

Déploiement : de l'automatisation à la containérisation

Simon Urli
urli@i3s.unice.fr

LP IDSE - IUT de Nice - 2016

Déployer ? Quesaco?

- Rendre accessible le logiciel créé à l'utilisateur final
 - De nombreuses formes de déploiements possibles : en fonction du type de logiciel, de l'utilisateur cible, de la plateforme cible, etc
- > Exemples :
- > Déployer un web service sur un serveur d'application
 - > Déployer une application mobile sur un store
 - > Déployer un site web

Déployer ? Comment ?

- En premier : compiler / packager le logiciel à déployer
- Puis : réaliser un ensemble d'instructions pour faire le déploiement
- Comme d'habitude : processus répétitif, ennuyeux, source d'erreur

Exemple de déploiement

- Déployer un web service :
 1. Compiler / Packager : `chmod +x build.sh`
`&& ./build.sh`
 2. Envoyer le .war au bon endroit sur le serveur :
`scp target/uno-3.0.1.war urli@iut-
outils-gl.i3s.unice.fr:/opt/jetty/
webapps/uno.war`

Exemple de déploiement (suite)

- `chmod +x build.sh && ./build.sh`
- `scp target/uno-3.0.1.war urli@iut-outils-gl.i3s.unice.fr:/opt/jetty/webapps/uno.war`
- ça parait facile ? Qu'est ce qui peut planter ?
 - => Déjà le build. Puis le nom du fichier dans le répertoire target/
 - => Problème de droit pour la copie dans jetty
 - => Jetty non configuré pour réinstaller le war
 - => Et mysql ?
 - => ...

Parfois c'est un peu plus compliqué...

```
cd /home/pulsetotem/pulsetotem_repos/admin
```

```
git checkout <branch>
```

```
git pull origin <branch>
```

```
npm install
```

```
bower install
```

```
sed -i '/backendUrl/d' ./app/common/services/adminConstants.js
```

```
sed -i -r ':a; s%(.*)/\*.*\*/%\1%; ta; /\*/ !b; N; ba' ./app/common/services/adminConstants.js
```

```
sed -i '/^$/d' ./app/common/services/adminConstants.js
```

```
sed -i '4ibackendUrl: "https://xxxxxxxxx.fr/",' ./app/common/services/adminConstants.js
```

```
grunt build
```

```
git checkout -- ./app/common/services/adminConstants.js
```

```
rm -rf /home/pulsetotem/pulsetotem_prod/admin
```

```
mkdir /home/pulsetotem/pulsetotem_prod/admin
```

```
cp -r ./dist/* /home/pulsetotem/pulsetotem_prod/admin/
```

Solution ? Automatisation !

- Etape 1 : script de déploiement avec les instructions
 - Attention : on ne construit jamais le script « from scratch »
 - Script construit par expérience : en fonction des manipulations qu'on a dû faire pour le premier déploiement
 - **Premier déploiement qui s'accompagne de la configuration des outils : installation de jetty, etc**

Premier déploiement

DOCUMENTATION

Automatisation par script ?

C'est tout ?

- Problèmes du script de déploiement :
 - Nom du serveur en dur dans le code ?
 - Parfois des informations sensibles (mot de passe d'accès à git ?!)
 - Comment chaîner le script de déploiement à un build jenkins ?
 - Comment avoir une trace des déploiements effectués ?

Solution ?

Des outils de déploiement !

- L'exemple de Rundeck :
 - Open-source
 - Permet de gérer des déploiements à travers une interface web
 - Gestion de plusieurs serveurs
 - Visualisation de l'ensemble des déploiements réalisés
 - Déploiement par script
 - Déclenchement distant des déploiements (plugin Jenkins etc)

Limites de l'approche

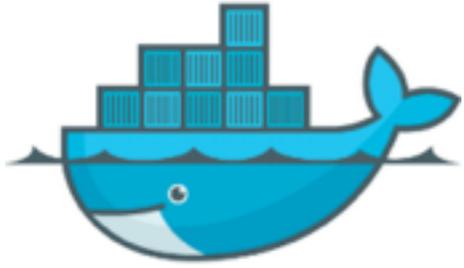
- Automatiser un script : ok mais sur quelle machine ?
- Quid du premier setup de la machine ?
- Comment faire pour déployer sur une nouvelle machine ?

Limites de l'approche

- Automatiser un script : ok mais sur quelle **machine** ?
- Quid du premier setup de la **machine** ?
- Comment faire pour déployer sur une nouvelle **machine** ?

Vers l'utilisation de VM ?

- VM : Virtual Machine
- Idée :
 - supporter sur une même machine **physique** plusieurs machines **virtuelles**
 - pouvoir facilement cloner une machine et toute sa configuration
- Inconvénient :
 - Nécessite beaucoup de ressources...



Docker : créer des container virtuels

- Container = une configuration créée **spécifiquement pour vos besoins** qui sera exécuté **dans sa propre petite VM**
- Utilisation : définition d'un fichier Dockerfile
- ... et lancement du fichier
- C'est tout !

HelloWorld Docker

```
FROM docker/whalesay:latest
```

```
RUN apt-get -y update && apt-get install -y  
fortunes
```

```
CMD /usr/games/fortune -a | cowsay
```

- Première ligne : création d'un container à partir d'un container existant -> on reprend toute la config déjà réalisée
- 2ème ligne : on installe de nouveaux paquets
- 3ème ligne : exécution d'un programme (ici affichage d'un message)

Quid du déploiement en production ?

- Configuration d'un serveur de déploiement Docker
- Mapping des ports entre toutes les VM et la machine physique !
- Possibilité de déployer de multiple fois une même application pour la scalabilité
- Plus d'infos ? <https://www.docker.com/>