

docker

Formation Docker - Examen

Sommaire

1 - Configuration de GitHub..... 3

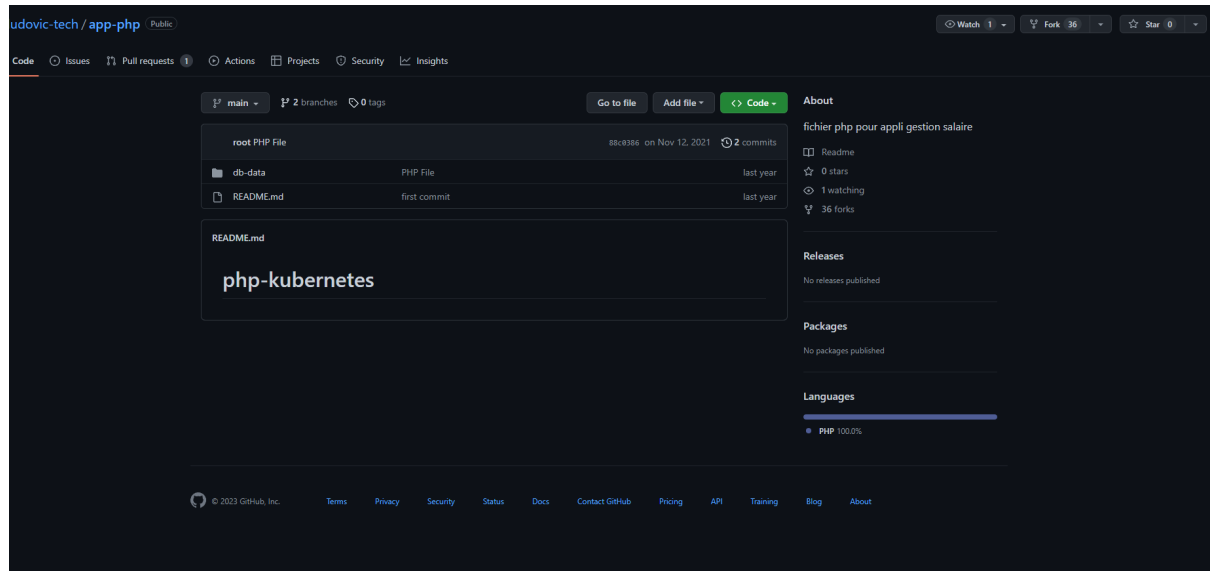
2 - Mise en place de l'environnement de travail.....5

2 - Configuration de Jenkins..... 9

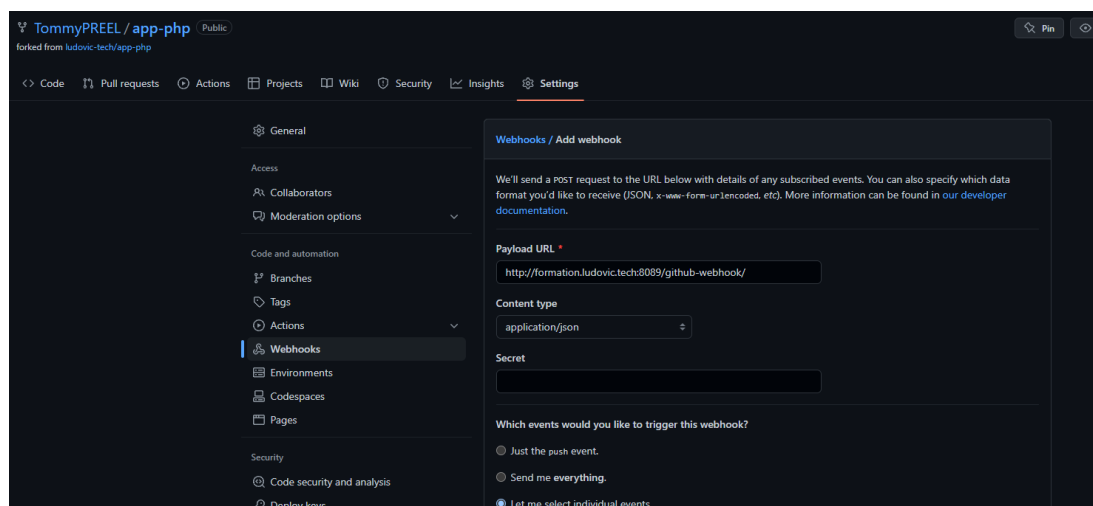
1 - Configuration de GitHub

1 - Ouvrir le github et cliquer sur “Fork” (en haut à droite).

Lien: <https://github.com/ludovic-tech/app-php>



2 - Création de webhooks



Payload URL : URL de l'application Jenkins, puis ajouter /github-webhook.

Content type: application/json.

Sélectionner “Let me select individual events”.

<input checked="" type="checkbox"/> Pull requests Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestoned, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.	<input checked="" type="checkbox"/> Pushes Git push to a repository.
<input type="checkbox"/> Registry packages Registry package published or updated in a repository.	<input type="checkbox"/> Releases Release created, edited, published, unpublished, or deleted.
<input type="checkbox"/> Repositories Repository created, deleted, archived, unarchived, publicized, privatized, edited, renamed, or transferred.	<input type="checkbox"/> Repository imports Repository import succeeded, failed, or cancelled.
<input type="checkbox"/> Repository vulnerability alerts Dependabot alert (aka dependency vulnerability alert) created, resolved, or dismissed on a repository.	<input type="checkbox"/> Secret scanning alert locations Secrets scanning alert location created
<input type="checkbox"/> Secret scanning alerts Secrets scanning alert created, resolved, or reopened	<input type="checkbox"/> Security and analyses Code security and analysis features enabled or disabled for a repository.
<input type="checkbox"/> Stars A star is created or deleted from a repository.	<input type="checkbox"/> Statuses Commit status updated from the API.
<input type="checkbox"/> Team adds Team added or modified on a repository.	<input type="checkbox"/> Visibility changes Repository changes from private to public.
<input type="checkbox"/> Watches User stars a repository.	<input type="checkbox"/> Wiki Wiki page updated.
<input type="checkbox"/> Workflow jobs Workflow job queued, waiting, in progress, or completed on a repository.	<input type="checkbox"/> Workflow runs Workflow run requested or completed on a repository.

☒ **Active**
 We will deliver event details when this hook is triggered.

Cocher Pull requests, pushes et Active (en bas)

1 - Mise en place de l'environnement de travail

1 - Se déplacer dans le répertoire de Jenkins

```
[root@tommy ~]# cd jenkins
```

2 - Modifier le docker-compose.yml de Jenkins

```
[root@tommy jenkins]# vim docker-compose.yml
```

```
version: '2.4'

services:
  jenkinsci:
    image: jenkinsci/blueocean:1.25.5
    ports:
      - 8080:8080
      - 50000:50000
    volumes:
      - ./jenkins_home:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
    command: sh -c "apk update && apk add docker-compose && /sbin/tini -- /usr/local/bin/jenkins.sh"
    user: root
    restart: on-failure
```

La ligne commande permet d'installer docker-compose, utile pour la suite.

2 - Lancement du docker-compose.yml de Jenkins

```
[root@tommy jenkins]# docker-compose up -d
[+] Running 15/15
# jenkinsci Pulled
# 2408cc74d12b Pull complete
# bf6bdadd7450 Pull complete
# 9a122f85f3c9 Pull complete
# aea48bee21ff Pull complete
# e2cfb1579085 Pull complete
# 6ed48a3271c3 Pull complete
# e4f28e1b9f4e Pull complete
# 78f4f1e35669 Pull complete
# 3968a4384d5c Pull complete
# 034428d37519 Pull complete
# f7f0bf5c5bf2 Pull complete
# 2ef560e709b6 Pull complete
# 23bd89992269 Pull complete
# 0b0c9ca38c9e Pull complete
[+] Running 2/2
# Network jenkins_default Created
# Container jenkins-jenkinsci-1 Started
```

1 - Créer un répertoire de projet

```
[root@tommy ~]# mkdir docker-project
```

2 - Se déplacer dans le dossier créé

```
[root@tommy ~]# cd docker-project/
```

3- Créer un fichier docker-compose.yml

```
[root@tommy docker-project]# vim docker-compose.yml
```

4 - Création du répertoire php-data

```
[root@tommy docker-project]# mkdir php-data
```

5 - Création du répertoire db-data

```
[root@tommy docker-project]# mkdir db-data
```

6 - Configurer le docker-compose.yml

```
version: '3.9'
services:
  web:
    image: tutum/apache-php
    hostname: web-php
    ports:
      - '9000:80'
    volumes:
      - ./php-data/app-php/db-data:/var/www/html/
    networks:
      - frontend
      - backend
  db:
    image: mariadb
    hostname: mysql
    volumes:
      - ./db-data:/var/lib/mysql
    environment:
      - MARIADB_ROOT_PASSWORD=root
    networks:
      - backend
networks:
  frontend:
  backend:
```

7 - Création de Jenkinsfile

```
[root@tommy docker-project-php]# vim Jenkinsfile
```

8 - Édition du Jenkinsfile

```
pipeline {  
  agent any  
  stages {  
    stage('Docker Compose Up') {  
      steps {  
        sh 'docker-compose up -d'  
      }  
    }  
  }  
}
```

9 - Installer git

```
(reverse-i-search)`inst': yum install git -y
```

10 - Connexion à git

Tapez ces 2 commandes:

```
Run  
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

11 - Se déplacer dans php-data

```
[root@tommy docker-project]# cd php-data/
```

12 - Cloner le dossier de travail en ligne

```
[root@tommy docker-project-php]# git clone https://github.com/TommyPREEL/app-php
```

NB : mettre le lien du fork

13 - Utilisation du dossier courant

```
[root@tommy docker-project-php]# git add .
```

14 - Commit

```
[root@tommy docker-project-php]# git commit -m "update"
```

15 - Push sur le dossier GitHub

```
[root@tommy docker-project-php]# git push
```

15 - Monter le docker-compose en mode détaché

```
[root@tommy docker-project]# docker-compose up -d
[+] Running 21/21
# db Pulled
# 10ac4908093d Pull complete
# 44779101e748 Pull complete
# a721db3e3f3d Pull complete
# 1850a929b84a Pull complete
# 397a918c7da3 Pull complete
# 806be17e856d Pull complete
# 634de6c90876 Pull complete
# cd00854cfb1a Pull complete
# web Pulled
# 6ffe5d2d6a97 Pull complete
# f4e00f994fd4 Pull complete
# e99f3d1fc87b Pull complete
# a3ed95caeb02 Pull complete
# c10ab9e629d8 Pull complete
# 5c65b067601c Pull complete
# 2cbc3ddde9c1 Pull complete
# 9a5e857e989f Pull complete
# 7bd95ceec22a Pull complete
# c27094051f1e Pull complete
# d820ec68d287 Pull complete
[+] Running 4/4
# Network docker-project_backend Created
# Network docker-project_frontend Created
# Container docker-project-db-1 Started
# Container docker-project-web-1 Started
```

16 - On vérifie que le serveur fonctionne

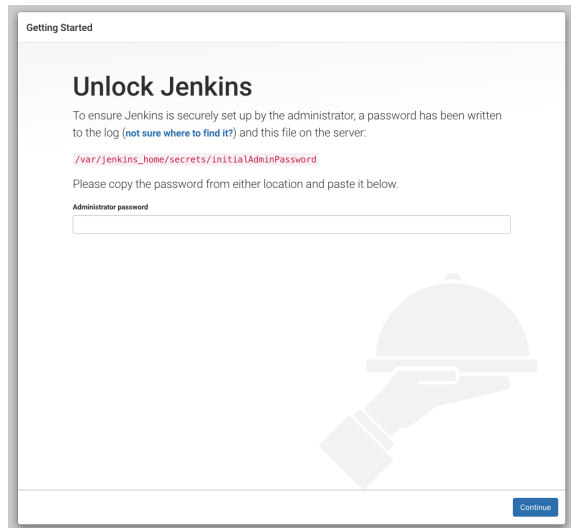
```
[root@tommy docker-project]# curl localhost:9000/create_db.php
```

Résultat:

```
Connected successfully
Database test_db created successfully
```


2 - Configuration de Jenkins

1 - Se connecter au Jenkins de votre machine virtuelle



Entrez le mot de passe, cliquez sur “continue” puis “installer tous les plugins”.

Démarrage

Installation Failures

Some plugins failed to install properly, you may retry installing them or continue without the failed plugins

Réessayer

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding Plugin
✗ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source Plugin	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git plugin	✓ SSH Build Agents	✓ Matrix Authorization Strategy Plugin	✓ PAM Authentication
✗ LDAP	✗ Email Extension	✓ Mailer Plugin	

Jenkins 2.346.1

Continue Réessayer

Cliquez sur “Continue” en bas à droite.

2 - Créer un utilisateur

Démarrage

Créer le 1er utilisateur Administrateur

Nom d'utilisateur:

Mot de passe:

Confirmation du mot de passe:

Nom complet:

Adresse courriel:

Jenkins 2.346.1

[Continuer en tant qu'Administrateur](#)

[Sauver et continuer](#)

Cliquer sur Sauver et continuer après avoir tout entré.

3 - Entrer l'url de jenkins (ici, laissée par défaut).

Démarrage

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 2.346.1

[Passer cette étape et terminer](#)

[Sauver et terminer](#)

Cliquer sur “Sauver et terminer”.

4 - Se connecter

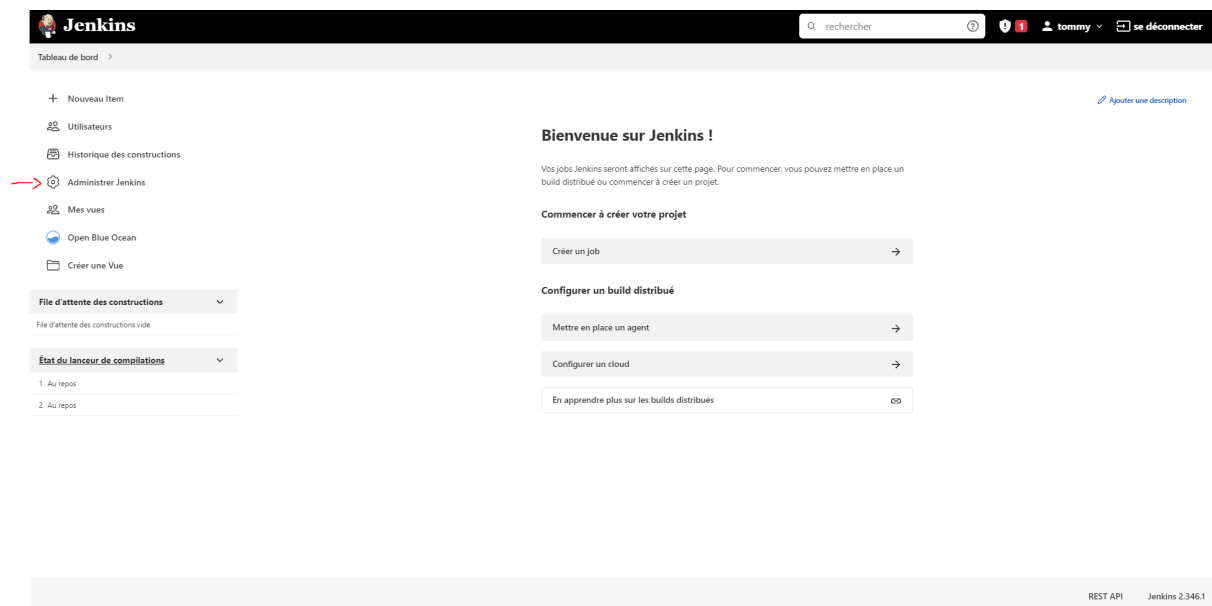


Bienvenue dans Jenkins !

☐ Garder ma session ouverte

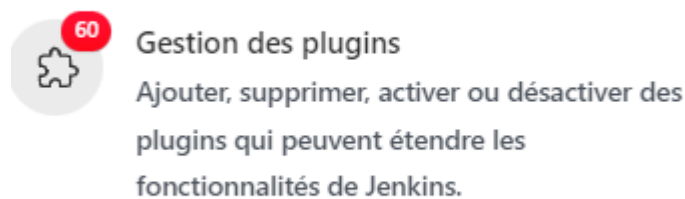
S'identifier

5 - Cliquer sur “Administrer Jenkins”



The screenshot shows the Jenkins dashboard interface. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains a list of navigation items: 'Nouveau Item', 'Utilisateurs', 'Historique des constructions', 'Administrer Jenkins' (highlighted with a red arrow), 'Mes vues', 'Open Blue Ocean', and 'Créer une Vue'. Below these are sections for 'File d'attente des constructions' and 'État du lanceur de compilations'. The main content area displays a welcome message and options to create a new job or configure a distributed build. The bottom status bar shows 'REST API' and 'Jenkins 2.346.1'.

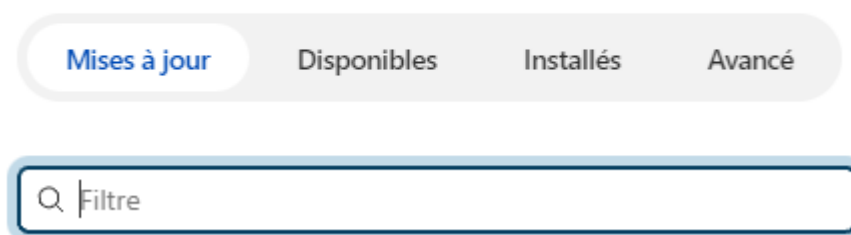
6 - Cliquer sur “Gestion des plugins”



A card titled 'Gestion des plugins' with a red badge containing the number '60'. The card features a gear icon and the text: 'Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.'

7 - Cliquer sur “Disponibles”

Plugin Manager



The image shows the 'Plugin Manager' interface. It features four tabs: 'Mises à jour' (highlighted in blue), 'Disponibles', 'Installés', and 'Avancé'. Below the tabs is a search bar with a magnifying glass icon and the text 'Filtre'.

8 - Installation des plugins

Taper “docker” dans la barre de recherche, sélectionner “Docker”, “Docker Commons”, “Docker Pipeline”.

Plugin Manager

Mises à jour Disponibles Installés Avancé

Install

Name ↓

Docker 1.3.0

Cloud Providers Cluster Management docker

☒ This plugin integrates Jenkins with Docker

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

Docker Commons 1.21

☒ Library plugins (for use by other plugins) docker

Provides the common shared functionality for various Docker-related plugins.

Docker Pipeline 563.vd5d2e5c4007f

pipeline DevOps Deployment docker

☒ Build and use Docker containers from pipelines.

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

Cliquez sur “Download now and install after restart”.

Download now and install after restart

Attendre l'installation, vous pouvez cocher la case "Redémarrer..." pour relancer Jenkins après l'installation des plugins.

Installation/Mise à jour des Plugins

Préparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Docker Commons

⚠ Downloaded Successfully. Will be activated during the next boot

Durable Task

⚠ Downloaded Successfully. Will be activated during the next boot

bouncycastle API

⚠ Downloaded Successfully. Will be activated during the next boot

Docker API

⚠ Downloaded Successfully. Will be activated during the next boot

Pipeline: Supporting APIs

⚠ Downloaded Successfully. Will be activated during the next boot

Docker

⚠ Downloaded Successfully. Will be activated during the next boot

Pipeline: Nodes and Processes

⚠ Downloaded Successfully. Will be activated during the next boot

Pipeline: Model API

⋮ En cours

Pipeline: Declarative Extension Points API

⋮ En cours

Pipeline: Stage Tags Metadata

⋮ En cours

Pipeline: Declarative

⋮ En cours

Pipeline: Basic Steps

⋮ En cours

Docker Pipeline

⋮ En cours

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

9 - Retourner sur l'interface de Jenkins et créer un item




+ Nouveau Item

10 - Entrer un nom, sélectionner “Pipeline” puis valider


Saisissez un nom

PipelineBuild

» Champ obligatoire



Construire un projet free-style
Ceci est la fonction principale de Jenkins qui sert à build (construire) votre projet. Vous pouvez intégrer tous les outils de gestion de version avec tous les systèmes de build. Il est même possible d'utiliser Jenkins pour tout autre chose qu'un build logiciel.



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.

11 - Configuration

Dans Build Triggers, cocher GitHub hook trigger

Build Triggers

☐ Construire après le build sur d'autres projets ?

☐ Construire périodiquement ?

☒ GitHub hook trigger for GITScm polling ?

Dans Pipeline:

Pipeline script from SCM.


SCM, choisir git.

Repository URL: mettre le lien du git.

12 - Lancer un build

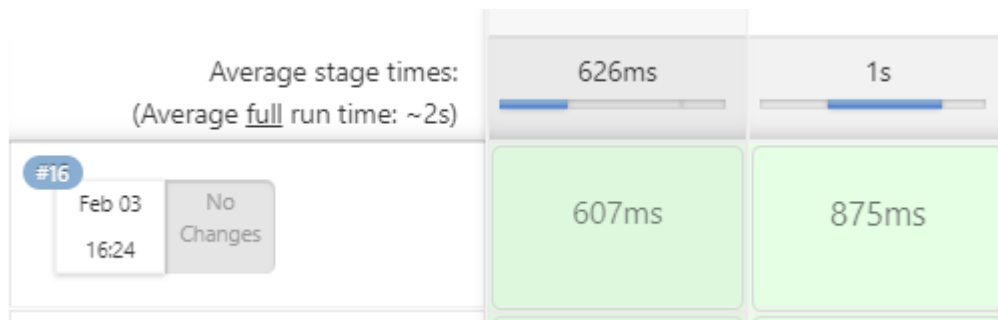
↑ Back to Dashboard

🔍 Status

 Changes

▶ Lancer un build

13 - Résultat:



Succès !