TREEPTiK.

THE CLOUD & JAVA COMPANY

jres
Montpellier 2015

TREEPTIK.

:: Gérer une infrastructure immuable dans des conteneurs avec Docker ::

⇒ Qu'est ce qu'une infrastructure immutable ?

⇒ Quels sont les *pour* et les *contre* ?

⇒ Comment implémenter ceci avec des containers ?

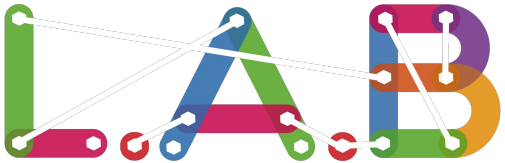
⇒ Démonstrations

TREEPTiK.

:: Nicolas MULLER ::

15 ans de BTP JAVA

Go, Docker, Shell, Linux… ADA



jres Montpellier 2015

TREEPTIK.

:: Infrastructure immutable ::

jres Montpellier 2015

TREEPTIK.

:: Règle numéro 1 : ne jamais changer quoi que ce soit sur le serveur ::

⇒ Ne pas installer de nouveaux packages

⇒ Ne pas mettre à jour de nouveaux packages

⇒ Ne pas supprimer de packages ou les downgrader

⇒ *Même pour raisons de sécurité*

⇒ Ne pas éditer de fichiers de configurations

⇒ Ne pas mettre à jour votre code métier

⇒ *Même en cas d'urgence client*

TREEPTIK.

:: Règle numéro 2 : si vous êtes tenté de modifier quoique ce soit sur le serveur ::

Relisez la règle numéro 1

Avec une petite exception plus tard...

TREEPTIK.

:: Comment se mettre à jour ? ::

⇒ Créer un nouveau serveur *from scratch*

⇒ Appliquer le processus de déploiement (scripts shell, chef, ansible…)

⇒ *Tester le nouveau serveur si le coeur vous en dit...*

⇒ Remplacer l'ancien serveur par le nouveau

⇒ *Garder l'ancien quelque temps au cas où...*

TREEPTIK.

:: Infrastructure immutable ::

Serveurs immutables

Serveurs phénix

TREEPTIK

POURQUOI ?!?

jres Montpellier 2015

TREEPTIK

:: Eviter le *Drift* ::

TREEPTIK.

:: Le Drift ::

⇒ La différence entre serveurs supposés être identiques

⇒ La cause :

- Un provisionnement fait à un moment différent
- Une opération manuelle

⇒ Conséquences

- Des erreurs aléatoires
- Même code métier mais des comportements différents
- Ca glisse encore plus avec le temps...

TREEPTIK.

:: Faire face au danger ::

⇒ Documenter les opérations manuelles

⇒ Automatiser les tâches mais jusqu'où ?

⇒ Utiliser des outils de déploiement de configuration

TREEPTIK.

:: L'échec de l'automatisation brute ::

⇒ Utilisons ***parallel-ssh*** ou tout autre outil

⇒ Problème interne

- Un des serveurs est inaccessible
- Devient inaccessible durant le processus
- Rencontre un problème de concurrence

⇒ Problème externe

- Dépôts de packages de distribution
- Dépôts de sources

TREEPTIK

:: L'échec de l'automatisation par configuration ::

- ⇒ Gestion des rollbacks difficiles
- ⇒ Package non disponibles sur les dépôts
  - version précédente
- ⇒ Risque de mise à jour vers une version non souhaitée

TREEPTIK.

:: Avec des serveurs immutables ::

⇒ Nous avons toujours le vieux serveur

⇒ Le remettre en serveur

⇒ Faire la mise à jour du récent

⇒ Inverser les deux

⇒ Facilité des rollbacks...

⇒ Mais ne résout pas le problème du drift

TREEPTIK.

:: Jeter vos serveurs ::

⇒ Reprovisionner régulièrement vos serveurs

⇒ Assurer vous que vous avez toujours des packages récents

⇒ Toute action manuelle sera supprimée car effacée par la machine

TREEPTIK.

:: The golden image ::

⇒ Créer un serveur from scratch

⇒ Appliquer le processus de déploiement/configuration

⇒ Snapshoter ce serveur pour en faire une image de référence

⇒ Créer plusieurs serveurs à partir de cette image

Réduit les nombres d'erreurs dans le processus

Historique à bas coût des versions

TREEPTIK

Malgré tout...

jres Montpellier 2015

TREEPTIK.

:: Les petits changements ::

⇒ Modifier une ligne de CSS

⇒ Précédemment ...

- Modification à la main
- Validation équipe
- Réplication partout en quelques minutes

⇒ Selon les nouvelles règles

- Modification à la main, validation équipe…
- Création d'une nouvelle image de référence : une heure par exemple
- Provisionner les nouveaux serveurs : quelques minutes
- Switch entre anciens/nouveaux serveurs
- Décommissioner les anciens serveurs...

TREEPTiK.

:: Solution ::

Il faut automatiser !!!

⇒ Oui mais… seulement après la phase de validation

⇒ Le temps imparti sera toujours de plus ou moins une heure

⇒ On peut imaginer des images intermédiaires pour gagner du temps

TREEPTIK

:: Problème ::

Le débogage est difficile

⇒ Installer n-fois les mêmes outils sur chaque instance

⇒ Comment conserver les logs entre deux instances ?

- *Flagger* la machine pour ne pas la supprimer
- Sauvegarder les logs sur un NAS

TREEPTIK

:: La solution ::

jres Montpellier 2015

TREEPTIK.

Containers immutables

jres Montpellier 2015

TREEPTIK

:: Retour arrière... ::

⇒ Création d'une image

- from scratch : long et facile
- intermédiaire : plus rapide mais plus complexe

⇒ Déploiement n-fois

Pourquoi ne pas utiliser des containers ?

TREEPTIK.

:: Construire des containers images ::

⇒ Prendre le meilleur de chaque part

- from scratch : construction propre sans effet de bords
- intermédiaire : construction rapide avec des changements mineurs

⇒ Pourquoi et comment ?

- les snapshots de containers ne coûtent rien (secondes versus minutes)
- chaque étape = une commande = un snapshot

TREEPTIK

:: Problème ::

Le débogage est difficile

⇒ Installer n-fois les mêmes outils sur chaque instance

⇒ Comment conserver les logs entre deux instances ?

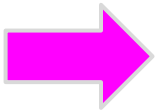
- *Flagger* la machine pour ne pas la supprimer
- Sauvegarder les logs sur un NAS

TREEPTIK

```
FROM debian:jessie
MAINTAINER Jessica Frazelle <jess@docker.com>

# Install dependencies
RUN apt-get update && apt-get install -y \
    build-essential \
    ... \
    --no-install-recommends

# Install node
RUN curl -sL https://deb.nodesource.com/setup | bash -
RUN apt-get install -y nodejs

# Clone atom
RUN git clone https://github.com/atom/atom /src
WORKDIR /src
RUN git fetch && git checkout \
    $(git describe --tags \
          `git rev-list --tags --max-count=1`)
RUN script/build && script/grunt install

# Autorun atom
CMD /usr/local/bin/atom --foreground
```

TREEPTIK

:: Premier build ::



```
FROM debian
RUN apt-get xxx
COPY ./src
RUN /src/build
```

⇒ Créer un container depuis une image de base

⇒ Mettre à jour les packages et faire un snapshot

⇒ Créer un container depuis ce snapshot

⇒ Copier les sources dans /src et faire un snapshot

⇒ Créer un container depuis ce snapshot

⇒ Exécuter /src/build dans ce container et faire un snapshot

Le snapshot final est notre image

TREEPTIK

:: build suivant ::

⇒ Avant d'exécuter chaque étape vérifiez si ceci n'a pas déjà été fait

- Si oui, on n'utilise le snapshot
- Sinon on exécute les tâches puis on snapshote

⇒ Dérouler le processus jusqu'à ce qu'un step change

Le résultat final est le même qu'une reconstruction totale mais bien plus rapide

TREEPTIK.

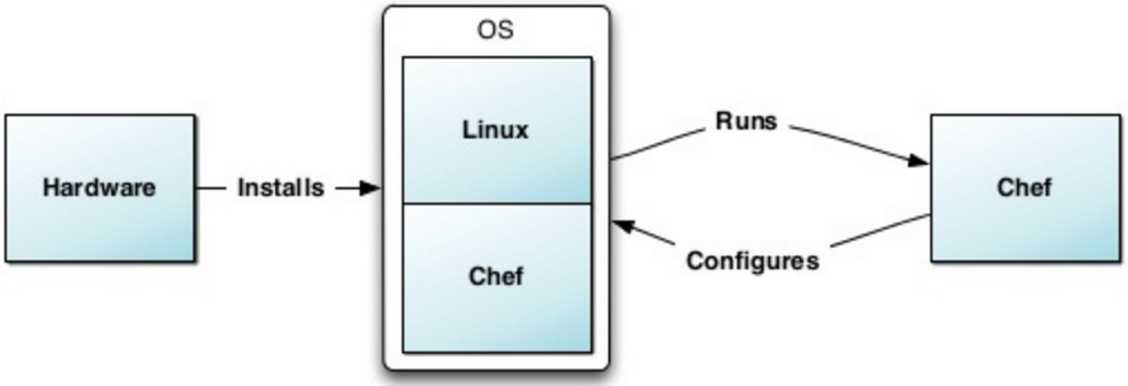
:: Avantages Docker ::

- ⇒ Utilisation des volumes
- ⇒ La Mise à jour est rapide
- ⇒ Le FS des containers peut être RO
  - Force l'immutabilité
  - plus facile à auditer
- ⇒ Moins coûteux
  - Consolidation
  - Réduire le coût sur les infrastructures IaaS

Le snapshot final est notre image

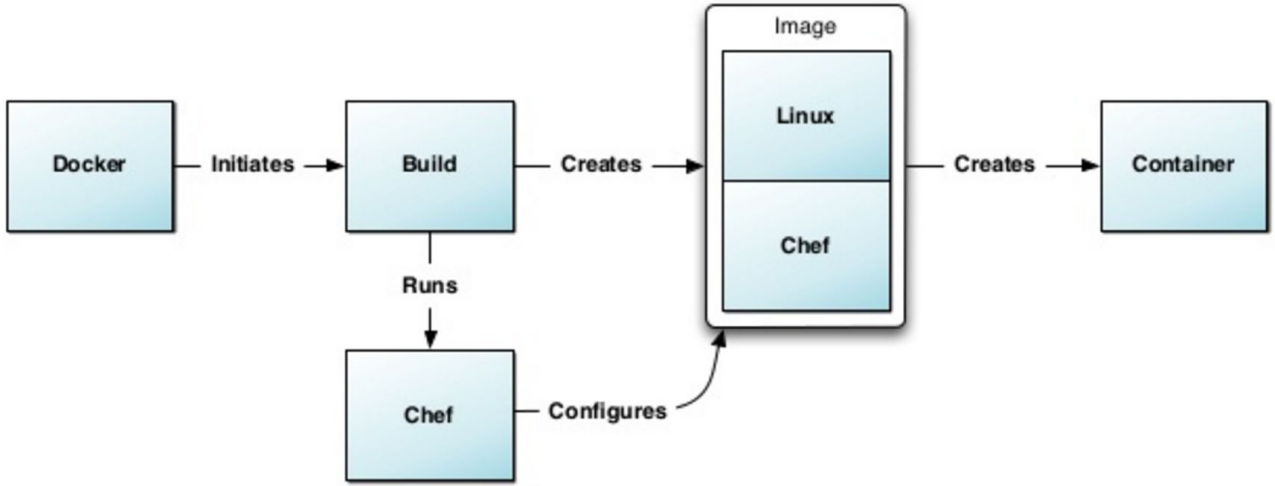
TREEPTIK

:: Chef traditionnel ::



TREEPTIK

:: Chef + Docker ::



Merci de votre attention