

CHEF: Mise en oeuvre d'une architecture de déploiement

Table of Contents

- [CHEF: Mise en oeuvre d'une architecture de déploiement](#)
- [Présentation des composants](#)
 - [Hyperviseur](#)
 - [Vagrant](#)
 - [Chef](#)
 - [Le serveur Chef](#)
 - [Station de travail \(workstation\)](#)
 - [Chef-solo](#)
 - [chef-client](#)
 - [Knife](#)
 - [Liste des dépôts du serveur Chef](#)
 - [Dépôts gérés par APACHE](#)
 - [Dépôts gérés par RUBY](#)
 - [Dépôts gérés par NGINX](#)

La structure de déploiement mise en oeuvre comporte l'architecture

- Une station de travail (chef-workstation)
- Le serveur d'hypervision sur lequel sont déployés :
 - serveur chef (chefdk + chef + knife-solo + chef-manage)
 - vagrant
 - des entrepôts locaux (apache)
 - un entrepôt local de paquets gems (geminabox)
 - un entrepôt local vagrant/chef (nginx)

Présentation des composants

Hyperviseur

- KVM, pour Kernel-Based Virtual Machine est un hyperviseur qui permet de virtualiser presque tous les systèmes basés sur une architecture x86 32 ou 64 bits. C'est un fork de QEMU intégré dans le noyau Linux depuis la version 2.6.20.
- libvirt est une API libre, daemon et outil de gestion pour gérer de nombreuses plateformes de virtualisation. Avec cet outil, vous allez disposer d'un ensemble de fonctions vous permettant de configurer et gérer des machines virtuelles Linux KVM, Xen, VMware ESX et qemu.
- Open vSwitch est un commutateur virtuel open source pour Xen, XenServer, KVM et VirtualBox. Il se présente comme un équivalent du commutateur distribué Cisco Nexus 1000V. Il permet

notamment aux administrateurs réseaux de gérer de façon précise le trafic réseau au sein d'un cluster de machines virtuelles, mais aussi de définir des politiques avancées de gestion des ressources du réseau virtuel.

Vagrant

Vagrant est un logiciel libre et open-source pour la création et la configuration des environnements de développement virtuel. Il peut être considéré comme un wrapper autour de logiciels de virtualisation comme KVM/libvirt.

Chef

Chef est un logiciel de gestion de configuration open-source permettant d'automatiser la création, le déploiement et la gestion des infrastructures informatiques. Il simplifie le déploiement de serveurs et applications sur n'importe quel emplacement physique, virtuel ou cloud (appelé nœud) quelque soit la taille de l'infrastructure..

Chef suit le concept d'architecture client-serveur, donc pour travailler avec Chef, il faut configurer une workstation Chef pour y développer la configuration localement, puis déployer sur le serveur Chef les configurations pour les faire fonctionner sur les nœuds Chef.

Le serveur Chef

Chef Server joue le rôle de centre d'information. Il stocke tous les 'cookbooks' et politiques qui seront appliqués aux nœuds, ainsi que les 'node objects' décrivant chaque nœud géré par ce Chef Server.

Les 'node objects' comportent deux aspects importants : les attributs et les 'run-lists'. Les attributs permettent de stocker des données spécifiques sur un nœud, comme par exemple un fichier système.

Les 'run-lists' sont des listes ordonnées de recettes de recettes et de rôles qui sont exécutés dans un ordre précis sur un nœud par chef-client afin de mettre le nœud en conformité avec la politique définie.

De cette manière, les détails de configuration décrite dans un livre de cuisine sont appliqués aux nœuds auxquels s'applique le scénario décrit dans le livre de recettes . Un inconvénient notable de Chef : les 'cookbooks' sont en push only du Chef Server vers les chef-client. Cette situation ne posera aucun soucis en utilisation normale, mais peut s'avérer critique si les nœuds ont besoin d'être souvent mis à jour.

Station de travail (workstation)

Un poste de travail est un ordinateur exécutant le kit Chef Development Kit (Chef DK) utilisé pour créer des cookbooks, interagir avec le serveur Chef et interagir avec les nœuds.

Le poste de travail est l'endroit où les utilisateurs font la plupart de leur travail :

- Développer et tester des livres de recettes et des recettes
- Test du code du chef
- Garder le référentiel Chef synchronisé avec le contrôle de la source de la version
- Configuration de la stratégie organisationnelle, y compris la définition des rôles et des environnements, et vérification de la conservation des données critiques dans des sacs de données
- Interagir avec les nœuds, selon les besoins (tels que l'exécution d'une opération d'amorçage)
- Installer chef en fonction de la destination du nœud

Chef-solo

Pour certains projets, il est souhaitable de ne pas travailler avec un chef-serveur comme un référentiel de code chef. Pour ce faire, on peut utiliser chef-solo ou utiliser le chef en mode local.

chef-client

Chef-client va permettre de récupérer les informations du Chef Server et de mettre à jour localement le nœud en fonction de ces informations sont transmises via les 'cookbooks' et recettes associées (indiquant les ressources nécessaires et leurs états désirés). Ce sont ces mêmes recettes qui permettent d'installer et de configurer des composants logiciels, de déployer des applications...

Knife

knife est un outil de ligne de commande qui fournit une interface entre un chef-repo local et le serveur Chef.

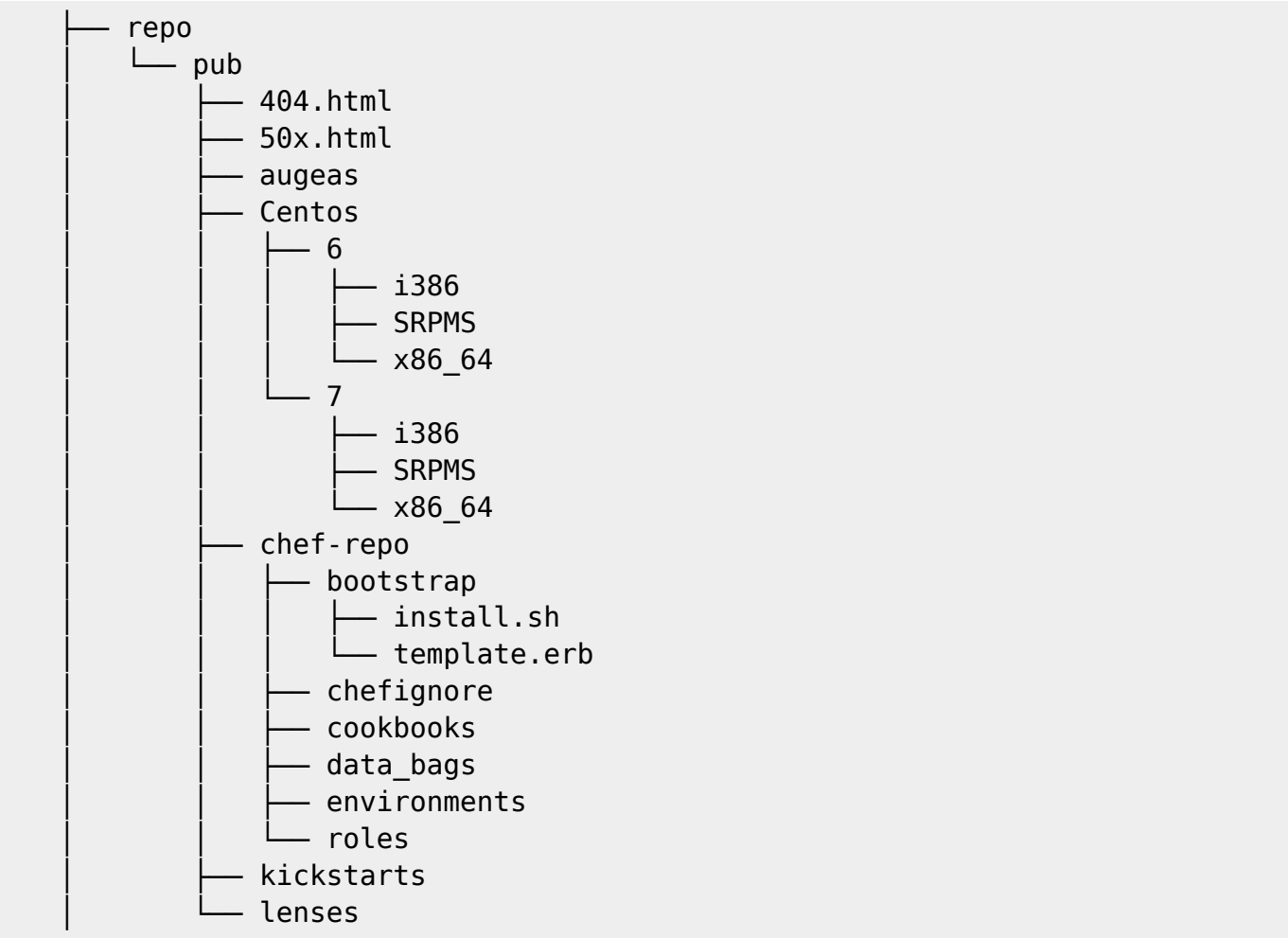
Knife aide les utilisateurs à gérer:

- Nœuds
- cookbooks et recettes
- Rôles, environnements et data bags
- Ressources dans divers environnements de cloud
- L'installation du chef-client sur les nœuds
- Recherche de données indexées sur le serveur Chef

Liste des dépôts du serveur Chef

Dépôts gérés par APACHE

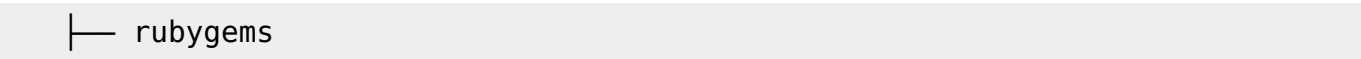
Entrée	Contenu
root	/var/www/html/repo/pub
Centos, augeas, kickstarts, lenses	dépôts locaux
chef-repo	dépôt contenant la procédure de bootstrap locale et les données décrivant chaque nœud gérés par le serveur Chef (livres de recettes, stratégies appliquées aux nœuds et métadonnées)



Dépôts gérés par RUBY



Pour faciliter l'installation des GEMS un dépôts RUBYGEMS doit être accessible depuis un serveur gem local [serveur gem local](#)



Dépôts gérés par NGINX

Entrée	Contenu
root	/var/www/html/vagrant
boxes	dépôt local de box vagrant
/<organization>/<distribution>	catalogue du dépôt local contenant un fichier <distribution>.json et un Vagrantfile de test



les répertoires **sandbox** et **prod** où sont exécutées les VM ne sont pas accessibles via l'url

```
├── vagrant
│   ├── boxes
│   │   ├── dgfipbase_1.0.0.box
│   │   └── dgfip-rhel7-0.4.0.box
│   ├── dgfip
│   │   ├── centos6
│   │   │   ├── centos6.json
│   │   │   └── Vagrantfile
│   │   ├── centos7
│   │   │   ├── centos7.json
│   │   │   └── Vagrantfile
│   ├── prod
│   ├── sandbox
│   └── Vagrantfile
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=vagrant:chef-local>

Last update: **2025/02/19 10:59**

