

Héberger son application Node.JS

Ce tutoriel vous expliquera simplement comment mettre en ligne votre application Node.JS exploitant les Websockets sous Windows, grâce à la plateforme OpenShift, fournie par RedHat.

Comme OpenShift utilise un dépôt Git pour synchroniser les sources de l'application, nous allons utiliser GitHub for Windows afin d'installer, configurer Git et générer une clé SSH simplement.

→ Créer un compte sur GitHub <https://github.com/>



→ Télécharger et installer GitHub for Windows <http://windows.github.com/>

→ Se connecter avec l'application « GitHub » sur le compte GitHub créé précédemment et remplir les champs demandés (nom et e-mail)

- Vous devriez recevoir un mail indiquant qu'une clé publique SSH a été créée pour l'ordinateur.
- Vous pouvez fermer Git for Windows.

Une fois Git installé et configuré, vous pouvez utiliser OpenShift :

→ Créer un compte sur OpenShift : <https://www.openshift.com/>

→ Confirmez votre adresse e-mail et acceptez les conditions

→ Connecter vous avec vos identifiants : <https://openshift.redhat.com/app/login>

→ Le site vous proposera automatiquement de créer une application.

Etape 1 : « Choose a type of application », recherchez et sélectionnez « Node.js 0.6 »

1 Choose a type of application 2 Configure and deploy the application 3 Next steps

Choose a web programming cartridge or kick the tires with a quickstart. After you create the application you can add **cartridges** to enable additional capabilities like databases, metrics, and continuous build support with Jenkins.

Node.js Browse by tag... ▾

Cartridge
A managed runtime that receives security updates and upgrades automatically.

QuickStart
An quick way to try out a new technology with code and libraries preconfigured. You are responsible for updating core libraries for security updates.

Matches search 'Node.js'

Node.js 0.6

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js is perfect for data-intensive real-time applications that run across distributed devices.

<http://www.nodejs.org/>

OpenShift maintained

JAVASCRIPT NODEJS

Etape 2 : Entrez le nom de l'application et choisir un nom de domaine qui sera utilisé par les autres applications créées sur le compte et cliquez sur le bouton « Create Application ».

http:// testtuto - totototo .rhcloud.com

Because this is your first application, you need to provide a domain name under which your applications will be grouped

OpenShift will automatically register this domain name for your application. You can add your own domain name later.

Source Code

Default [Change](#)

We'll create a Git code repository in the cloud, and populate it with a set of reasonable defaults.

Gears

Small

Gears are the application containers running your code. For most applications, the 'small' gear size provides plenty of resources.

Cartridges

Node.js 0.6

Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.

Scaling

No scaling [Change](#)

OpenShift automatically routes web requests to your web gear. If you allow your application to scale, we'll set up a load balancer and allocate more gears to handle traffic as you need it.

[Back](#) [Create Application](#)

Etape 3 : Ajouter la clé publique SSH créée par Git.

- ➔ Allez dans l'explorateur Windows, assurez-vous que vous **affichez bien les fichiers et dossiers cachés** et allez dans le dossier « C:\Users\NomUtilisateur\.ssh »
- ➔ Ouvrez le fichier github_rsa.pub avec un éditeur de texte
- ➔ Collez le contenu dans le champ texte prévu à cette effet sur OpenShift et cliquez sur le bouton « Save ».

Set your Public Key

Before you can upload code, you need to provide us with a **public key** to identify you to our servers. If this is your first time creating a public key [read more about SSH keys and remote access](#).

Paste the contents of your public key file (.pub)

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA6vV916WqJzmLxtbSBULRJpgoxHA9yjAxUamtccPR7/2q3Dc5YE4tvRVR
Fgnx0fzNbRkLMNo65fkBYdRiNRA3WHBxfIEEIQiTwqGGCh/rdakJUvK30CmcYTaqxkr3+IMHAtuufSSBjaHiTbbKC
SDJpz+ubdtUBChnoe5GjmTnJBzorweElrAGjMrJxt01KHa6T/4FXdu94fbxlnpH9au+d6g7N11vdzX/zuBbCTOGKZsJw
8r9wSg7IUSuaehtj7zVthHNa/mr7GUJMOGI5lydr5iQ0gqp6emrlpFvBxt8x+wmnXlzxsFCIVkZlciP9c+RBprXSle/MYU
```

[Save](#)

Un dépôt Git est créé, dans notre exemple, il s'agit de :

Making code changes

OpenShift uses the [Git version control system](#) to manage the code of your application. Each cartridge has a single Git **repository** that you'll use to check in changes to your application. When you **push** a change to your Git repository we'll automatically deploy your code and restart your application if necessary.

Install the Git client for your operating system, and from your command line run

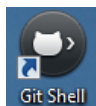
```
git clone ssh://517a859a5973ca08a20000ba@testtuto-totototo.rhcloud.com/~/.git/testtuto.git/
cd testtuto/
```

This will create a folder with the source code of your application. After making a change, **add**, **commit**, and **push** your changes.

```
git add .
git commit -m 'My changes'
git push
```

When you push changes the OpenShift server will report back its status on deploying your code. The server will run any of your configured [deploy hooks](#) and then restart the application.

Il est désormais possible d'utiliser Git pour envoyer ses fichiers sur les serveurs d'OpenShift.



- Lancez Git Shell
- Accéder au répertoire « C:\Users\NomUtilisateur\Documents\GitHub » (si vous n'y êtes pas déjà), tapez :

```
cd c:\Users\NomUtilisateur\Documents\GitHub
```

- Taper les commandes suivantes tel qu'expliqué dans sur le site dans la section « Making code changes » dans le premier encadré
 - Cela crée un répertoire portant le nom de l'application dans le répertoire GitHub
- Les fichiers de votre applications seront contenus dans répertoire créé
C:\Users\NomUtilisateur\Documents\GitHub\testtuto\
- Supprimez le dossier « node_modules »
- Remplacez-le par le dossier node_modules de votre projet
- Supprimer les fichiers index.html et server.js
- Placez les fichiers sources de votre application dans le répertoire
- Renommez le fichier lancé par Node.JS en « server.js »
- Dans le fichier « server.js », remplacez la ligne suivante

```
// launch the http server on given port
server.listen(8080);
```

Par :

```
var ipaddr = process.env.OPENSIFT_NODEJS_IP || "127.0.0.1";
var port   = process.env.OPENSIFT_NODEJS_PORT || 8080;

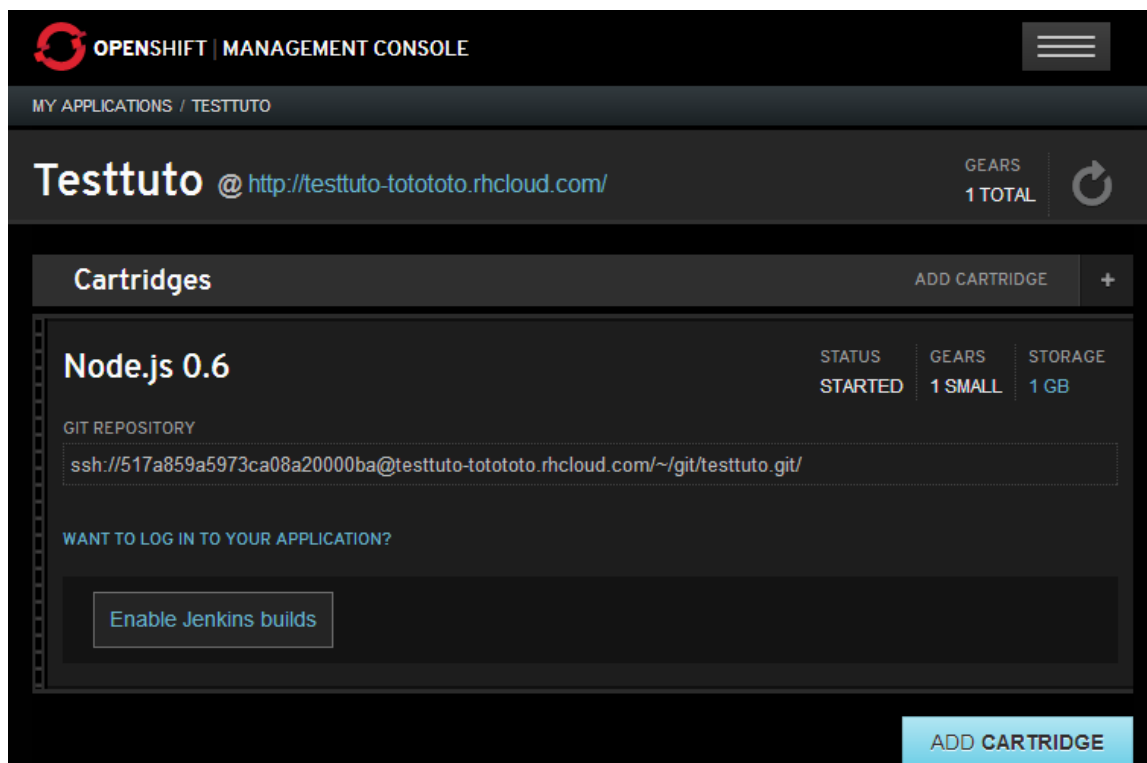
// launch the http server on given port
server.listen(port, ipaddr);
```

→ Pour envoyer les fichiers vers le serveur, taper dans le shell les commandes suivantes

```
git add .  
  
git commit -m 'message_de_log'  
  
git push
```

Votre application est désormais hébergée par Openshift, pour accéder à l'application, utiliser le port 8000, sinon les websockets ne fonctionnent pas !

Dans notre exemple, la console de gestion d'OpenShift indique :



OPENSIFT | MANAGEMENT CONSOLE

MY APPLICATIONS / TESTTUTO

Testtuto @ <http://testtuto-totototo.rhcloud.com/>

GEARS
1 TOTAL

Cartridges ADD CARTRIDGE +

	STATUS	GEARS	STORAGE
Node.js 0.6	STARTED	1 SMALL	1 GB

GIT REPOSITORY

`ssh://517a859a5973ca08a20000ba@testtuto-totototo.rhcloud.com/~/.git/testtuto.git/`

WANT TO LOG IN TO YOUR APPLICATION?

Enable Jenkins builds

ADD CARTRIDGE

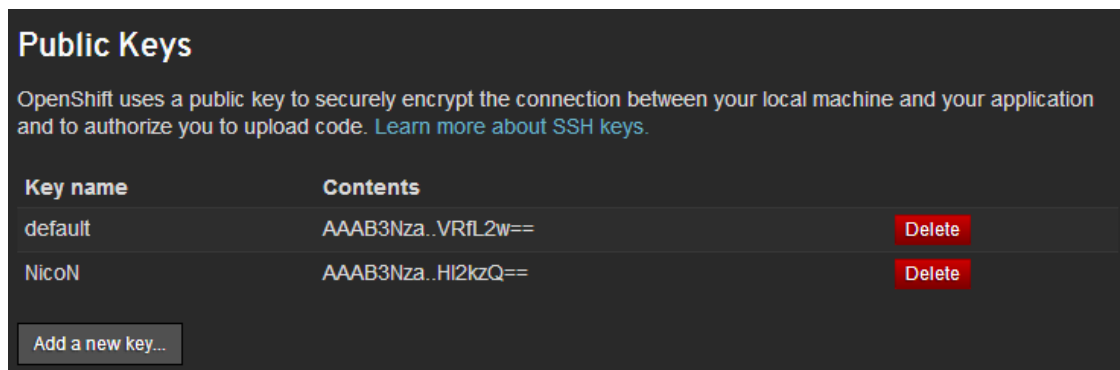
Mais afin que les WebSockets fonctionnent, il est nécessaire d'y accéder par l'URL :

<http://testtuto-totototo.rhcloud.com:8000/>

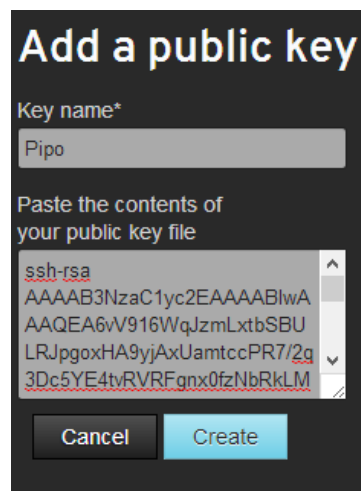
Notez que le bouton actualiser en haut à droite permet de redémarrer l'application en cas de problème.

Ajouter d'autres participants au projet

- Chaque participant doit ouvrir un compte GitHub et avoir « GitHub for Windows » d'installé, connecté et configuré sur leur machine.
- Ils doivent ensuite communiquer au créateur du projet le contenu de leur fichier github_rsa.pub.
- Le créateur doit ajouter les clés de ses partenaires sur OpenShift en accédant aux paramètres de son compte (« My Account » puis « Change my settings ») puis dans la section « Public Keys », cliquer sur « Add a new Key ».



- Mettre le nom souhaité à la clé du partenaire et collez la clé publique associée, et cliquez sur le bouton « Create ».



- Enfin les partenaires peuvent accéder au contenu du dépôt git en exécutant la commande « git clone » sur le dépôt git indiqué dans l'interface de gestion de l'application.
- Enfin ils peuvent effectuer des modifications et les envoyer au serveur grâce aux commandes

```
git add .  
git commit -m 'message_de_log'  
git push
```

- Pour mettre à jour le contenu du répertoire local au fur et à mesure de l'évolution du projet, il suffit de lancer « Git Shell », d'accéder au répertoire du projet et d'exécuter la commande :

```
git pull
```