

DEVOPS

Jenkins

Objectifs

- Utiliser Jenkins pour le CI intégration continue
- Installer Jenkins
- Configurer Jenkins avec Docker, Jobs DSL et Jenkins pipelines
- Utiliser des plugins pour l'intégration de Jenkins avec différents softwares
- Configurer les options d'authentification et d'autorisation pour renforcer la sécurité.

C'est quoi Jenkins

- Un CI/CD outil open source écrit en Java
- Un serveur d'automatisation utilisé pour construire (build) et livrer un projet logiciel
- Beaucoup de plugins disponibles
- Open source, gratuit et populaire

CI

- l'intégration continue (CI) est la pratique consistant à fusionner toutes les copies de travail des développeurs sur une ligne principale partagée plusieurs fois par jour - wikipédia

CD

- La livraison continue (CD) est une approche d'ingénierie logicielle dans laquelle les équipes produisent des logiciels dans des cycles courts, garantissant que le logiciel peut être publié de manière fiable à tout moment et, en suivant un pipeline à travers un « environnement de type production », sans le faire manuellement -
wikipédia

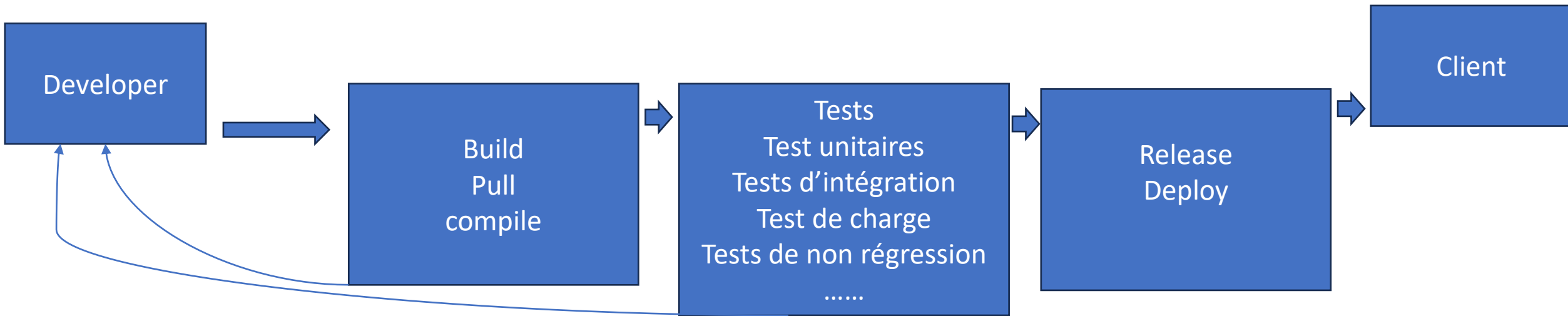
CI/CD

- vérifier et publier les travaux en déclenchant des builds et des tests automatisés
- Au moins une fois par jour (on push, on pull request)
- Tous les développeurs push leurs changements to git repository leur travail sera buildé et testé par un outil CI/CD (gitHub actions, Jenkins)
- Jenkins ne fusionne pas le code et ne résout pas les conflits (c'est aux développeurs de le faire en utilisant une version de contrôle: git par exemple)

Avantages Jenkins

- Donne un retour au développeur pour fixer les éventuelles erreurs qui surviendront lors des étapes de pipeline
- Il est plus simple pour un développeur de corriger un problème tant que c'est frais dans sa mémoire
- Jenkins publie les artéfacts de votre software
- Les artéfacts passent par des tests automatisés avant d'être produits
- Le but étant de raccourcir le SDLC

CI/CD dans SLDC



Jenkins concurrents

- DroneCI
- TeamCity
- Wercker
- CircleCI
- CodeShip
- SemaphoreCI
- Amazon AWS CI/CD tools

Installation de Jenkins

- On peut installer Jenkins directement sur notre OS

Windows :

<https://www.jenkins.io/doc/book/installing/windows/>

Ubuntu :

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04>

- On peut l'installer via docker

<https://medium.com/@eloufirhatim/install-jenkins-using-docker-e76f41f79682>

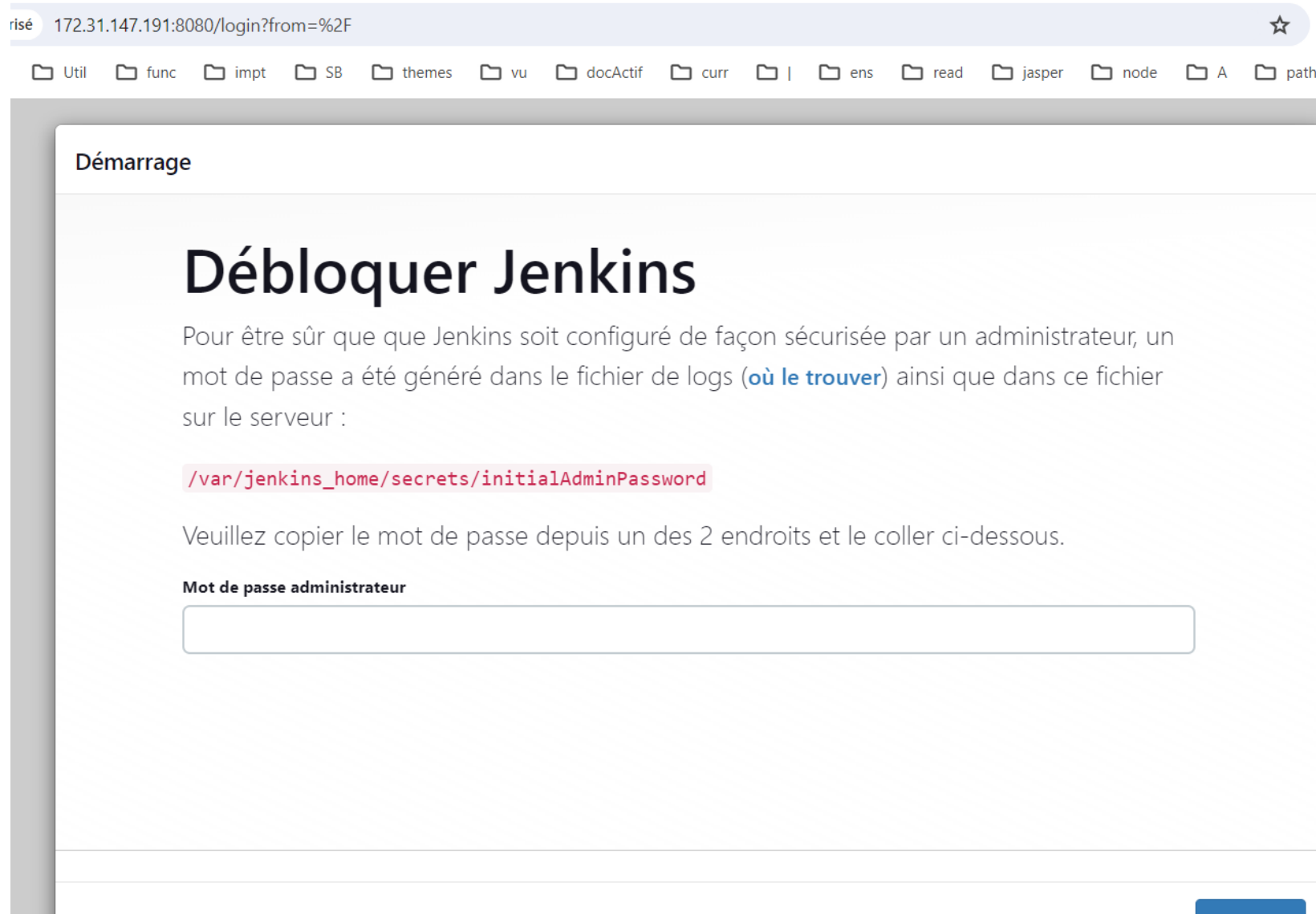
Installation sur docker

- `mkdir -p /var/jenkins_home`
- `chown -R 1000:1000 /var/jenkins_home/`
- `docker run -p 8080:8080 -p 50000:50000 -v /var/jenkins_home:/var/jenkins_home -d --name jenkins jenkins/jenkins:lts`

Installation sur docker

```
abdou@abdou-Virtual-Machine:/var$ sudo docker run -p 8080:8080 -p 50000:50000 -v /var/jenkins_home:/var/jenkins_home -d --name jenkins jenkins/jenkins:lts
Unable to find image 'jenkins/jenkins:lts' locally
lts: Pulling from jenkins/jenkins
90e5e7d8b87a: Pull complete
f46e108600e7: Pull complete
67ee1cced2cb: Pull complete
3dfb3d6cb44e: Pull complete
46014cb180c0: Pull complete
20f6ee282779: Pull complete
f8ee3f626f99: Pull complete
b636b23fb8d6: Pull complete
a3ffaaf08d83: Pull complete
fc8874a22e0d: Pull complete
eef971f7a8d4: Pull complete
fb46046a6402: Pull complete
Digest: sha256:186a48ae298e34a21b27fe737bf0a854c3e73421ce858c4d40c403802589e23f
Status: Downloaded newer image for jenkins/jenkins:lts
9ec491fa94956051e2047a04a103ca8929998810d52848dd2511308e4fab3d74
```

Le port par défaut est 8080



Mettre le mot de passe et continuer

Débloquer Jenkins

Pour être sûr que Jenkins soit configuré de façon sécurisée par un administrateur, un mot de passe a été généré dans le fichier de logs ([où le trouver](#)) ainsi que dans ce fichier sur le serveur :

```
/var/jenkins_home/secrets/initialAdminPassword
```

Veuillez copier le mot de passe depuis un des 2 endroits et le coller ci-dessous.

Mot de passe administrateur

Continuer

Télécharger les plugins

Personnaliser Jenkins

Les plugins étendent Jenkins avec des fonctionnalités additionnelles pour satisfaire différents besoins.

Installer les plugins suggérés

Installer les plugins que la communauté Jenkins trouve les plus utiles.

Sélectionner les plugins à installer

Sélectionner et installer les plugins les plus utiles à vos besoins.

Installation en cours...



☒ Folders	☐ OWASP Markup Formatter	☐ Build Timeout	☐ Credentials Binding	** Ionicons API Folders OWASP Markup Formatter
☐ Timestampper	☐ Workspace Cleanup	☐ Ant	☐ Gradle	
☐ Pipeline	☐ GitHub Branch Source	☐ Pipeline: GitHub Groovy Libraries	☐ Pipeline: Stage View	
☐ Git	☐ SSH Build Agents	☐ Matrix Authorization Strategy	☐ PAM Authentication	
☐ LDAP	☐ Email Extension	☐ Mailer		

Créer le 1er utilisateur Administrateur

Nom d'utilisateur

abdou

Mot de passe

.....

Confirmation du mot de passe

.....

Nom complet

Abderrahman CHEIKH AHMEDOU

Adresse courriel

sidi.mohamed@gti-intl.com

Configuration de l'instance

URL de Jenkins :

L'URL de Jenkins est utilisée pour fournir l'URL de base pour les liens absolus vers les diverses ressources Jenkins. Cela signifie que cette valeur est nécessaire pour le bon fonctionnement de nombreuses fonctionnalités de Jenkins, notamment les notifications par mail, les mises à jour des statuts des pull requests, et la variable d'environnement BUILD_URL fournie pour les étapes de build.

La valeur par défaut affichée **n'est pas encore sauvegardée** et est générée à partir de la requête actuelle, lorsque c'est possible. Il est fortement recommandé d'utiliser comme valeur l'URL qui est censée être utilisée par les utilisateurs. Cela évitera des confusions lors du partage ou de la visualisation de liens.

Jenkins is almost ready!

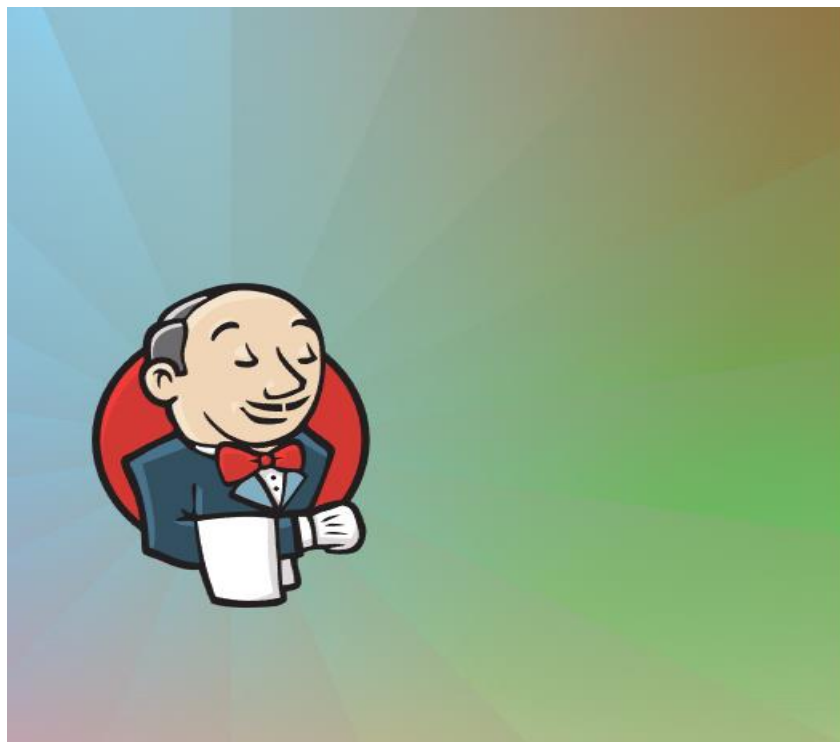
Vous avez passé la **configuration de l'instance**.

Vous pouvez toujours vous rendre sur les pages de configuration pour remplir les informations qui pourraient manquer.

Your Jenkins setup is complete, but some plugins require Jenkins to be restarted.

Redémarrer

Se connecter



Sign in to Jenkins

Utilisateur

Mot de passe

☐ Garder ma session ouverte

S'identifier



Nouveau Item

Ajouter une description

Utilisateurs

Historique des constructions

Administrer Jenkins

Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

1 Au repos

2 Au repos

Bienvenue sur Jenkins !

Vos jobs Jenkins seront affichés sur cette page. Pour commencer, vous pouvez mettre en place un build distribué ou commencer à créer un projet.

Commencer à créer votre projet

Créer un job



Configurer un build distribué

Mettre en place un agent



Configurer un cloud



En apprendre plus sur les builds distribués



Builder et déployer une application

- On va utiliser nodejs (javascript runtime environment)
- Nodjs
 - open source
 - Cross-plateforme
 - Exécute javascript coté serveur (Backend)
 - Event driven architecture asynchrone I/O, réponse aux requêtes rapide, le traitement qui prend du temps est asynchrone

Comment builder une application nodejs

- Npm init
- Npm install
- Npm test
- Packager on peut le faire dans docker
- Distribuer l'image docker, on peut utiliser docker registry pour publier l'image

Créer une application nodejs

- Npm init -y

```
PS D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js> npm init -y
Wrote to D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js\package.json:

{
  "name": "exemple_node_js",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
PS D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js> ls

Directory: D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js

Mode                LastWriteTime         Length Name
----                -
-a-----          1/19/2024  4:46 PM             94 package-lock.json
```

Ajouter axios

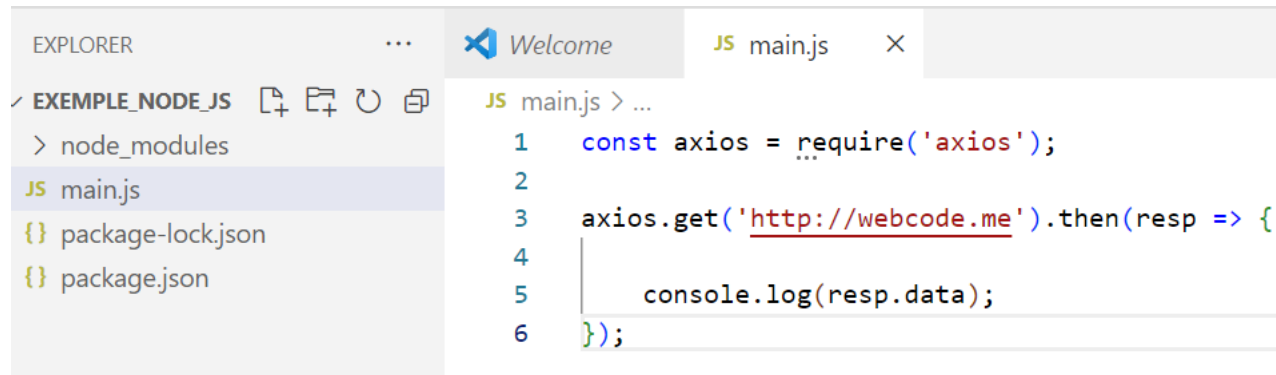
npm i axios

```
PS D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js> npm i axios

added 9 packages, and audited 10 packages in 5s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'EXEMPLE_NODE_JS'. It includes a 'node_modules' directory and files 'main.js', 'package-lock.json', and 'package.json'. The 'main.js' file is selected. The main editor area shows the content of 'main.js', which contains a JavaScript snippet that uses the 'axios' library to perform a GET request to 'http://webcode.me' and log the response data to the console.

```
JS main.js > ...
1  const axios = require('axios');
2
3  axios.get('http://webcode.me').then(resp => {
4
5      console.log(resp.data);
6  });
```

Ajouter un serveur express

```
PS D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js> npm i express
```

```
added 60 packages, and audited 70 packages in 4s
```

```
12 packages are looking for funding
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
const express = require('express')
const app = express()
const port = 3000

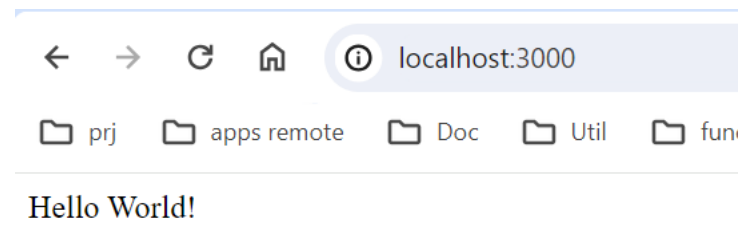
app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

```
PS D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js> npm start
```

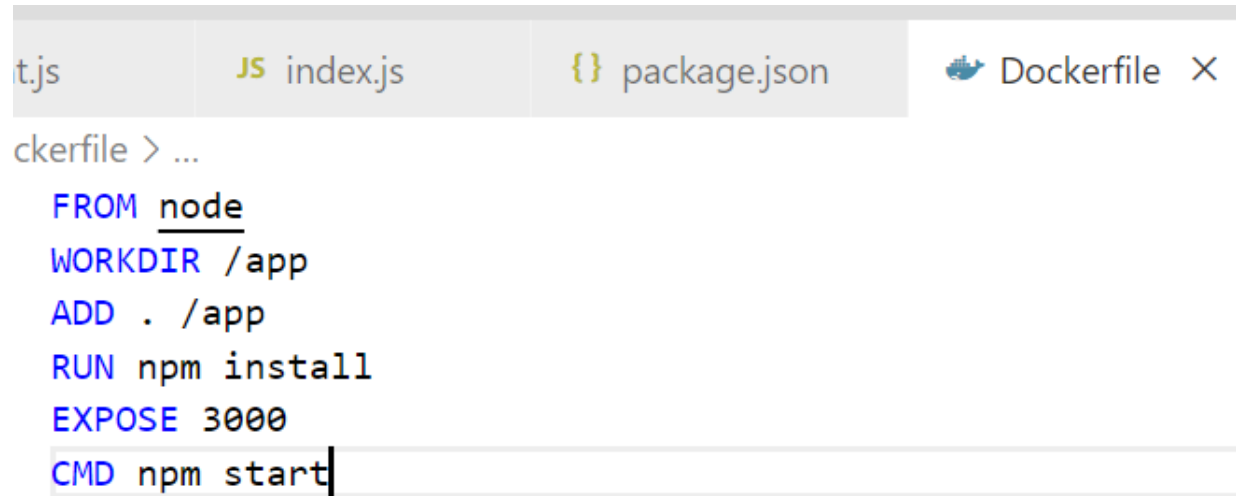
```
> exemple_node_js@1.0.0 start
> node index.js
```

```
Example app listening on port 3000
```



```
PS D:\workspace\2.Training\Ens\InstNumerisation\2024\s3\tp_exmples\jenkins_exemples\exemple_node_js> node .\main.js
Hello World!
```

Docker file

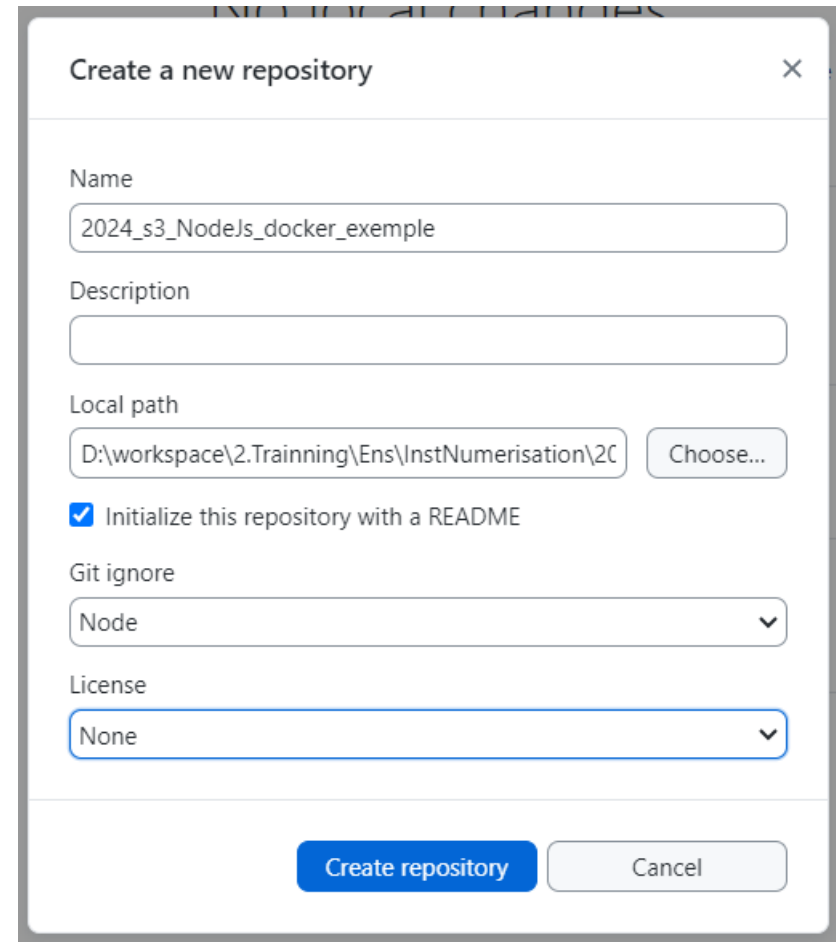


The image shows a code editor window with four tabs at the top: 't.js', 'JS index.js', '{} package.json', and 'Dockerfile X'. The 'Dockerfile X' tab is active. Below the tabs, the text 'ckerfile > ...' is visible. The main content of the Dockerfile is as follows:

```
FROM node
WORKDIR /app
ADD . /app
RUN npm install
EXPOSE 3000
CMD npm start
```

Créer un repository github avec notre application nodejs

- On peut le faire facilement avec gitHubDesktop
- Allez dans file/new repository



The screenshot shows the 'Create a new repository' dialog box in GitHub Desktop. The dialog has a title bar with a close button. It contains several input fields and checkboxes. The 'Name' field is filled with '2024_s3_NodeJs_docker_exemple'. The 'Description' field is empty. The 'Local path' field is filled with 'D:\workspace\2.Training\Ens\InstNumerisation\2C', and there is a 'Choose...' button next to it. A checkbox labeled 'Initialize this repository with a README' is checked. The 'Git ignore' dropdown is set to 'Node'. The 'License' dropdown is set to 'None'. At the bottom, there are two buttons: 'Create repository' (blue) and 'Cancel' (gray).

Create a new repository

Name
2024_s3_NodeJs_docker_exemple

Description

Local path
D:\workspace\2.Training\Ens\InstNumerisation\2C Choose...

☒ Initialize this repository with a README

Git ignore
Node

License
None

Create repository Cancel

Publier le repository

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Publish your repository to GitHub

This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.

Always available in the toolbar for local repositories or **Ctrl** + **P**

Publish repository

Publish repository



GitHub.com

GitHub Enterprise

Name

2024_s3_NodeJs_docker_exemple

Description

☒ Keep this code private

Organization

None



Publish repository

Cancel



sidielmoctar / 2024_s3_NodeJs_docker_exemple

🔍 Type to search

<> Code

🕒 Issues

🔗 Pull requests

🎬 Actions

📁 Projects

🛡 Security

📈 Insights

⚙ Settings



2024_s3_NodeJs_docker_exemple

Private

👁 Unwatch

1

🔗 main ▾

🔗 1 Branch 🔖 0 Tags

🔍 Go to file

t

Add file ▾

<> Code ▾



sidielmoctar Initial commit

5450bd1 · 1 minute ago

🕒 1 Commits



.gitattributes

Initial commit

1 minute ago



.gitignore

Initial commit

1 minute ago



README.md

Initial commit

1 minute ago



README



2024_s3_NodeJs_docker_exemple

Rendre votre repository public

Danger Zone

Change repository visibility

This repository is currently private.

Change visibility

Make sidiemoctar/2024_s3_NodeJs_docker_exemple public



sidiemoctar/2024_s3_NodeJs_docker_exemple

☆ 0 stars 👁 1 watcher

I want to make this repository public

Builder notre premier application

- Administrer Jenkins/ plugins/ nodejs

+ Nouveau Item

Utilisateurs

Historique des constructions

Administrer Jenkins

Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

1 Au repos

2 Au repos

Administrer Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

L'URL Jenkins est vide cependant elle est nécessaire au bon fonctionnement de nombreuses fonctionnalités de Jenkins telles que les notifications par e-mail, la mise à jour du statut des telles que BUILD_URL.

Veuillez renseigner la valeur dans [Configuration de Jenkins](#).

System Configuration

System

Configurer les paramètres généraux et les chemins de fichiers.

Tools

Configurer les outils, leur localisation et les installeurs automatiques.

PlUGINS

Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

Clouds

Add, remove, and configure cloud instances to provision agents on-

Plugins disponibles/ nodeJs

 **Jenkins**

rechercher (CTRL+K) ?

 1  1  **Abderrahman CHEIKH AHMEDOU** ▾  se déconnecter

Tableau de bord > Administrer Jenkins > Plugins

Plugins

Mises à jour

Plugins disponibles

Plugins installés

Paramètres avancés

Progression des téléchargements

Q node /

Installer

Installer	Nom ↑	Publié
	NodeJS 1.6.1 npm NodeJS Plugin executes NodeJS script as a build step.	5 mo. 6 j ago

Choisir installer après démarrage

Installer

Nom ↓

Publié

☒

[NodeJS](#) 1.6.1
[npm](#)
NodeJS Plugin executes [NodeJS](#) script as a build step.

5 mo. 6 j ago

Installer

↓

Installer après le redémarrage

Plugins

-  Mises à jour
-  Plugins disponibles
-  Plugins installés
-  Paramètres avancés
-  Progression des téléchargements

Progression du téléchargement

Préparation	• Checking internet connectivity • Checking update center connectivity • Success
Ionicons API	Succès
Folders	Succès
OWASP Markup Formatter	Succès
JSON Path API	Succès
Structs	Succès
Pipeline: Step API	Succès
Token Macro	Succès
Build Timeout	Succès
bouncycastle API	Succès
Instance Identity	Succès
JavaBeans Activation Framework (JAF) API	Succès



Veillez attendre pendant que Jenkins redémarre ...

Votre navigateur se rafraîchira automatiquement quand Jenkins sera prêt.

Redémarrage sans danger

Les builds sur les agents peuvent généralement se poursuivre.

Configurer un job manuellement

- Revenir vers le tableau de bord et créer un job

Saisissez un nom

» Champ obligatoire



Construire un projet free-style
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline
Organise des activités de longue durée qui peuvent s'étendre sur plusieurs agents de construction. Adapté pour la création des pipelines (anciennement connues comme workflows) et/ou pour organiser des activités complexes qui ne s'adaptent pas facilement à des tâches de type libre.

- ☐ Supprimer les anciens builds ?
- ☐ Throttle builds ?
- ☐ Exécuter des builds simultanément si nécessaire ?

Avancé ▾

Gestion de code source

☐ Aucune

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git

Credentials ?

- aucun -

+ Ajouter ▾

Étapes du build

Ajouter une étape au build ^

Filter

Appeler Ant

Exécuter un script shell

Exécuter une ligne de commande batch Windows

Exécuter le script NodeJS

☰ Exécuter un script shell ?

Commande

Voir [la liste des variables d'environnement disponibles](#)

```
npm install
```

Avancé ▾

Ajouter une étape au build ▾

Actions à la suite du build

Ajouter une action après le build ▾

Sauvegarder

Appliquer

- Administrer Jenkins/
tools/ ajouter nodeJs

NodeJS

Nom

nodejs

! Required

☒ Install automatically ?

≡ Installer depuis nodejs.org

Version

NodeJS 21.6.1

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

☐ Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the `packageName@version`

Global npm packages refresh hours

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

72

Ajouter un installeur ▾

Ajouter NodeJS

Enregistrer Appliquer

+ Nouveau Item

 Ajouter une description

 Utilisateurs

Tous



 Historique des constructions

 Administrer Jenkins

 Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
		Exemple nodejs	s. o.	s. o.	ND

Icône: S M L Légende

 Modifications

 Répertoire de travail

 Lancer un build

 Configurer

 Supprimer Projet

 Renommer

Atom feed for failures

 Atom feed for just latest builds

Environnements de Build

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Provide Configuration files ?
- ☐ Add timestamps to the Console Output
- ☒ Ajouter le répertoire bin/ de Node/npm au PATH

NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

nodejs

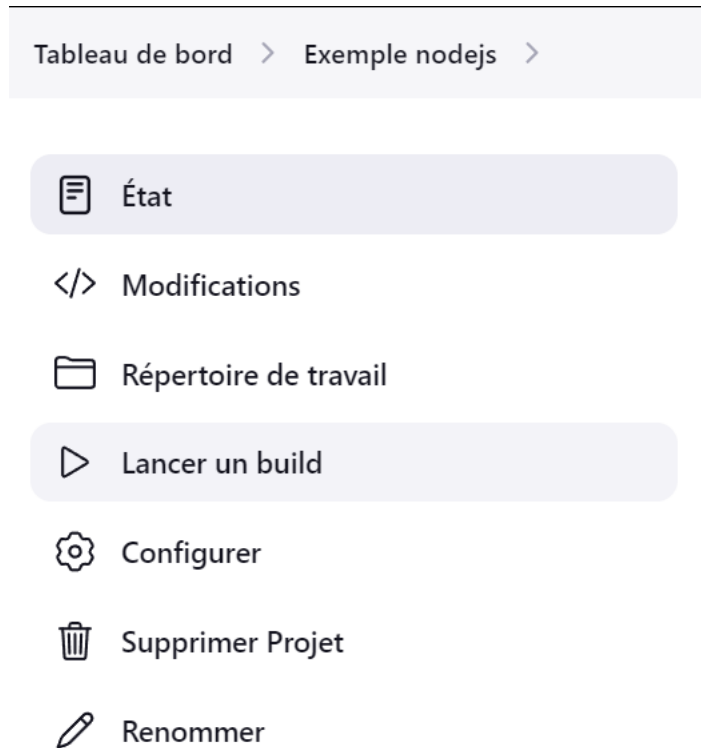


npmrc file

utiliser le système par défaut



- Puis lancer le build



Avancé ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Navigateur de la base de code ?



Historique des builds

tendance ▼



Filter...



✕ #4

23 janv. 2024 13:35





Sortie de la console

Progression:  

```
Started by user Abderrahman CHEIKH AHMEDOU
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Exemple nodejs
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/Exemple nodejs/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git #
timeout=10
Fetching upstream changes from https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git fetch --tags --force --progress -- https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision f356ba9bb5df34d919da6c8619f38acd4f23ae7d (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f f356ba9bb5df34d919da6c8619f38acd4f23ae7d # timeout=10
Commit message: "add files"
First time build. Skipping changelog.
Unpacking https://nodejs.org/dist/v21.6.1/node-v21.6.1-linux-x64.tar.gz to
/var/jenkins_home/tools/jenkins.plugins.nodejs.tools.NodeJSInstallation/nodejs on Jenkins
```



Sortie de la console

```
Started by user Abderrahman CHEIKH AHMEDOU
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Exemple nodejs
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/Exemple nodejs/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git #
timeout=10
Fetching upstream changes from https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git
> git --version # timeout=10
> git --version # 'git version 2.39.2'
> git fetch --tags --force --progress -- https://github.com/sidielmoctar/2024_s3_NodeJs_docker_exemple.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision f356ba9bb5df34d919da6c8619f38acd4f23ae7d (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f f356ba9bb5df34d919da6c8619f38acd4f23ae7d # timeout=10
Commit message: "add files"
First time build. Skipping changelog.
Unpacking https://nodejs.org/dist/v21.6.1/node-v21.6.1-linux-x64.tar.gz to
/var/jenkins_home/tools/jenkins.plugins.nodejs.tools.NodeJSInstallation/nodejs on Jenkins
[Exemple nodejs] $ /bin/sh -xe /tmp/jenkins17541496790778553506.sh
+ npm install

added 69 packages, and audited 70 packages in 16s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
Finished: SUCCESS
```

Configurer un job avec Docker

- Administrer Jenkins/ plugins/ docker

Plugins

cloudBees docker

Installer

Installer après le redémarrage

Installer	Nom ↓	Publié
☒	CloudBees Docker Build and Publish 1.4.0 Outils de build docker This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.	Il y a 1 an. 5 mo.

Progression du téléchargement

Préparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Authentication Tokens API

! Téléchargé avec succès. Sera activé lors du prochain démarrage

Docker Commons

! Téléchargé avec succès. Sera activé lors du prochain démarrage

SSH server

! Téléchargé avec succès. Sera activé lors du prochain démarrage

CloudBees Docker Build and Publish

! Téléchargé avec succès. Sera activé lors du prochain démarrage

→ [Revenir en haut de la page](#)

(vous pouvez commencer à utiliser les plugins installés dès maintenant)

→ ☐ Redémarrer Jenkins quand l'installation est terminée et qu'aucun job n'est en cours

Progression du téléchargement

Préparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Authentication Tokens API

⚠ Téléchargé avec succès. Sera activé lors du prochain démarrage

Docker Commons

⚠ Téléchargé avec succès. Sera activé lors du prochain démarrage

SSH server

⚠ Téléchargé avec succès. Sera activé lors du prochain démarrage

CloudBees Docker Build and Publish

⚠ Téléchargé avec succès. Sera activé lors du prochain démarrage

Redémarrer Jenkins

⋮ En cours

→ [Revenir en haut de la page](#)

(vous pouvez commencer à utiliser les plugins installés dès maintenant)

→ ☒ Redémarrer Jenkins quand l'installation est terminée et qu'aucun job n'est en cours

Plugins

↓ Mises à jour

📦 Plugins disponibles

⚙️ Plugins installés

⚙️ Paramètres avancés

🔍 docker

Nom ↓

[CloudBees Docker Build and Publish plugin](#) 1.4.0

This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker
[Report an issue with this plugin](#)

[Docker Commons Plugin](#) 439.va_3cb_0a_6a_fb_29

Provides the common shared functionality for various Docker-related plugins.
[Report an issue with this plugin](#)