



Etudes de cas : 30 mn

Vous êtes spécialisé dans le développement d'application Web et vous avez d'industrialiser vos développements.

Vous décidez de mettre en place l'automatisation à partir d'une application Web pilote.

1. Décrire le workflow DevOps **de haut niveau** pour déployer l'application Web à l'aide de pratiques CI/CD.
2. Reportez-vous à la partie **Mise en œuvre** et réaliser la chaîne de déploiement continue.

Rendu

- Le rendu du TP est à effectuer par groupe.
- Pour chaque groupe les éléments suivant devront être présentés lors de la présentation finale du cursus :
 - Une présentation décrivant les différents éléments de l'infrastructure et leurs objectifs ainsi que les choix réalisés lors de la réalisation
 - On peut se servir de diapositives afin d'avoir un support oral. L'idée est de voir la gestion du temps, l'expression orale et évidemment le côté technique. Et attention, à la répartition de parole dans le groupe, chacun doit occuper sa place.
 - La qualité des diapositives est notée également.
 - La présentation dure 10 mn, 5 mn de plus de questions
 - Pas de rapport écrit à part les diapositives



Partie 1 : workflow DevOps

Nouvelles fonctionnalités
ou correctifs

Environnement de
développement
Appi Web/Nginx
Git/Docker

Pull du code
source

Push des modifications
dans la forge

Forge logicielle
Repository github

Détection d'un
changement de la forge
et récupération du code
source

Pipeline CI/CD
Jenkins

- 1 Vérification qualités du code (Linting)
- 2 Construction de l'application
- 3 Exécution des tests (unitaire, intégration, fonctionnelle)
- 4 Création de la release

Livraison de la
release (Artefact)

Récupération de la
release à déployer

Espace de livraison
Registre image
Web/Nginx
Dockerhub



Déploiement de la
release en production

Environnement de
production



Partie 2 : mise en œuvre

Créer un dépôt GitHub



Rendez-vous sur <https://github.com> et créer votre compte et un repository. Par exemple : **monTP**

Mettre en place le projet sous GIT



❑ Installer git (voir <https://git-scm.com/downloads>)

❑ Configurer votre environnement git :

```
git config --global user.email "<votre email>"
git config --global core.editor vim
git config --global core.autocrlf input
git config --global credential.helper store
```

❑ Créer un répertoire de travail :

```
cd && mkdir monTP && cd monTP
```

❑ Récupérer le projet jenkins du formateur :

```
git clone https://github.com/fpicot31/jenkinsTP.git
cd jenkins
mv ./jenkins/* .
rm -rf ./jenkins
rm -rf .git
```

❑ Synchroniser le répertoire de travail avec la forge github :

```
git init
git add .
git commit -m «initialisation»
git remote add origin https://github.com/<votre login>/monTP.git
git push -u origin master
```

Tester le serveur WEB en local

- ❑ Construire l'image de l'application Web dans l'espace de travail **jenkinsTP** :

```
docker build -t monServeurWeb .
```

- ❑ Lancer le serveur Web :

```
docker run -d --name serveurweb -p 8081:80 monServeurWeb
```

- ❑ Ouvrir un navigateur sur le port 8081 en local :

http://localhost:8081



Welcome to Bienvenue a la formation DevOps !!!!!

If you see this page, the Bienvenue a la formation DevOps !!!!! web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [Bienvenue a la formation DevOps !!!!!.org](#).

Commercial support is available at [Bienvenue a la formation DevOps !!!!!.com](#).

Thank you for using Bienvenue a la formation DevOps !!!!!.

Mettre en place l'espace de livraison



❑ Rendez-vous sur <https://hub.docker.com> et créer votre compte.

❑ Créer un repository. Par exemple **monAppliWeb**

Démarrer les environnements (jenkins et serveur de production)

- ❑ Installer Virtualbox : <https://www.virtualbox.org/wiki/Downloads>
- ❑ Installer Vagrant : <https://developer.hashicorp.com/vagrant/docs/installation>
- ❑ Récupérer le projet vagrant du formateur :
git clone <https://github.com/fpicot31/vagrantTP.git>
- ❑ Se placer dans le répertoire **vagrantTP** et démarrer les environnements (**hostname = jenkins IP=192.168.5.10** et **hostname=srvprod IP=192.168.5.20**) :
Voir le fichier ReadMe

Création du pipeline (Jenkinsfile)



❑ Mettre à jour le fichier jenkinsfile du répertoire **jenkinsTP/jenkins** :

```
registry = « <votre dockerhub login>/<votre dockerhub repository> »
```

❑ Pousser le pipeline sur la forge :

```
git add .
```

```
git commit -m «creation du pipeline»
```

```
git push
```

Configuration du pipeline Jenkins

❑ Positionner les propriétés :

Scrutation de l'outil de gestion de version avec un planning : * * * * *

Pipeline script from SCM

SCM : Git

Repositories URL : URL <https://github.com/<Votre login github>/monTP.git>

Credentials : github credential

Branches to build : */master

Script Path : Jenkinsfile



Déclencher le pipeline



- ☐ Modifier le message de la page d'accueil du serveur WEB :

Dans le fichier Dockerfile du projet jenkins/application:

```
RUN sed -i 's/nginx/<votre message>/g' /usr/share/nginx/html/index.html
```

- ☐ Pousser la modification sur la forge :

```
git add .
```

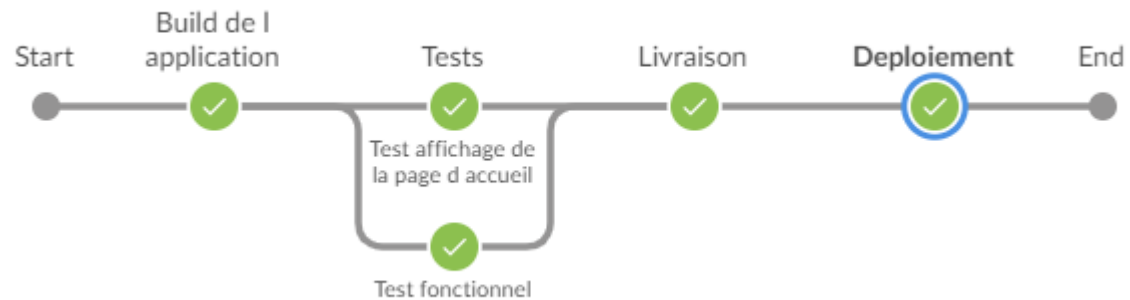
```
git commit -m "modification de l'application«
```

```
git push
```

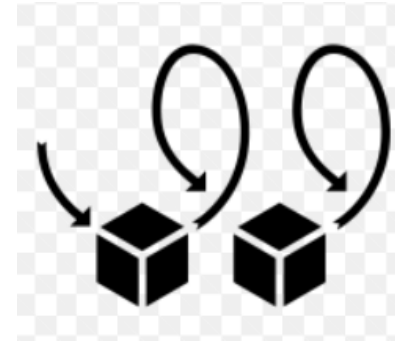
Visualisation de l'exécution du pipeline Jenkins



- ☐ Sélectionner **Mes Vues** à partir du **Tableau de bord**
- ☐ Sélectionner le **Nom du pipeline**
- ☐ Sélectionner le dernier job lancé dans l'**historique des builds** (numéroté avec #<numéro>)
- ☐ **Open Blue Ocean**



Verification de la livraison



- ❑ Ouvrir un navigateur et accéder au serveur Web de production :

`http://192.168.5.20:80801`

Welcome to BRAVO Ã tous !!!!!

If you see this page, the BRAVO Ã tous !!!!! web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to BRAVO Ã tous !!!!!.org.
Commercial support is available at BRAVO Ã tous !!!!!.com.

Thank you for using BRAVO Ã tous !!!!!.

Synthèse workflow CI/CD

