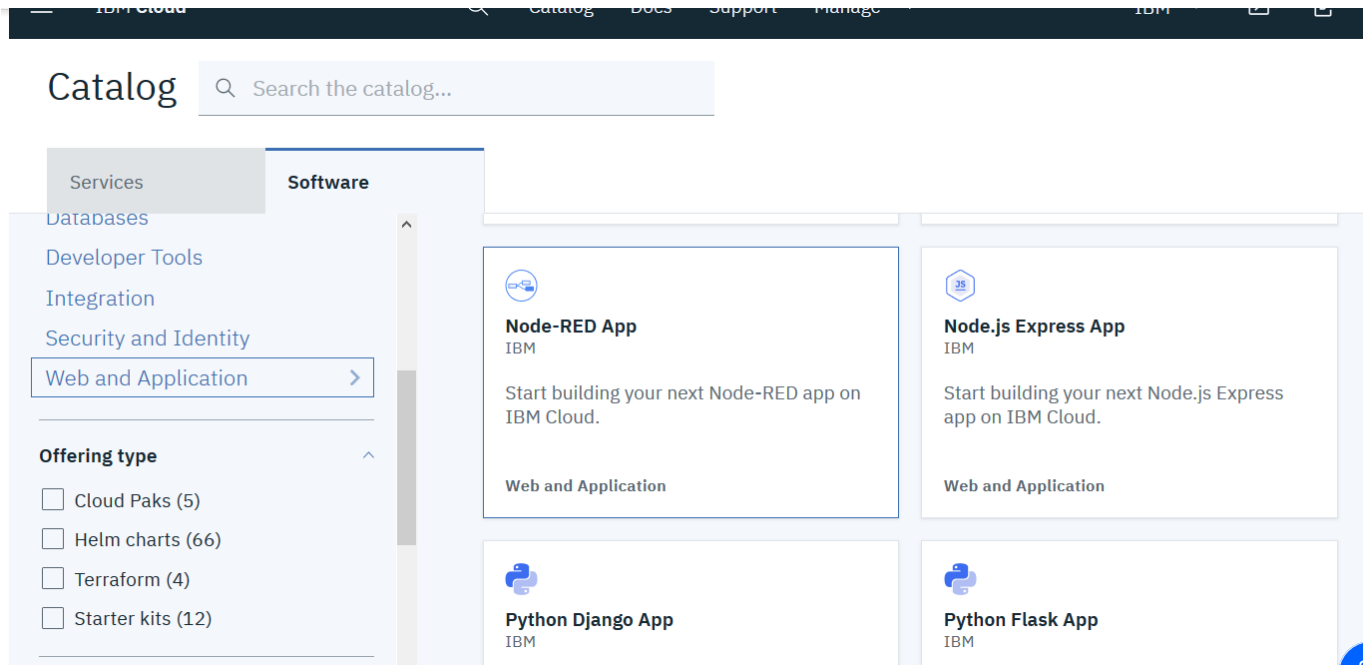


IBM Node-RED IBM Cloud template 2020.

Розгортання додатку в IBM Cloud

Вибрати в каталозі хмарних сервісів Node-Red



кліком мишки стартуємо створення додатку. Після цього відкриється вікно, де знадобиться вибрати регіон. Можна вибрати любий, де додаток є, але мені особисто подобається London. Регіон буде запитуватися кілька разів і бажано вибрати одн і той же регіон для всіх компонентів. Тому:

- на наступному екрані вводимо назву нашоно додатку та вибираємо бажаний регіон (London)

Catalog / Starter kit details



Node-RED

Code Pattern • Lite

Create app

create the application.

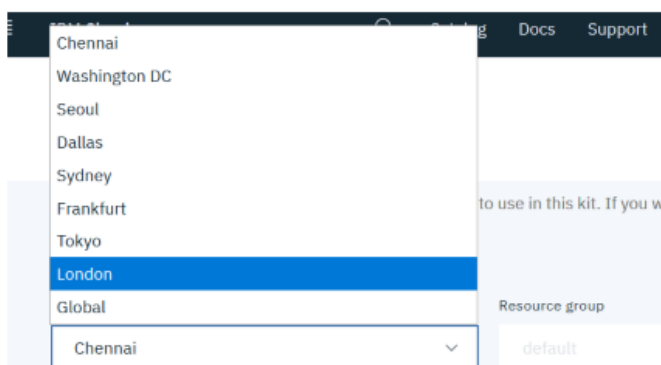
App details

App name

Accept the default name, or enter a value up to 128 characters.

nod-red-wshp

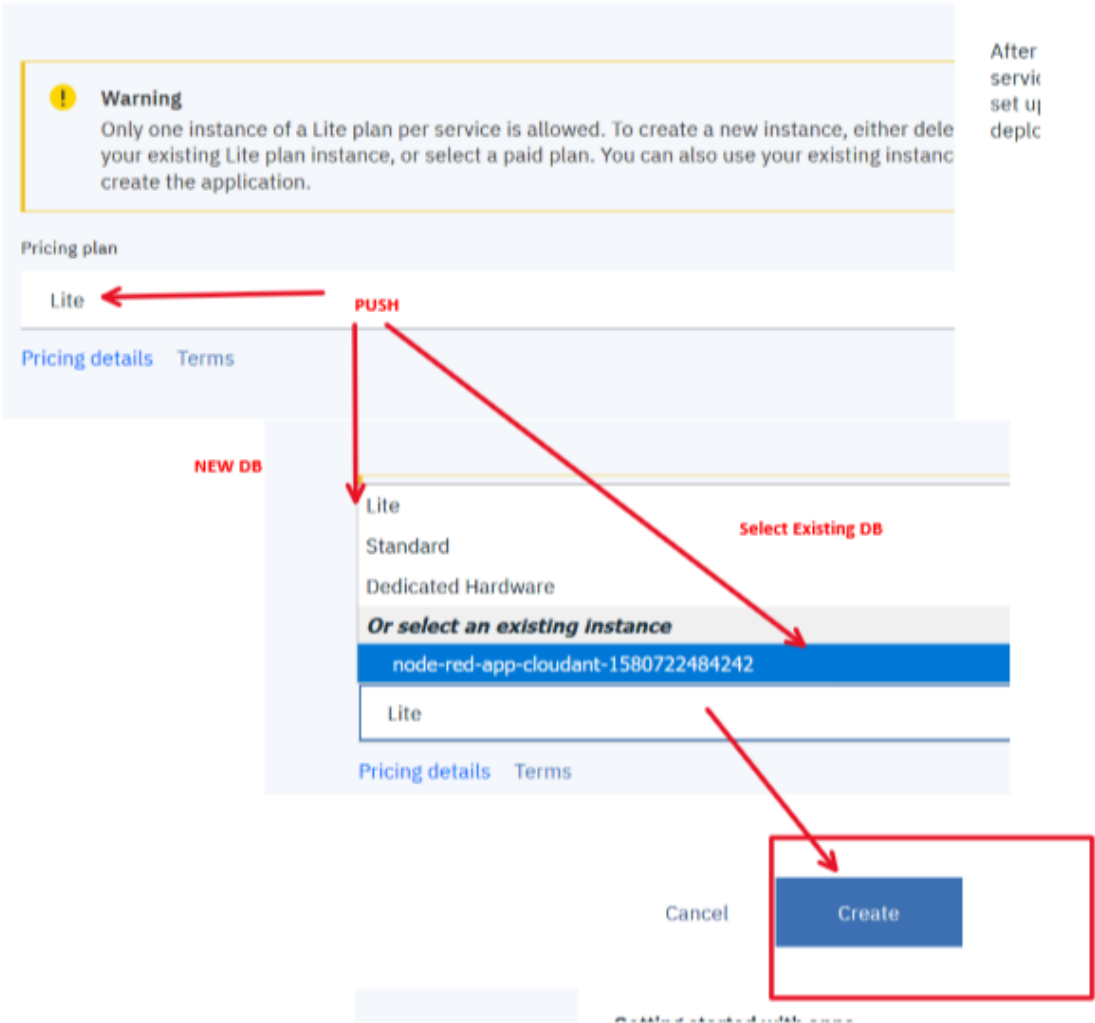
Resource group



і натискаємо "Create app".

- З'явиться екран на створення бази даних Cloudant.

Якщо БД ще не створювалась, то система автоматично запропонує створити БД з тарифним планом Lite. Але, якщо така БД уже була створена, потрібно вибрати її у випадаючому списку, або створити нову, але уже платно.



- Настройка екрану, що створює toolchain

IBM Cloud

Resource list / App details

nod-red-wshp Add tags

Resource group: default Download code

App details Developer resources

Name	Resources	Actions
Cloudant	Documentation	

Configure Continuous Delivery

Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more.

Deploy your app

Knowledge Guide

Next steps

To enable Continuous Delivery as deploy to IBM Cloud:

1. Click **Deploy your app**.
2. Select a deployment target, set the toolchain settings, and click **Create**. The deployment starts automatically.
3. Check the deployment status in the **Delivery Pipeline** section.

Download your app for local development:

1. Install the IBM Cloud CLI. [Learn more](#).

Run locally

After you download your app by using the `ibmcloud dev code` command, you can quickly have it up and running in a local development environment.

1. Install the [IBM Cloud Developer Tools](#).
2. Run it in a local development container from the app directory. This step requires [Docker](#).

```
$ ibmcloud dev build
$ ibmcloud dev run
3. Deploy
$ ibmcloud dev deploy
```



From <<https://cloud.ibm.com/developer/appservice/apps/153348d7-9e97-4ce8-b69b-ba459e473ae0>>

Тут можна загрузити вихідний код на робочу станцію, та, таки створити toolchain. На скріншоті показаний діалог створення toolchain. Не забуваємо про один і той же дата центр!


Resource list / App details

nod-red-wshp


IBM Cloud API key

.....   [New](#) [+](#)


Number of instances

1 


Memory allocation per instance

64 MB  2000 MB [128](#)


Select region to deploy in

London 

Select an organization

Pavlo.Shcherbukha 


Select a space

dev 

Host

nod-red-wshp

Domain

eu-gb.mybluemix.net 


DevOps toolchain name

Accept the default name, or enter a value up to 100 characters.


nod-red-wshp

Select the region to create your toolchain in, and then select the resource group that provides access to your new toolchain.

Region

London 

Resource group

default 

В кінці кінців отримуємо фінальний екран:

Resource list / App details

nod-red-wshp Add tags

App details Developer resources

Name	Resources	Actions
Cloudant	Documentation	...

Your git repo

Cloudant Credentials - store them on your workstation

Continuous Delivery Remove from toolchain

https://eu-gb.git.cloud.ibm.com/Pavlo.Shcherb... [Redacted]

Toolchain

Name: nod-red-wshp
Location: London
Resource group: default
Tool Integrations: [Icons]

Delivery Pipelines

Name: nod-red-wshp
Status: In progress
Last input: Last commit by IBM Cloud (2 minutes ago)
Clone from zip

Credentials [Redacted] Show [Icon]

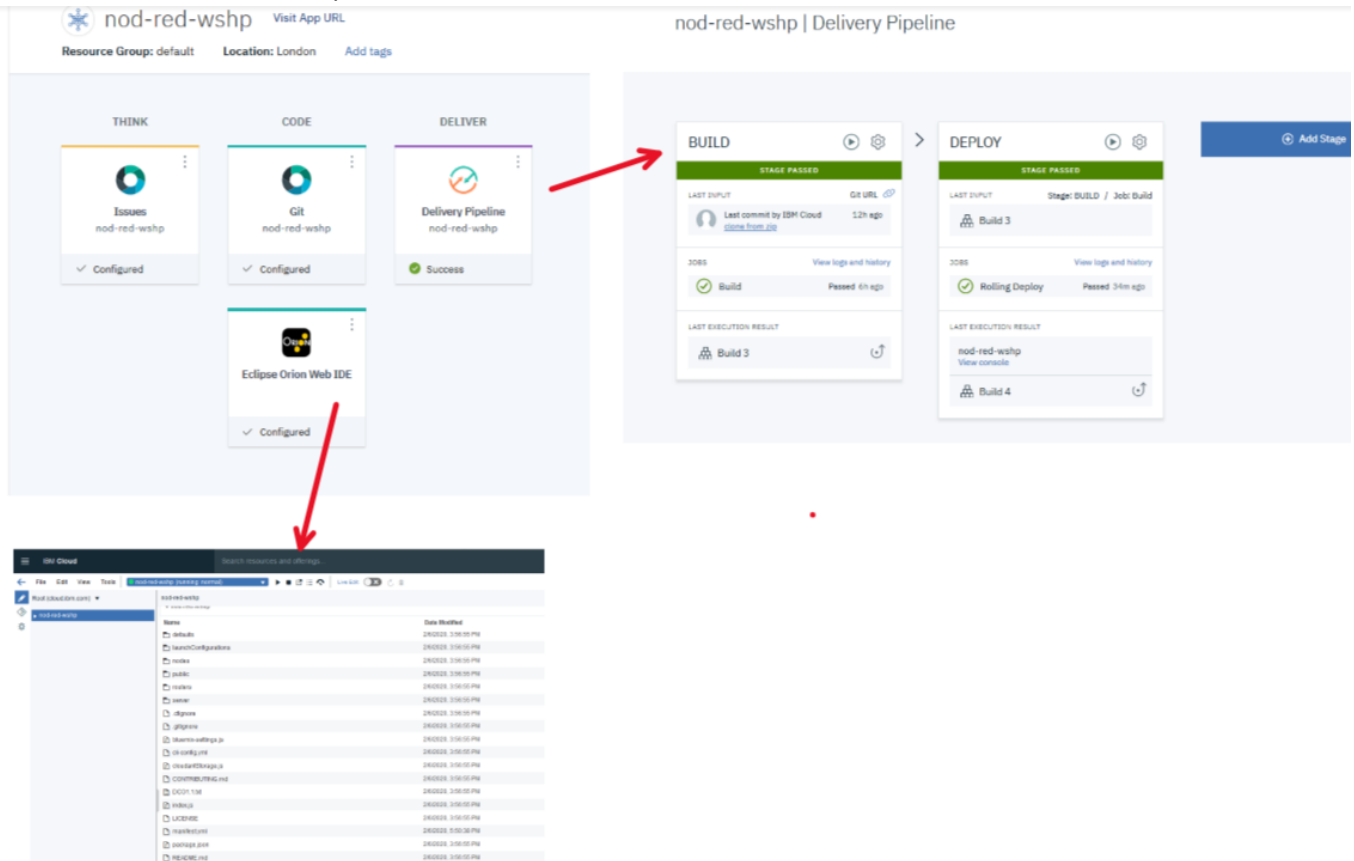
Go to the Cloud Git

View repo

з усіма нашими компонентами. А якщо перейти у вікно списку ресурсів, побачимо 3 компоненти

Cloud Foundry apps (4)				
Node-RED-app	Pavlo.Shcherbukha / dev	London	SDK for Node.js™	Started
Nodejs Express App YSLGL	Pavlo.Shcherbukha / dev	London	SDK for Node.js™	Started
nod-red-wshp	Pavlo.Shcherbukha / dev	London	SDK for Node.js™	Started
Apps (4)				
Node-RED-app	default	Global	Cloud Application	
Nodejs Express App YSLGL	default	Global	Cloud Application	
nod-red-wshp	default	Global	Cloud Application	
Developer tools (4)				
Node-RED-app-psh	default	London	Toolchain	
NodejsExpressAppYSLGL	default	London	Toolchain	
nod-red-wshp	default	London	Toolchain	

Далі, заходмо в toolchain і розглядаємо її складові



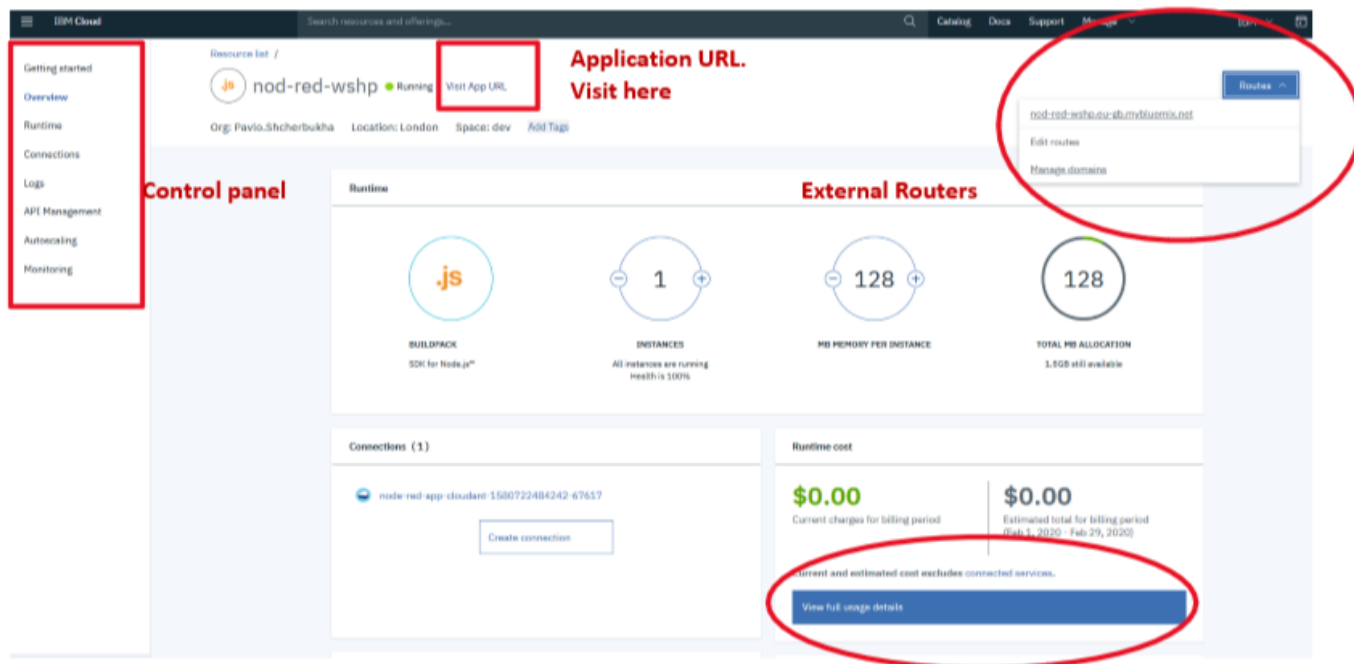
Тут бачимо елементи інтеграції з git. Середовище, для перегляду та зміни коду прямо в репозиторії. З цього середовища можна і deployment запустити, а можна і debug. Також, справа, бачимо складові "BUILD" та DEPLOY.

Перевірити, що сервіс запустився в хмарі

Для цього зайти в розділ CloudFoundry app вашого списку ресурсів та клікнути на вашому app.

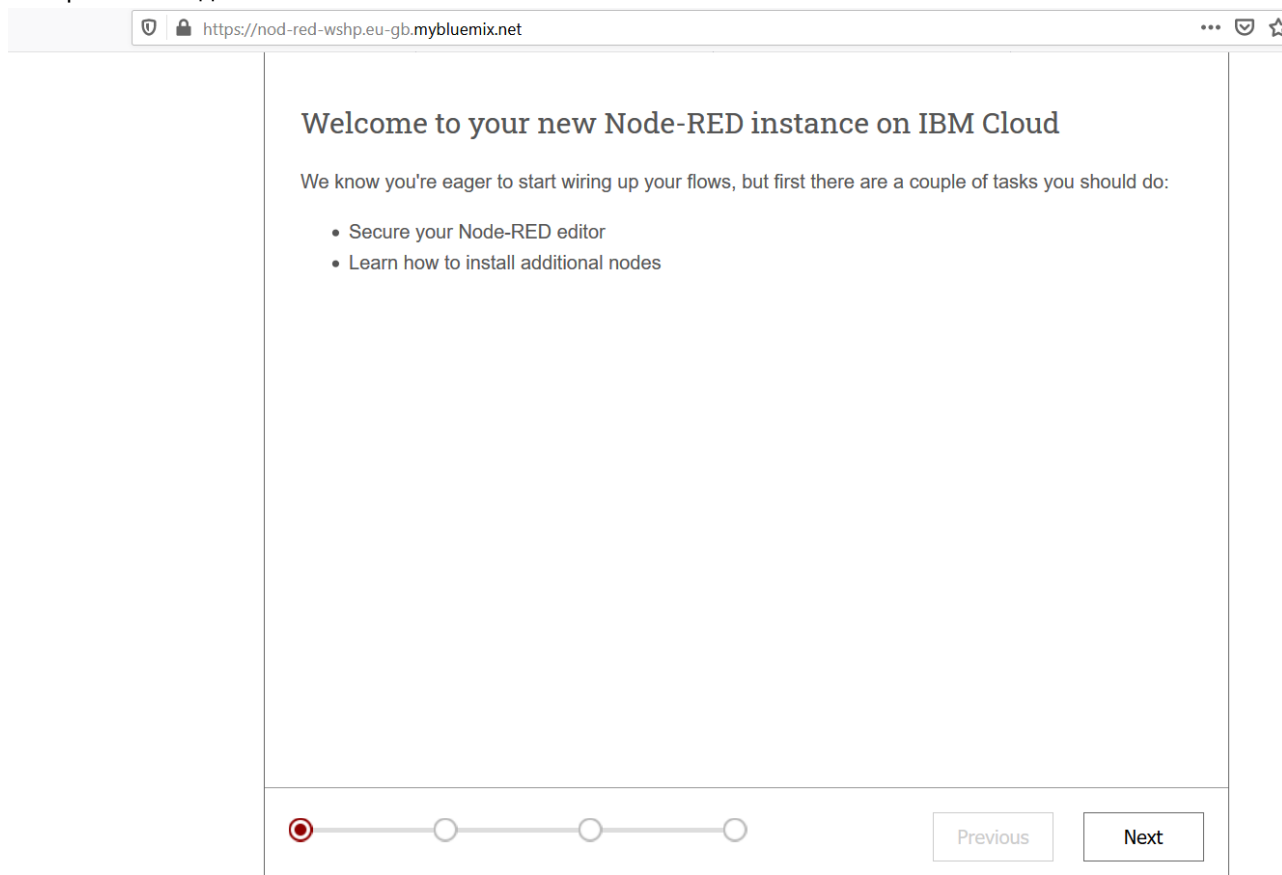
Cloud Foundry apps (4)				
node-red-app	Pavlo.Shcherbukha / dev	London	SDK for Node.js™	Started
Nodejs Express App YSLGL	Pavlo.Shcherbukha / dev	London	SDK for Node.js™	Started
nod-red-wshp	Pavlo.Shcherbukha / dev	London	SDK for Node.js™	Started

Відкриється робоча консоль



і потрібно зайти на **application URL** та зробити кілька кроків.

- запаролити вхід



(для простоти виберемо опцію "без пароллю")

Secure your Node-RED editor

☒ Secure your editor so only authorised users can access it

Username

Password Must be at least 8 characters

☐ Allow anyone to view the editor, but not make any changes

☐ **Not recommended: Allow anyone to access the editor and make changes**

Secure your Node-RED editor

☐ Secure your editor so only authorised users can access it

☒ **Not recommended: Allow anyone to access the editor and make changes**

Your editor will not be secured. Anyone with the URL will be able to access your flows, data and bound services.

☒ Tick this box to confirm you want your editor to be insecure

☒ ☐ ☐ ☐

Previous Next

- ознайомитися з посиланнями на ресурси

Learn how to install additional nodes

Node-RED provides a **huge catalog of extra nodes** you can install into the editor.

Many of these nodes can be installed directly from the editor's palette manager feature. However that can cause issues due to the limited memory of the default Node-RED starter application.

The *recommended approach* is to edit your application's `package.json` file to include the additional node modules and then redeploy the application. This can be done using the Continuous Delivery feature on the application's IBM Cloud dashboard.

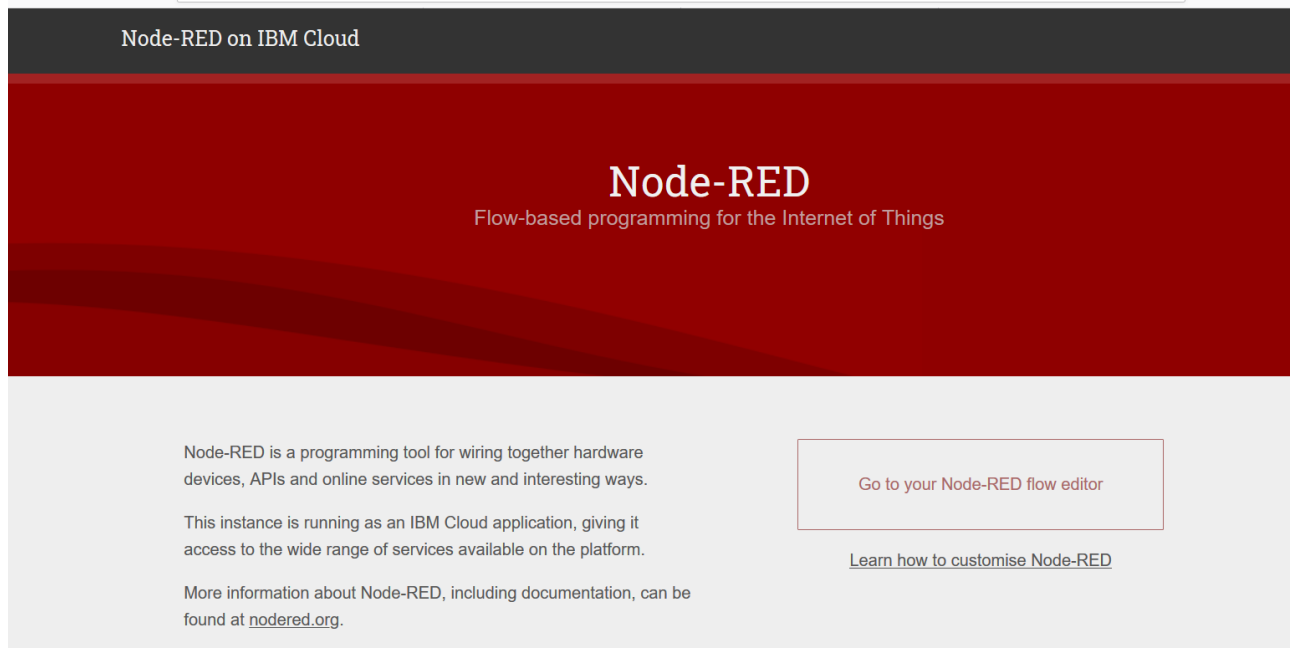
For more information, follow [this tutorial on IBM Developer](#).



Previous

Next

- Перейти на стартовий екран, та зайти в Flow Editor



В результаті побачимо тестовий потік:



Корисні ресурси

- [IBM Node-Red Starter](https://developer.ibm.com/tutorials/how-to-create-a-node-red- starter-application/) Тут в основному описано теж саме, що і в цій лабораторній роботі
- [Каталог додаткових вузлів та бібліотек](#)
- [Різні варіанти кастомізації IBM Node-RED](#)

Customising your instance of Node-RED

This instance of Node-RED is enough to get you started creating flows.

You may want to customise it for your needs, for example replacing this introduction page with your own, adding http authentication to the flow editor or adding new nodes to the palette.

To start customising your instance of Node-RED, you can either download the application locally or use IBM DevOps Services to edit and deploy your changes directly.

+ Securing the editor

+ Enabling Application Metrics for Node.js monitoring

+ Adding new nodes to the palette

+ Upgrading the version of Node-RED

+ Changing the static web content

+ Remove static web content and serve the flow editor from the root path

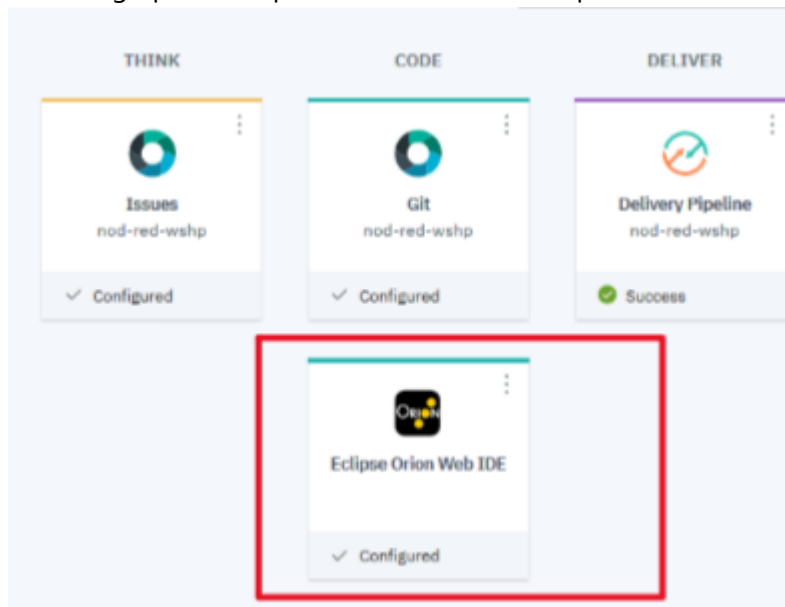
Захистити доступ до редактора потоків

Для цього потрібно установити env variables

```
NODE_RED_USERNAME - the username  
NODE_RED_PASSWORD - the password  
NODE_RED_GUEST_ACCESS - if set to `true`,  
read-only access to the editor
```

Дадавання нових бібліотек

- Зайти в git репозитрій нашого toolchain через Orion editor



або так, як показано на цьому малюнку

The screenshot shows the 'Continuous Delivery' page for a toolchain named 'NodeREDSSLPB'. At the top, there is a link to the repository: https://us-south.git.cloud.ibm.com/nick_oleary/NodeREDSSLPB. A red arrow points from a red box with the number '1' to this link. Below the link, the 'Toolchain' section displays the following details:

- Name:** NodeREDSSLPB
- Location:** Dallas
- Resource group:** default
- Tool integrations:** (Icons for Jenkins, Docker, and others)

Below the toolchain details, the 'Delivery Pipelines' section shows:

- Name:** NodeREDSSLPB
- Status:** In progress (with a refresh icon)
- Last input:** Last commit by IBM Cloud (19 seconds ago) with a link 'Clone from zip'.

- Знайти файл package.json та відкрити його для редагування (1)

The screenshot shows the 'IBM Cloud' repository interface for the 'NodeREDSSLPB' project. The left sidebar contains navigation options: Project, Details, Activity, Releases, Repository, Issues (0), Merge Requests (0), Wiki, Snippets, and Settings. The main area displays a list of files in the repository, all of which were cloned from zip 3 weeks ago. A red arrow points from a red box with the number '1' to the 'package.json' file in the list.

File Name	Source	Time
CONTRIBUTING.md	clone from zip	3 weeks ago
DCO1.1.txt	clone from zip	3 weeks ago
Dockerfile	clone from zip	3 weeks ago
Dockerfile-tools	clone from zip	3 weeks ago
LICENSE	clone from zip	3 weeks ago
README.md	clone from zip	3 weeks ago
bluemix-settings.js	clone from zip	3 weeks ago
cli-config.yml	clone from zip	3 weeks ago
cloudantStorage.js	clone from zip	3 weeks ago
index.js	clone from zip	3 weeks ago
manifest.yml	clone from zip	3 weeks ago
package.json	clone from zip	3 weeks ago
run-debug	clone from zip	3 weeks ago
run-dev	clone from zip	3 weeks ago
service.yaml	clone from zip	3 weeks ago

- додати потрібну бібліотеку (1) Додати commit message (2) Зберегти commit в git (3)

Edit file

The screenshot shows a code editor with a file named `package.json` on the `master` branch. The file content is as follows:

```
1 {
2   "name": "node-red-app",
3   "version": "1.1.1",
4   "dependencies": {
5     "node-red-dashboard": "2.x",
6     "@cloudant/cloudant": "^4.1.1",
7     "appmetrics-dash": "^5.2.0",
8     "bcrypt": "^3.0.7",
9     "body-parser": "1.x",
10    "cfenv": "^1.2.2",
11    "express": "4.x",
12    "http-shutdown": "1.2.1",
13    "node-red": "1.x",
14    "node-red-node-cf-cloudant": "0.x",
15    "node-red-node-openwhisk": "0.x",
16    "node-red-node-watson": "0.x",
17    "node-red-nodes-cf-sqlldb-dashdb": "0.x",
18    "ibm-cloud-env": "^0"
19  },
20  "scripts": {
21    "start": "node --max-old-space-size=160 index.js --settings ./bluemix-settings.js -v"
22  },
23  "engines": {
24    "node": "12.x"
25  }
26 }
```

Annotations:

- 1**: Points to the `"node-red-dashboard": "2.x",` line in the dependencies section.
- 2**: Points to the `Commit message` field containing `Add node-red-dashboard`.
- 3**: Points to the `Commit changes` button.

- У вас запуститься процес build (1) та deploy (2)

The screenshot shows a CI/CD pipeline with two stages: **BUILD** and **DEPLOY**.

BUILD Stage:

- Status: **STAGE PASSED**
- LAST INPUT: Last commit by Nicholas O'Leary 2m ago, [Add node-red-dashboard](#)
- JOBS: Build (Passed now)
- LAST EXECUTION RESULT: Build 2

DEPLOY Stage:

- Status: **STAGE RUNNING...**
- LAST INPUT: Stage: BUILD / Job: Build
- JOBS: Rolling Deploy (Running 23%)
- LAST EXECUTION: No results

Annotations:

- 1**: Points to the `Build` job in the BUILD stage.
- 2**: Points to the `Rolling Deploy` job in the DEPLOY stage.

Але, як на мене то з локальної станції краще. Тому в наступних лабораторних роботах розглянемо модифікований підхід