

# Devoir rendu

*Devoir fait par Steven Sivakaran et Loris Paolinelli*

## Jenkins Job 1

Lien du repo git avec les fichiers <https://github.com/Steven-176/JenkinsP>

Les scripts sont disponibles dans le dépôt GitHub.

Pour commencer le job, nous avons d'abord procédé étape par étape pour détecter les changements sur un dépôt GitHub.

Nous avons mis en place les paramètres suivants pour détecter les changements :

- Dans la gestion du code, nous avons coché Git et mis le lien URL de notre dépôt avec la branche spécifiée qui était *master* pour nous.
- Ensuite, nous avons mis en place une scrutation de l'outil de gestion de version avec un crontab pour vérifier la version.

Pour démarrer un nouveau conteneur "test", il suffit de connaître les commandes lxc et d'utiliser la commande suivante :

```
lxc init ubuntu:22.04 test
```

Pour installer un serveur web avec Apache et PHP dans le conteneur "test", il suffit d'exécuter la commande suivante :

```
lxc exec test bash -- apt update
lxc exec test bash -- apt install -y apache2
lxc exec test bash -- apt install -y php
```

Pour cloner le dépôt dans le conteneur (/var/www/html), utilisez la commande suivante :

```
lxc exec $NAME -- git clone '[<https://github.com/Steven-176/JenkinsP.git>](<https://github.com/Steven-176/JenkinsP.git>)' /var/www/html/
```

Procédez à un ensemble de tests :

- Un dossier de test doit être présent sur le dépôt.
- Chaque fichier php de test dans ce dossier doit faire l'objet d'un curl.
- Chaque fichier php de test doit retourner "true" ou "false" en fonction de si son test a fonctionné ou non.

Si tous les tests sont à "true" :

- Éteindre le conteneur "prod".
- Renommer le conteneur "prod" en "rollback".
- Renommer le conteneur "test" en "prod".
- Supprimer le dossier de tests unitaires du nouveau conteneur.

Si au moins un des tests est à "false" :

- Éteindre le conteneur "test".
- Supprimer le conteneur "test".
- Faire passer le job en erreur.

Voici le script Bash que j'ai mit dans Build Step:

```
#!/bin/bash

NAME='test'

#création du conteneur
lxc init ubuntu:22.04 $NAME
sleep 2

#attachement du conteneur à la carte réseau
lxc network attach lxdbr0 $NAME
sleep 2

#démarrage du container
lxc start $NAME
sleep 2
#configuration du réseau
lxc exec $NAME -- sed -i 's|#DNS=|DNS=1.1.1.1|g' /etc/systemd/resolved.conf
lxc exec $NAME -- systemctl restart systemd-resolved
lxc exec $NAME -- bash -c 'echo -e "[Match]\nName=*\n\n[Network]\nDHCP=ipv4" > /etc/systemd/network/10-all.network'
lxc exec $NAME -- systemctl restart systemd-networkd
sleep 2

#installation des packages
lxc exec $NAME apt update
lxc exec $NAME -- apt install apache2 -y
lxc exec $NAME -- apt install php -y
lxc exec $NAME -- apt install git -y

#suppression de index.html
lxc exec $NAME rm /var/www/html/index.html

#récupération du répertoire git
lxc exec $NAME -- git clone 'https://github.com/Steven-176/JenkinsP.git' /var/www/html/
sleep 2

#récupération de l'adresse ip du conteneur test
CT_IP=$(lxc ls test -f csv -c 4 | cut -d ' ' -f 1)
tab=()
allValid=true

for A in {1..4}
do
    #récupération du résultat de la page
    RESULT=$(curl $CT_IP/test/test$A.php)
    sleep 2

    echo "$RESULT"

    #vérification si le contenu de la page renvoie "oui"
    if [ "$RESULT" == "oui" ]
    then
        echo "true"
        tab+=("yes")
    else
        echo "false"
        tab+=("no")
    fi
done

for i in "${tab[@]}"
do
    if [ "$i" -eq "no" ]
```

```

    then
        allValid=false
        break
    fi
done

if $allValid
then
    ### Si les tests sont tous a true ###
    #récupération du conteneur de prod
    lxc list -f csv -c n | grep -x prod

    #vérification si un conteneur prod existe
    if [ $? -eq 0 ]
    then

        #arrêt du conteneur de prod
        lxc stop prod
        sleep 2

        #renommage du conteneur de prod en conteneur de rollback
        lxc rename prod rollback
        sleep 2
    fi
    #arrêt du conteneur de test
    lxc stop $NAME
    sleep 2

    #renommage du conteneur de test en conteneur de prod
    lxc rename $NAME prod
    sleep 2

    #démarrage du nouveau conteneur de prod
    lxc start prod
    sleep 2
else
    ### Si au moins un des tests est a false ###
    #arrêt du conteneur test
    lxc stop test
    sleep 2

    #suppression du conteneur de test
    lxc rm test

    exit 1
fi

```

## Pour le Job 2 :

### Job 2 - Rollback

L'objectif est de créer un job capable de remettre en activité la dernière version de l'application manuellement. Aucun test ne sera lancé, il s'agit simplement de réactiver l'ancien conteneur renommé "rollback" lors des étapes de déploiement continu.

Le job doit :

- Éteindre le conteneur "prod" : pour cela, exécutez la commande suivante :  
lxc stop prod
- Supprimer le conteneur "prod" :  
lxc rm prod
- Renommer le conteneur "rollback" en "prod" :

```
lxc rename rollback prod
```

- Allumer le conteneur "prod" :

```
lxc start prod
```

```
#!/bin/bash

### Eteins le conteneur prod ###
lxc stop prod

### Supprime le conteneur prod ###
lxc delete prod

### Renomme le conteneur de rollback en conteneur de prod ###
lxc rename rollback prod

### Allume le conteneur prod ###
lxc start prod
```

## Job 3 : Forcer la dernière version

L'objectif est de créer un job qui installe un conteneur avec la dernière version du code source sans validation de celui-ci.

Le job doit :

- Démarrer un conteneur "pre-prod"
- Installer un serveur web avec apache et php dans le conteneur "pre-prod"
- Cloner le repository dans le conteneur (/var/www/html)
- Supprimer le dossier de tests unitaires du nouveau conteneur
  - Nous pensons qu'il faut exécuter la commande suivante : `lxc exec pre-prod /bin/bash rm /var/www/html/test`
- Éteindre le conteneur "prod" : `lxc stop prod`
- Supprimer le conteneur "prod" : `lxc rm prod`
- Renommer le conteneur "pre-prod" en "prod" : `lxc rename pre-prod prod`

Ci-dessous se trouve un script de build qui n'a pas fonctionné par manque de temps :

```
$NAME='pre-prod'

### Crée un conteneur pre-prod ###
lxc init ubuntu:22.04 $NAME

# Attachement du conteneur à la carte réseau
lxc network attach lxdbr0 $NAME
sleep 2

# Démarre le conteneur
lxc start $NAME
sleep 2

# Configuration du réseau
lxc exec $NAME -- sed -i 's|#DNS=[DNS=1.1.1.1|g' /etc/systemd/resolved.conf
lxc exec $NAME -- systemctl restart systemd-resolved
lxc exec $NAME -- bash -c 'echo -e "[Match]\nName=*\n\n[Network]\nDHCP=ipv4" > /etc/systemd/network/10-all.network'
lxc exec $NAME -- systemctl restart systemd-networkd
```

```
sleep 2

# Installation des packages
lxc exec $NAME apt update
lxc exec $NAME -- apt install apache2 -y
lxc exec $NAME -- apt install php -y

# Récupération du répertoire git
lxc exec $NAME -- git clone 'https://github.com/Steven-176/JenkinsP.git' /var/www/html/
sleep 2

# Supprime dossier test du conteneur pre-prod
lxc exec $NAME /bin/bash/ rm /var/www/html/test/

# Eteins le conteneur prod
lxc stop prod

# Supprime conteneur prod
lxc delete prod

# Renomme conteneur pre-prod en prod
lxc rename $NAME prod
```