

Docker

Examen



# Sommaire:

## I. Enoncé

## II. Prérequis

### I. Ajouter un utilisateur sur votre serveur

### II. Github

#### 1. Initialisation du projet

#### 2. Installer git sur le serveur

#### 3. Ajouter le projet sur le serveur

## III. Docker

### 1. Générer votre docker-compose

### 2. Exécution du docker compose

## IV. Jenkins

### I. Installation jenkins

### II. initialisation Jenkins

#### 1. Ajouter des plugins

#### 2. Ajouter votre compte Docker à Jenkins

#### 3. Fichier Jenkinsfile

## III. Jenkins destock

### I. Ajouter un item

### II. Tester l'item

# I. Enoncé

Vous êtes engagés en qualité d'ingénieur DevOps. Par la société OnFaitDuBonBoulot.com

Vous avez en charge de mettre en œuvre d'un Pipeline DevOps, Docker, Jenkins, Github afin de déployer une application de gestion de paye.

Vous trouverez a URL suivante, le logiciel développé par le client en php.  
<https://github.com/ludovic-tech/app-php.git>

Vous devez fournir un fichier docker compose, afin d'exécuter l'application. Les données de l'application devront être persistentes. L'application est développée en PHP.

Vous devez fournir un Pipeline Jenkins, github afin de pouvoir mettre à jour les images Docker. Les images mise à jour devront être disponibles via le Hub Docker. Ce Pipeline doit pouvoir mettre à jour votre docker-compose. (exécution du compose via Jenkins, après mise à jour de l'image via dockerfile). Vous devez fournir au client un compte rendu de votre intervention. Les fichiers docker, jenkins etc et la documentation au format PDF, doivent être accessibles via un repo github.

## II. Prérequis

### I. Ajouter un utilisateur sur votre serveur

Ajouter votre utilisateur au groupe docker via cette commande:

```
useradd -g docker Login_de_votre_choix
```

Ensuite modifier votre mot de passe par celui de votre choix

```
passwd Login_de_votre_choix
```

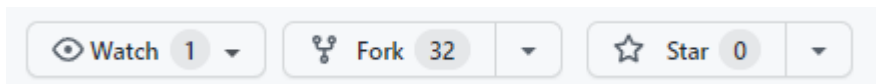
Connecter vous avec l'utilisateur que vous venez de créer

```
su Login_de_votre_choix
```

## II. Github

### 1. Initialisation du projet

Avoir un compte Github et se rendre sur ce projet <https://github.com/ludovic-tech/app-php> et utiliser "Fork".



Pour récupérer le projet sur votre Github.

### 2. Installer git sur le serveur

Installer Git avec l'utilisateur root

```
yum install git
```

### 3. Ajouter le projet sur le serveur

Ajouter le projet sur le serveur avec cette commande

```
git clone https://github.com/NyckolasP/app-php.git
```

## III. Docker

### 1.Générer votre docker-compose

Après avoir récupérer votre projet on créer un “**docker-compose.yml**” avec 2 services dedans :

- Le premier qu'on appellera **web** qui correspond à une image **apache**, qu'on dispose sur le port 200
- Le second service s'appellera **mysql** qui correspond à une image **mariadb**, on défini également le mdp pour ce connecter en base de données.

### 2.Exécution du docker compose

Pour monter votre docker-compose

```
docker-compose up -d
```

## IV.Jenkins

### I.Installation jenkins

Créer un dossier jenkins avec la commande

```
mkdir jenkins
```

Dans le dossier jenkins créer un fichier “**docker-compose.yml**” qui contiendra le code en dessous.

```
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
    user: root
    restart: on-failure
```

Puis lancer le fichier **docker-compose.yaml** avec cette commande qui la le construire.

```
docker-compose up -d
```

Quand vous ouvrez sur votre navigateur qui gère Jenkins vous serez sur cette page.

Démarrage

## 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

Il faut utiliser la commande en dessous pour récupérer notre code

```
docker-compose logs
```

- Une fois le code entré choisir l'option "**Installation des plugins suggérer**".
- Après avoir vu les installations se terminer, cliquer sur "**Continuer**".
- Créer votre premier utilisateur.
- Après avoir créé votre utilisateur cliquer sur "**Sauver et terminer**".
- Pour finir redémarrer Jenkins

Voilà jenkins est installer 👍

## II. initialisation Jenkins

### 1.Ajouter des plugins

- Une fois connecté sur Jenkins rendez-vous dans "**Administrer Jenkins**".
- Ensuite allez sur "**Gestion des plugins**".

Cliquer sur "**Disponibles**" et rechercher ces trois plugins :

- Docker
- Docker Commons
- Docker Pipeline

Ensuite cliquer sur "**Download and install after restart**" après avoir cliqué ne pas oublier

de cocher la checkbox pour redémarrer Jenkins.

## 2. Ajouter votre compte Docker à Jenkins

Aller de nouveau sur “**Administrer Jenkins**” et trouver “**Manage credentials**”.

Ensuite cliquer sur le nom “**jenkins**” en bleu.

Par la suite, faites glisser votre souris sur “**Identifiants globaux (illimité)**” qui ouvrira un menu dropdown, cliquer sur la petite flèche une petite case apparaîtra “**add credentials**” cliquer dessus.

Pour les champs dans le formulaire :

- Nom d'utilisateur : Qui correspond à votre ID Docker
- Mot de passe : Qui correspond à votre MDP Docker
- ID : Qui correspond à un tag qui sera utilisé pour identifier notre compte on le nommera “**id\_docker\_1**”

## 3. Fichier Jenkinsfile

Dans votre répertoire git ajouter un fichier **JenkinsFile** qui dedans aura un e suite de procédure pour exécuter des tests

## 4. Webhook

Se rendre sur votre projet git; se rendre dans “**Settings**”, puis cliquer sur “**webhooks**” dans la colonne de gauche.

Ajouter le webhook avec en configuration :

- Payload URL : <http://formation.ludovic.tech:8180/github-webhook/>

Qui correspond à l'adresse de notre Jenkins

- Content type : application/json

Et cocher le bouton radio “**Just the push event**” et la checkbox Active

# III. Jenkins destock

## I. Ajouter un item

Cliquer sur “**nouveau item**” donner le nom que voulez et choisir “**pipeline**”

Dans **Build Triggers** :

Cocher la case : **GitHub hook trigger for GITScm polling**

Descendre jusqu'à “**pipeline**” :

- Définition : Pipeline script from SCM
- SCM : Git
- Repository URL : <https://github.com/NyckolasP/app-php.git>
- Branch Specifier : \*/main

Puis sauvegarder.

## II. Tester l’item

Cliquer sur “**lancer le build**” puis observer les résultats. Si les fichiers sur le git sont modifier alors le build se lancera automatiquement.