

# **Le déploiement continu**

## **Les éléments essentiels**

# Simple et efficace !

Cette formation s'adresse à des développeurs, pas des Admin Sys !  
On va donc découvrir deux méthodes **simple** et **efficace**.

Deux méthodes au programme :

- Avec un *script* dans le pipeline
- Avec l'outil *deployer*

# Déployer avec un script

L'approche la plus *simple* est le déploiement au travers d'un script :

- En **bash**
- Au travers d'une connexion **SSH**

Les avantages sont nombreux :

- Très simple à maintenir
- Lisible et abordable pour une équipe qui débute

# Un exemple de script (pour GitLab)

*Nous mettrons en place ce genre de script ensemble plus tard dans la formation, pas d'inquiétudes !*

```
- 'which ssh-agent || ( apt-get update -y && apt-get install openssh-client git -y )'  
- eval $(ssh-agent -s)  
- echo "$SSH_PRIVATE_KEY" | tr -d '\r' | ssh-add - > /dev/null  
- mkdir -p ~/.ssh  
- chmod 700 ~/.ssh  
- '[[ -f /.dockerenv ]] && echo -e "Host *\n\tStrictHostKeyChecking no\n\n" >> ~/.ssh/config'  
- cd /var/www  
- git pull  
- composer install  
- symfony console d:m:m -n  
- npm install  
- npm run build
```

# Déployer avec un outil : Deployer

Lorsque les besoins de déploiement sont plus complexe, l'utilisation d'un outil *spécialisé* peut apporter des solutions élégante.

- Outil spécialisé dans le déploiement **PHP**
- **Fabrication** du scénario de déploiement en **PHP**
- Offre un cadre plus **standardisé**
- Offre des modèles pour les Framework populaire, dont **Symfony**

Pour une équipe plus mature, cette approche est à privilégier, elle offre une plus large palette de possibilités.

# Extrait avec Deployer

```
<?php // Extrait simplifié et incomplet, c'est pour l'exemple :)

add('recipes', ['symfony']);

desc('Clears cache');
task('deploy:cache:clear', function () {
    if (false !== strpos(get('composer_options', ''), '--no-scripts')) {
        run('{{bin/console}} cache:clear {{console_options}}');
    }
});

task('deploy', [
    'deploy:cache:clear',
]);
```