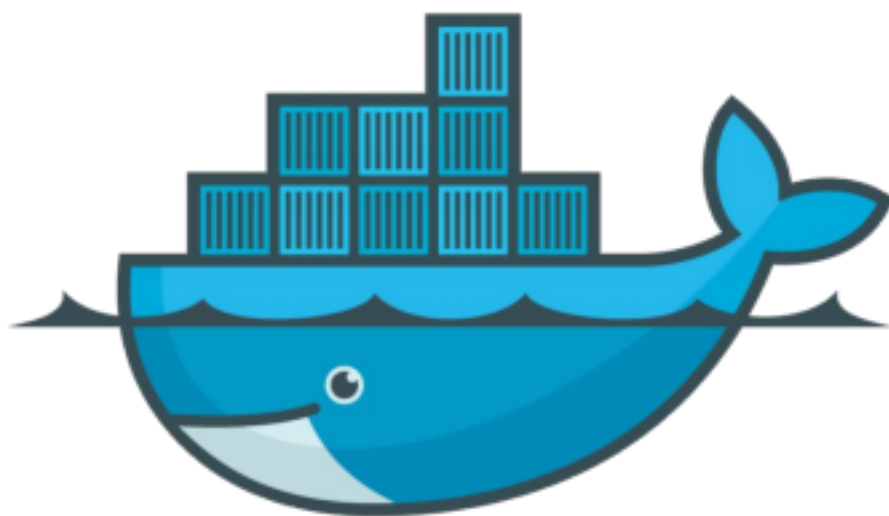


# Formation



docker

## Formation Docker

<b>Examen Formation Docker</b>	<b>3</b>
Énoncé	3
Initialisation Jenkins	4
Initialisation Git	5
Création pipeline	6

# Examen Formation Docker

## Énoncé

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 paie.

Vous trouverez à l'URL suivant, 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 mises 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.

## Initialisation Jenkins

1. Créez dans le dossier jenkins

```
mkdir jenkins
```

2. Déplacez-vous dans le dossier jenkins

```
cd jenkins
```

3. Créez et éditez le fichier docker-compose

```
vim docker-compose.yml
```

4. Ajoutez les instructions suivantes :

```
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
```

5. Démarrez l'application en arrière-plan :

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

6. Ouvrez dans le navigateur l'adresse suivante

<http://formation.ludovic.tech:8088/>

7. Débloquez Jenkins en récupérant le mot de passe trouvable dans les logs

```
docker-compose logs
```

8. Installez les plugins suggérés
9. Créez un premier utilisateur et connectez-vous
10. Installez les plugins :

- a. Rendez-vous dans le L'onglet "Administrer Jenkins"
  - b. Cliquez sur "Gestion des plugins"
11. Recherchez "docker" dans les plugins disponibles
  - a. Sélectionnez
    - i. "Docker 1.3.0"
    - ii. "Docker commons"
    - iii. "Docker Pipeline"
  - b. Cliquez sur "Download now and install after restart"
  - c. Une fois lancé cochez la case "Relancer Jenkins"

## Initialisation Git

12. Passez en administrateur

```
su -
```

13. Installez git

```
yum install git
```

14. créer un nouveau répertoire

```
mkdir projet
```

15. Vous rendre dans le répertoire créé

```
cd projet
```

16. clonez répertoire

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

17. Créez et éditez un fichier docker-compose.yml

```
vim docker-compose.yml
```

18. Ajoutez les instructions suivantes :

```
version: '3.9'
services:
  web:
    image: tutum/apache-php
    hostname: web-php
    ports:
```

```
- '9000:80'
volumes:
- ./app-php/db-data:/var/www/html/
networks:
- frontend
- backend

db:
image: mariadb
hostname: db
volumes:
- ./db-data:/var/lib/mysql
environment:
- MARIADB_ROOT_PASSWORD=root
networks:
- backend
networks:
frontend:
backend:
```

19. se connecter à docker

```
docker login
```

## Création pipeline

Sur le tableau de bord Jenkins

20. Créez un nouvel item en cliquant sur “+ Nouvel item”
  - a. Saisissez un nom de pipeline
  - b. sélectionner “Pipeline”
  - c. Cliquez sur “OK”
  - d. Rentrez les informations du git
  - e. Validé