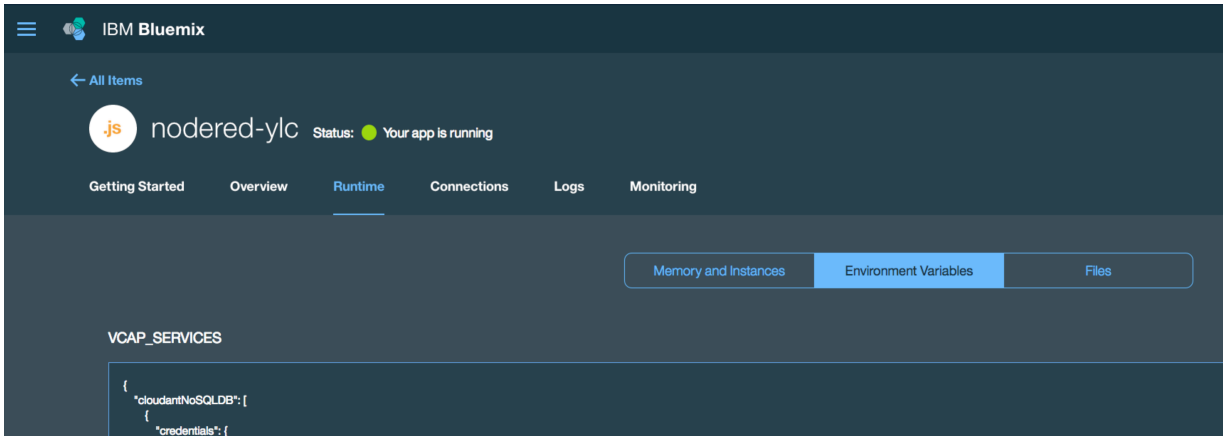




2.4 Securing your Node-RED editor (recommended)

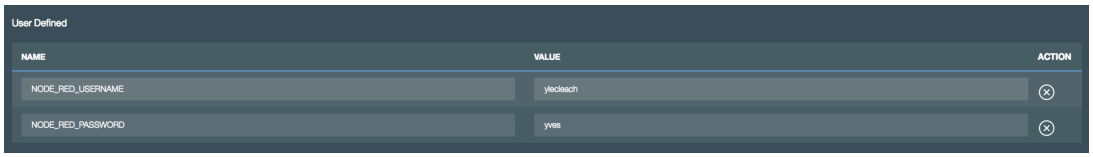
By default, your Node-RED editor will be public, let's configure the access.

First enter your app details, and select Runtime, then Environment Variables tab.



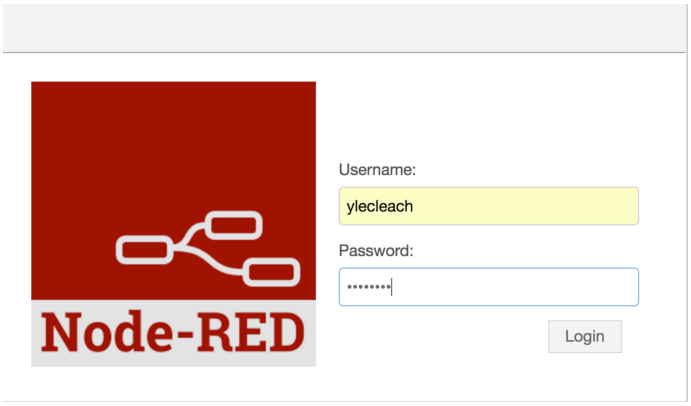
Under the section **"User-Defined"** at the bottom of this page, add the following those 2 variables with their corresponding values.

- 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. Your Node-RED app will restage quickly.

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



2.5 Enable Continuous Delivery to your app

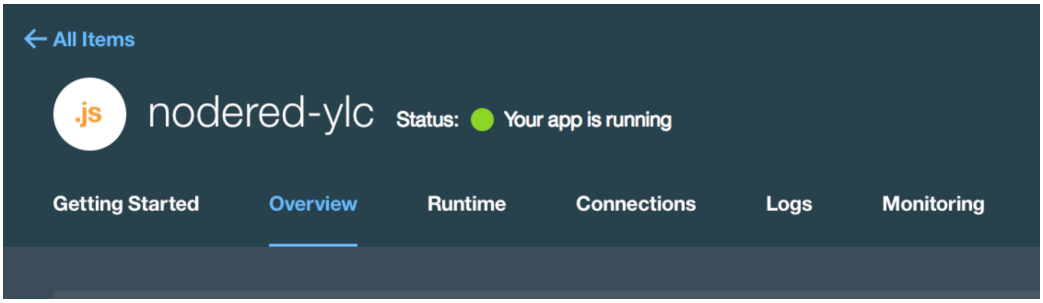
Before you start and to make use of the starter kit applications in section 4, you will need some extra node-RED nodes.

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

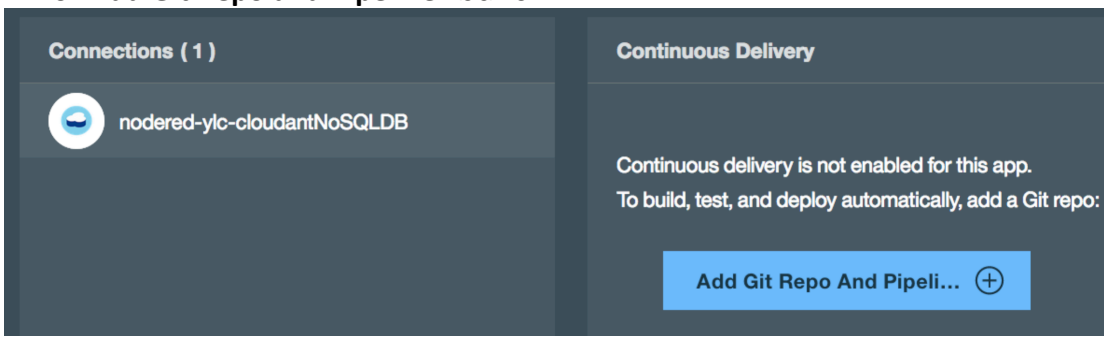
To do so, you have to Enable the **Continous Delivery** on your app that will :

- create a Git repository for the Node-RED app code (the Project)
- create a default deployment pipeline

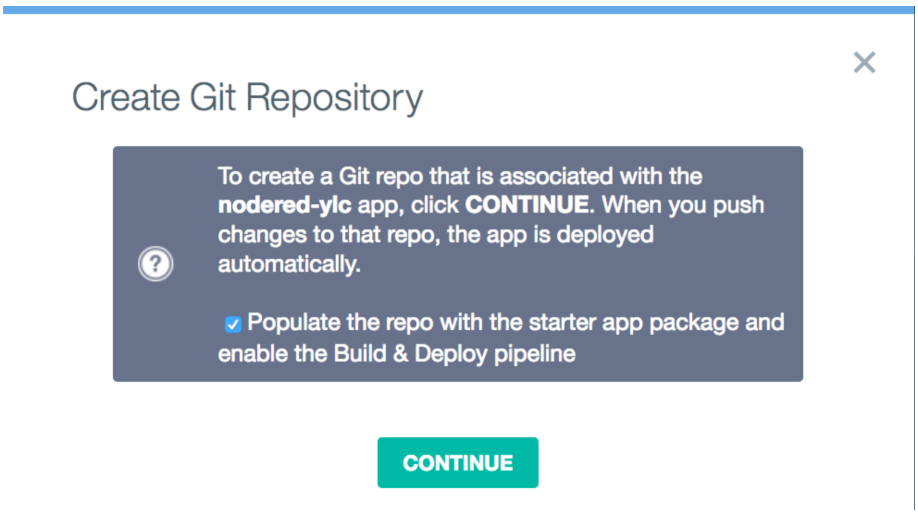
Click on the Overview tab of your application.



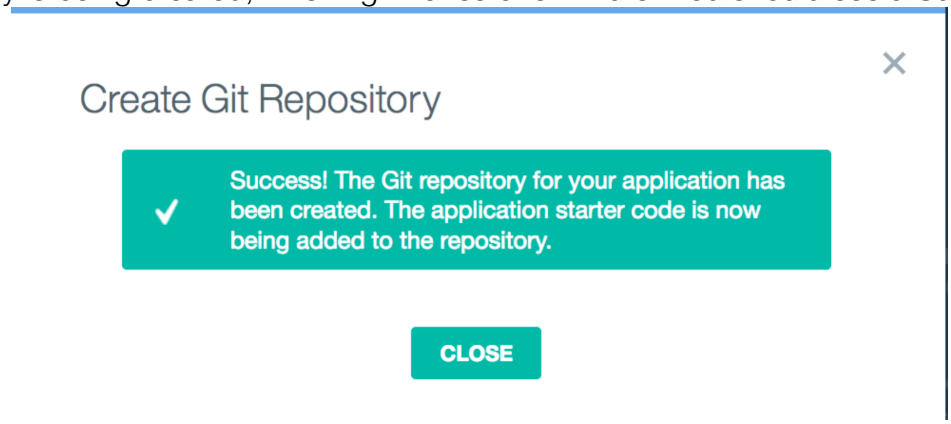
Then click on the "**Add Git Repo and Pipeline**" button:



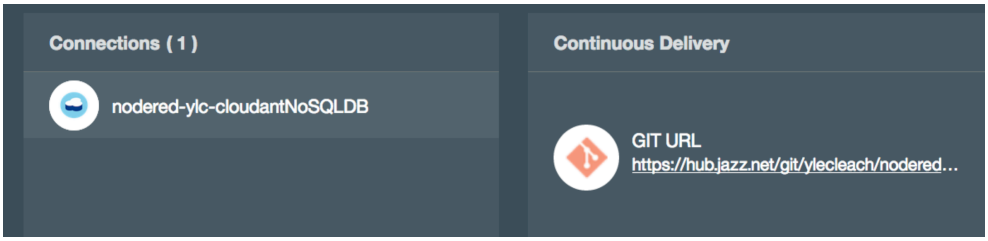
Click on Continue, and leave the Populate checkbox as is. (checked by default)



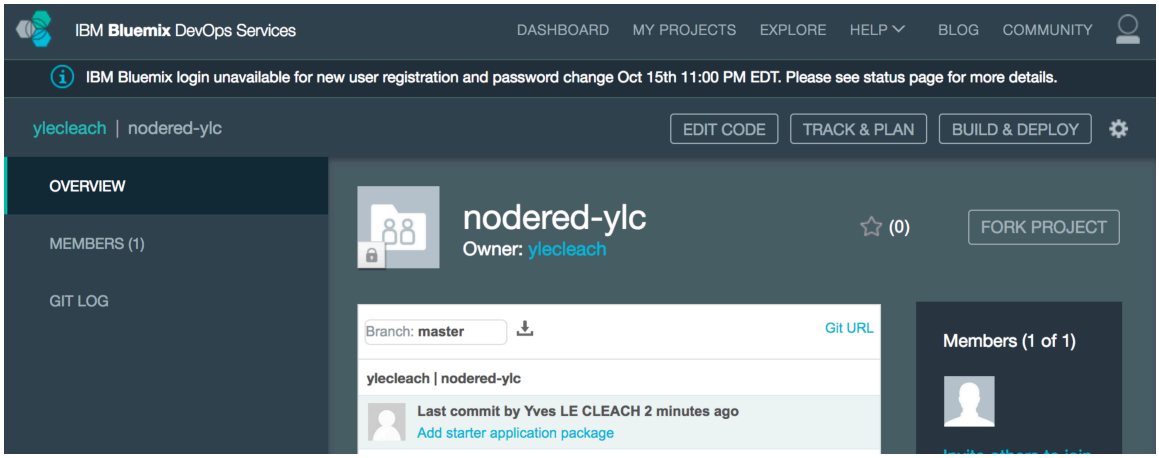
Your Git repository is being created, This might takes one minute. You should see a Success message:



Click on the Close button. Now a new link is available so you can update your application, commit eventually your change, redeploy manually or automatically your app.

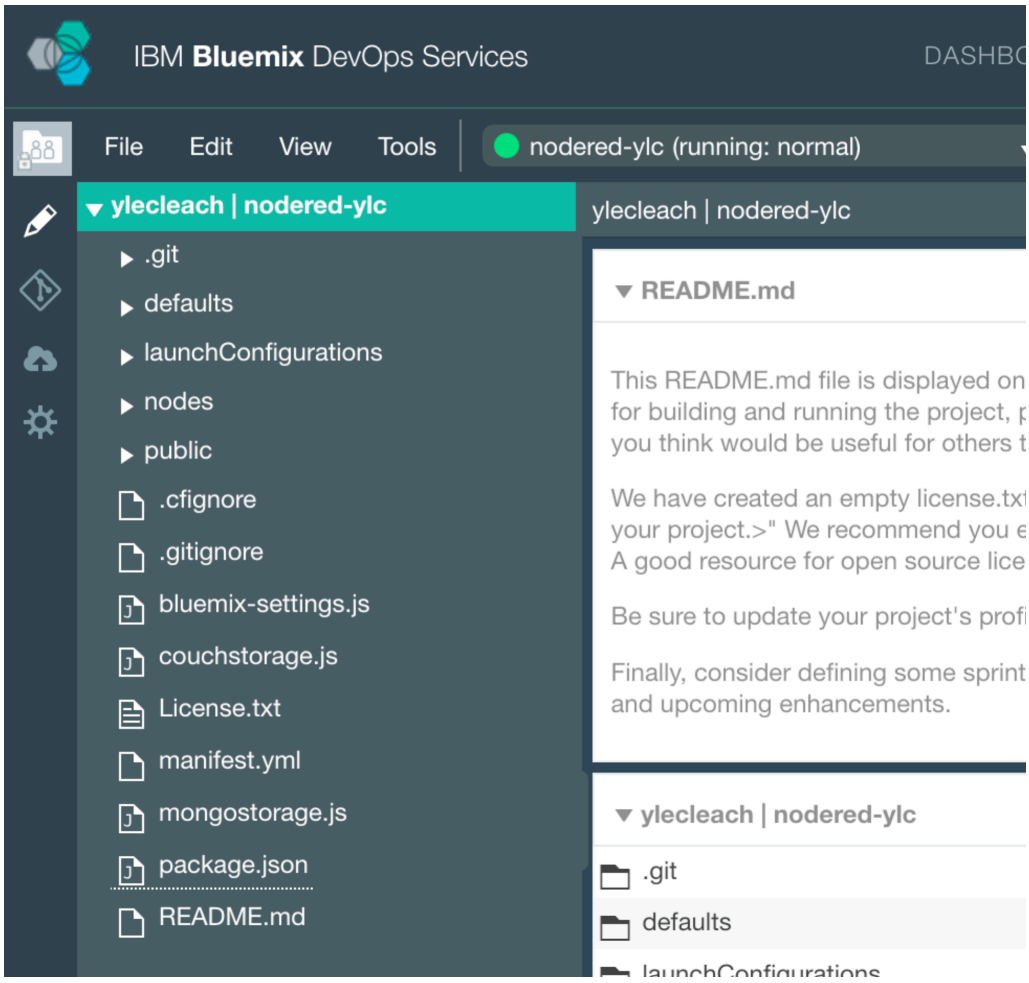


Click on that GIT URL link to do the following change.



Click on EDIT CODE button.

Click on the package.json file



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",
"node-red-node-box": "0.x"
```

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

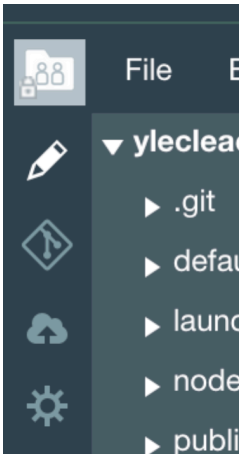
package.json

```
{
    "name"         : "node-red-bluemix",
    "version"      : "0.5.0",
    "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"
    },
    "scripts": {
        "start": "node --max-old-space-size=384 node_modules/node-red/red.js --settings ./bluemix-
    },
    "engines": {
        "node": "4.x"
    }
}
```

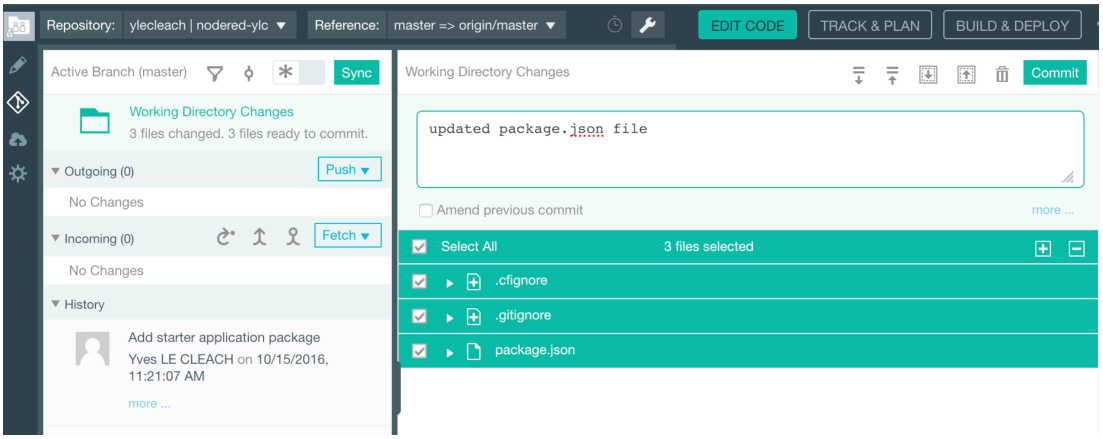
Lets commit this change so the app will be automatically redeployed by the default pipeline.

First, Save your change. (Control + S)

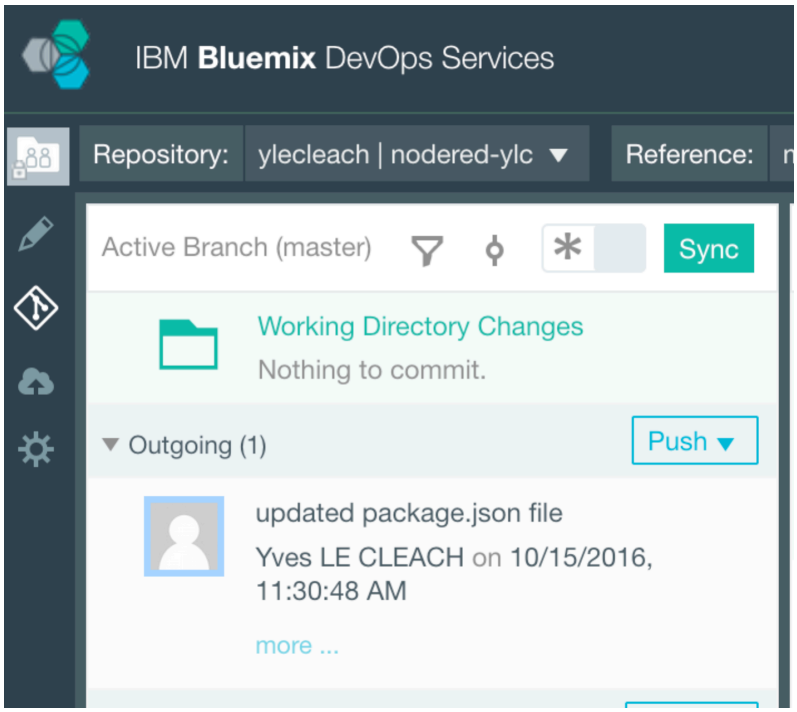
Then click on Git Icon  in the Left bar (icon just below the “Pencil” Icon used for Edit):



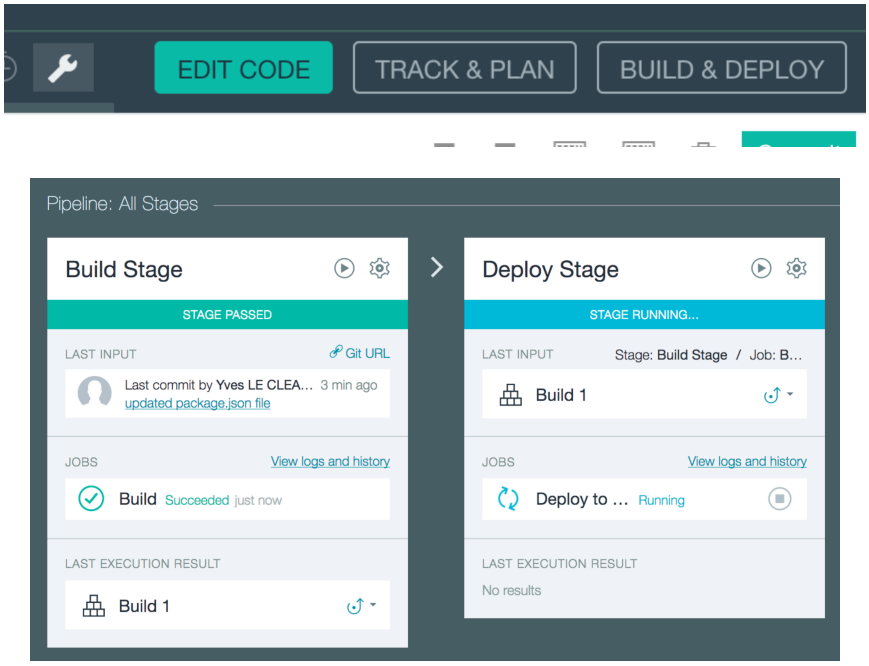
Enter a commit message, and click on the **Commit button**.



Then Click on the **Push button**

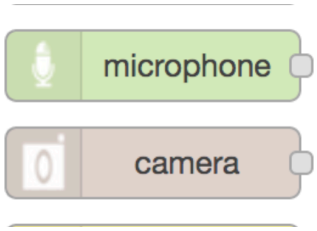


You can click now on the **BUILD @ DEPLOY** button to see the Pipeline in action :



You can follow the restaging sequence by clicking on the “View logs and history link”.

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

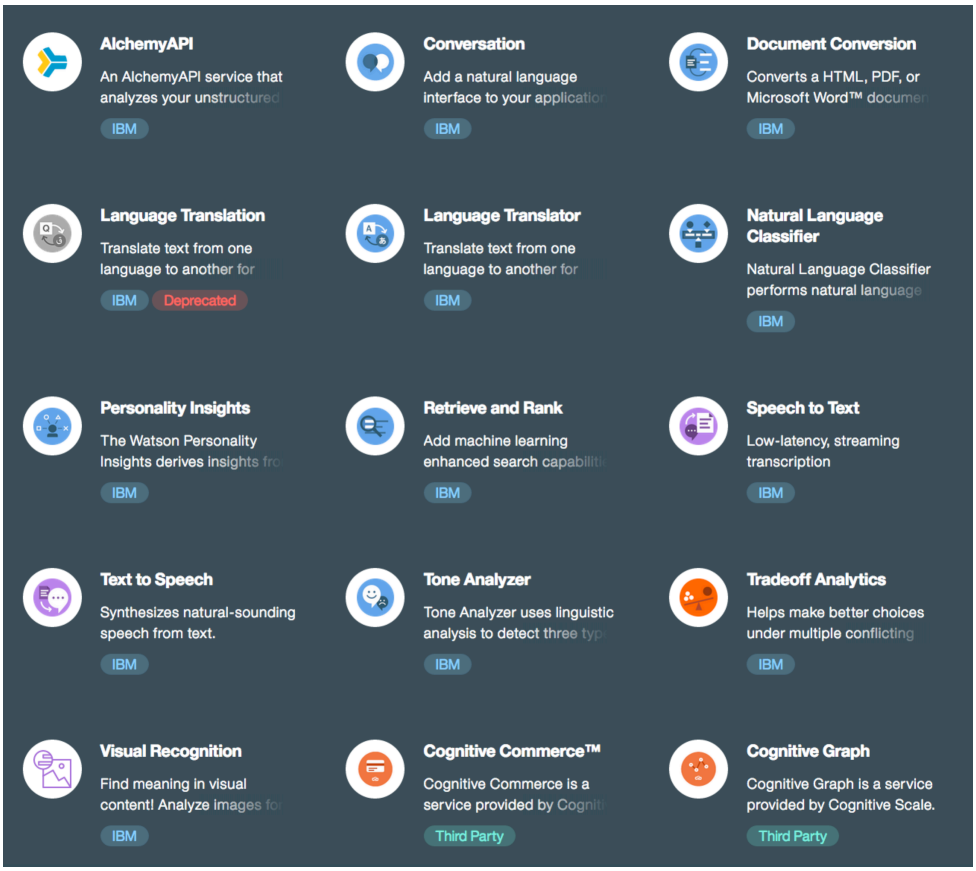
- **Basic Labs** (section 3): if you never used Watson APIs or Watson nodes, or
- **Node Red Watson Starter Kits** (section 4): if you are already familiar with Watson APIs or nodes.

3 BASICS Labs

The basics labs 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 done prior using the API. Some others as Watson Language Translator service offers pre-defined domain specifics domains.



3.1 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 this Lab, to try the Training of this service with a customize domain.

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

Tips:

- You can use the Dropbox or Box nodes as described in their respective documentations :
  Dropbox setup :
  https://github.com/watson-developer-cloud/node-red-labs/tree/master/utilities/dropbox_setup
  Box setup:
  https://github.com/watson-developer-cloud/node-red-labs/tree/master/utilities/box_setup

Watson documentation :
http://www.ibm.com/watson/developercloud/doc/language-translator/

Watson API documentation :
http://www.ibm.com/watson/developercloud/language-translator/api/v2/

3.2 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 ADD A SERVICE OR API. A restage of the app is required.

Watson documentation :
https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/tone-analyzer/

Watson API documentation :
https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/tone-analyzer/api/v3/

3.3 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 ADD A SERVICE OR API. A restage of the app is required.

Watson documentation :
https://www.ibm.com/watson/developercloud/doc/text-to-speech/

Watson API documentation :
https://www.ibm.com/watson/developercloud/text-to-speech/api/v1/

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 at 3 hackathons and will be used at World of Watson and the Watson Developer Conferences. 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 (eg. Speech to Text, Translation, Natural Language Classifier, Personality Insights, Alchemy Data Analysis, Text to Speech).

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: (as of 18th October 2016)

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 below.

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

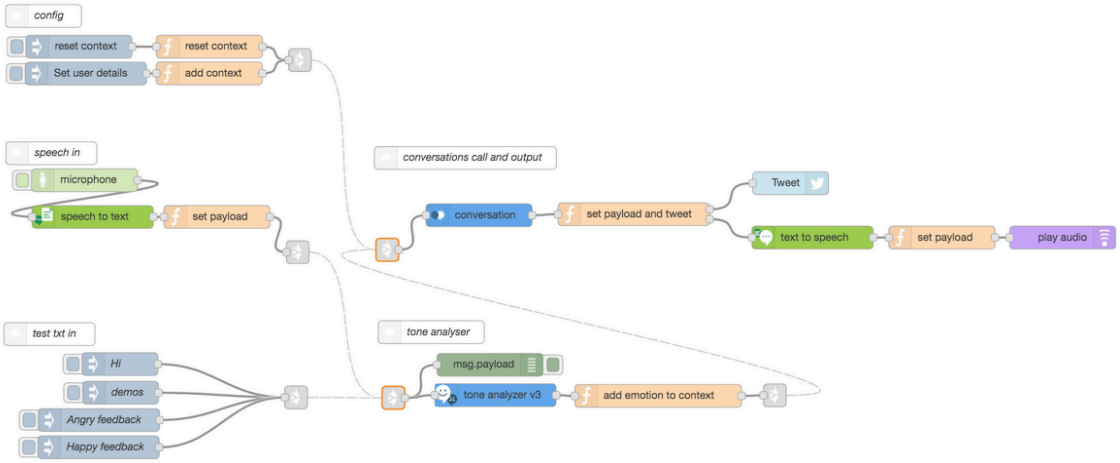
Tips : to speed up, Connect the following Watson services, but choose the Restage only when the last service has been Connected to your Node-RED app :

- **Speech to Text**
- **Text to Speech**
- **Conversation**
- **Tone Analyser**

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.

Important : when using Bluemix, you do not need to configure your service credentials on the corresponding nodes.

Lab instructions:
https://github.com/watson-developer-cloud/node-red-labs/blob/master/starter-kits/ok_watson/README.md



4.2 Selfie Training

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

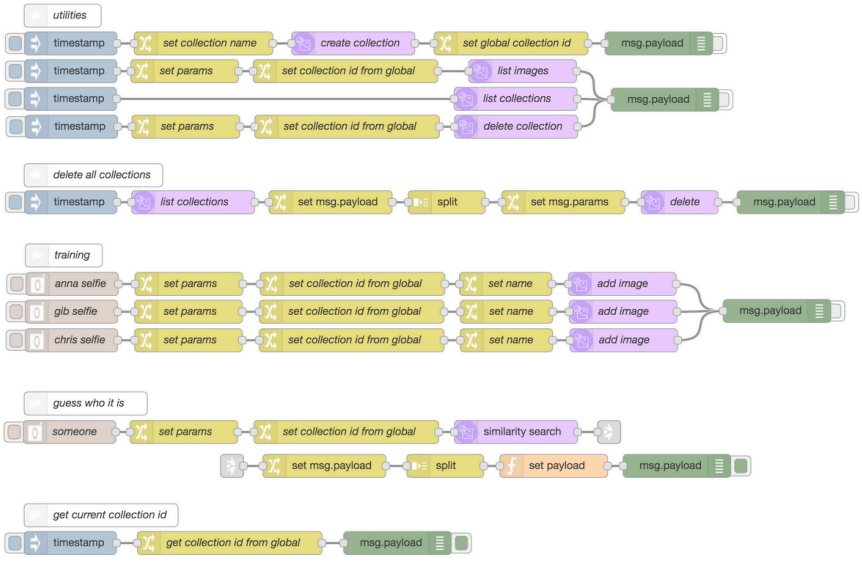
Tips : to speed up, Connect the following Watson services, but choose the Restage only when the last service has been added :

- **Visual Recognition**

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

Important : when using Bluemix, you do not need to configure your service credentials on the corresponding nodes.

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



Additional Resources

Watson on Node-RED support :
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 support :
- Introduction to Node-RED: https://ibm.biz/BdHvfc
- Node Red Forum on GoogleGroups

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 more, Want to share your Bluemix or Watson Projects experiences in France

Bluemix/Watson communities in France :
- Bluemix Paris Meetup : http://ibm.biz/BluemixParisMeetup
- Bluemix Usergroup France blog : http://ibm.biz/MyBluemixBlog

Follow us me :

- Twitter : https://twitter.com/ylecleach
- GitHub : https://github.com/ylecleach

Follow IBM France Lab :
- Twitter : https://twitter.com/IBMFranceLab
- Slideshare : http://fr.slideshare.net/IBMFranceLab

5 Annexes

5.1 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** 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 "Get started free"

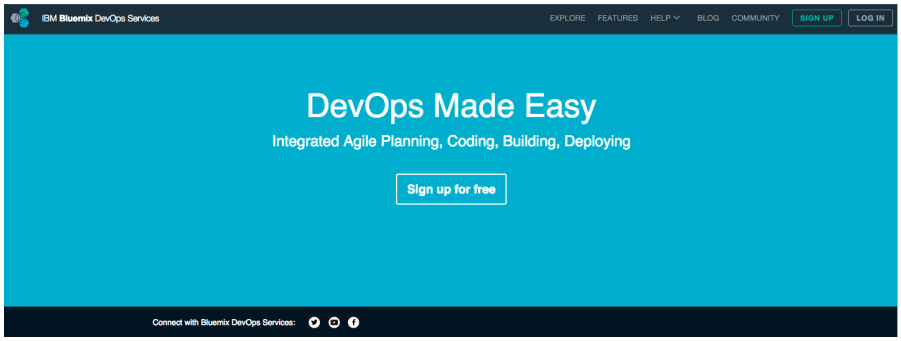
2. _ If you don't have an **IBM ID** yet, please fill in the form and click on "Create account", it will create an **IBM ID** and a Bluemix account. By default the Bluemix account will provide you free 30-day trial access with no credit card required. You get access to:

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

5.2 Create your JazzHub account

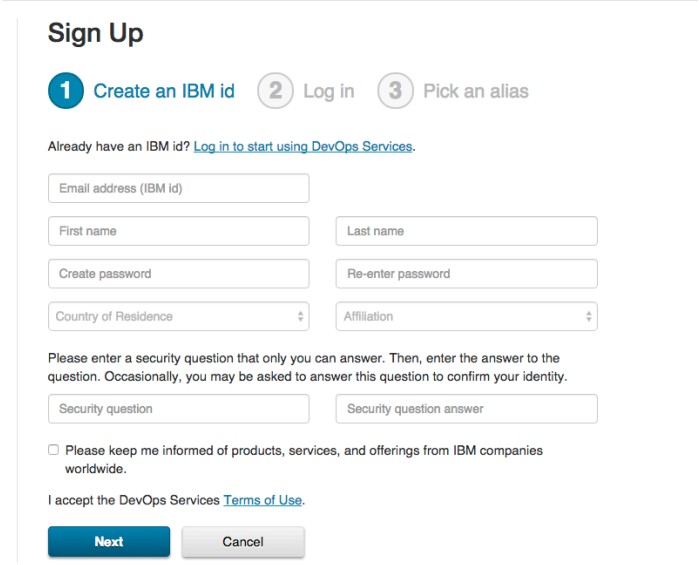
Please follow the 3 steps below.

1. Browse https://hub.jazz.net/ and click on SIGN UP (up and right).



IBM DevOps approach is a set of practices, tools and services for accelerating the development and release cycle of software applications. It helps enterprises get their ideas into production fast.

2.Then, click on “Log in to start using DevOps Services”



3. _ Log In and chose an alias associated to your **IBM ID**. The alias is unique and it will be used as a short name in Git repository paths. Click on “Continue”. You successfully registered on DevOps Services.