

## Installation et configuration du serveur Chef

La version Ubuntu Linux 16.04 LTS fonctionne sous Debian.Y récupérer et installer la dernière version 64 bits du paquetage Chef.

Pour l'installer procédez comme suit :

```
wget https://packages.chef.io/files/stable/chef-server/12.16.2/ubuntu/16.04/chef-server-core_12.16.2-1_amd64.deb
dpkg -i chef-server-core_12.16.9-1_amd64.deb
chef-server-ctl reconfigure
chef-server-ctl user-create
                        user_name first_name
                        last_name email 'PASSWORD'
                        --filename user.pem

chef-server-ctl org-create short_name 'full_organization_name'
                        --association_user user_name
                        --filename ORGANIZATION-validator.pem
```

**Exemple :**

```
chef-server-ctl user-create admin
                        admin
                        admin
                        admin@example.org
                        12345678
                        -f admin.pem

chef-server-ctl install chef-manage
chef-server-ctl org-create exemple-org « Exemple.org »
                        --association_user admin
                        --filename exemple-org-validator.pem
```

## Installation et configuration de Chef-manage

Une fois que cela est fait, il faut se rendre sur cette page <https://downloads.chef.io/chef-manage/ubuntu/> (<https://downloads.chef.io/chef-manage/ubuntu/>) pour récupérer la dernière version de Chef Manage ( /c'est grâce à ce paquetage qu'on a une interface graphique via l'adresse web [https://server\\_domain\\_or\\_IP](https://server_domain_or_IP) ([https://server\\_domain\\_or\\_IP](https://server_domain_or_IP)) )

```
wget https://packages.chef.io/files/stable/chef-manage/2.5.4/ubuntu/16.04/chef-manage_2.5.4-1_amd64.deb
dpkg -i chef-manage_2.5.4-1_amd64.deb
chef-server-ctl reconfigure
chef-manage-ctl reconfigure
```

Le lien web pour accéder à l'interface graphique est : <https://chef.exemple.org/organizations/exemple-org/> (<https://chef.exemple.org/organizations/exemple-org/>)

**\*\*ATTENTION\*\*:**

Si le message "This installation is managing X nodes, but is only licensed for 25. Please upgrade now to ensure proper licensing." apparaît, il n'est pas bloquant et on peut si on veut utiliser Chef sans avoir l'interface web. Il faut à ce moment là désinstaller le paquetage chef-manage. Le message signalé ci dessus n'apparaît que depuis la version 12 de Chef et il est lié au paquetage chef-manage n'altère en rien le bon fonctionnement d'un serveur Chef.

## Mise à jour de Chef

Pour mettre à jour Chef sur le serveur Chef il faut se rendre à la page de téléchargement du paquetage : <https://downloads.chef.io/chef-server/ubuntu/> (<https://downloads.chef.io/chef-server/ubuntu/>)

```
dpkg -i chef-server-core_12.16.14-1_amd64.deb
chef-server-ctl upgrade
chef-server-ctl start
chef-server-ctl cleanup
```

**\*\*ATTENTION\*\*:**

Lors de la mise à jour de Chef vers la version 12.15.6 il y a un message d'erreur qui a été résolu en faisant ce qui est indiqué sur le lien suivant : <https://github.com/chef/chef-server/issues/1220>

Procédez comme suit pour mettre à jour la version Chef-manage si vous l'utilisez :

```
wget https://packages.chef.io/stable/ubuntu/16.04/chef-manage_2.5.4-1_amd64.deb
dpkg -i chef-manage_2.5.4-1_amd64.deb
chef-manage-ctl reconfigure
```

Une fois cela terminé il faut modifier le fichier new.html.erb du répertoire /opt/chef-manage/embedded/service/chef-manage/app/views/sessions pour y enlever la partie Already have an account. Faire comme indiqué ci-dessous devrait suffire :

```
cd /opt/chef-manage/embedded/service/chef-manage/app/views/sessions
cp new.html.erb-a-garder new.html.erb
chef-manage-ctl graceful-kill
chef-manage-ctl start
```

Il faut ensuite vérifier qu'on peut s'authentifier via la page web: <https://chef.exemple.org> (~~https://chef.exemple.org~~) et que la partie Already have an account sur celle-ci a bien disparu comme on veut.

Si tout fonctionne bien on a alors mis la version Chef et Chef-manage à jour sur le serveur.

## Procédure pour l'installer le référentiel Chef :

```
wget https://packages.chef.io/files/stable/chefdk/2.1.11/ubuntu/16.04/chefdk_2.1.11-1_amd64.deb
dpkg -i chefdk_*.deb
chef verify
```

La commande chef verify permet de vérifier l'installation du serveur Chef :

```
info@chef-dk ~ $ chef verify
Running verification for component 'berkshelf'
Running verification for component 'test-kitchen'
Running verification for component 'tk-policyfile-provisioner'
Running verification for component 'chef-client'
Running verification for component 'chef-dk'
Running verification for component 'chef-provisioning'
Running verification for component 'chefspect'
Running verification for component 'generated-cookbooks-pass-chefspect'
Running verification for component 'rubocop'
Running verification for component 'fauxhai'
Running verification for component 'knife-spork'
Running verification for component 'kitchen-vagrant'
Running verification for component 'package installation'
Running verification for component 'openssl'
Running verification for component 'inspec'
.....
-----
Verification of component 'rubocop' succeeded.
Verification of component 'kitchen-vagrant' succeeded.
Verification of component 'openssl' succeeded.
Verification of component 'berkshelf' succeeded.
Verification of component 'tk-policyfile-provisioner' succeeded.
Verification of component 'fauxhai' succeeded.
Verification of component 'inspec' succeeded.
Verification of component 'test-kitchen' succeeded.
Verification of component 'chef-dk' succeeded.
Verification of component 'chefspect' succeeded.
Verification of component 'knife-spork' succeeded.
Verification of component 'chef-client' succeeded.
Verification of component 'package installation' succeeded.
Verification of component 'chef-provisioning' succeeded.
Verification of component 'generated-cookbooks-pass-chefspect' succeeded.
```

Exécutez ensuite les commandes suivantes :

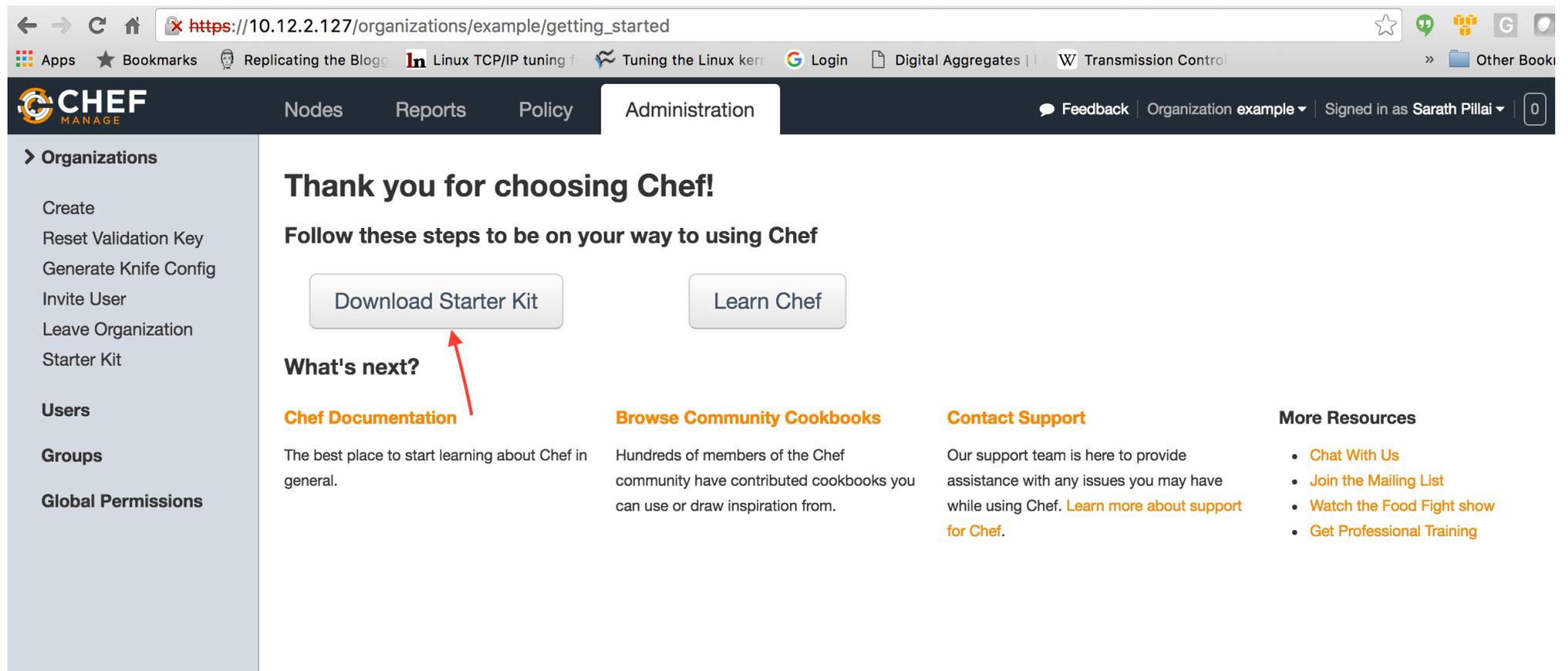
```
echo 'eval "$(chef shell-init bash)"' >> ~/.bash_profile
scp root@server_domain_or_IP:/root/admin.pem ~/.chef
scp root@server_domain_or_IP:/root/exemple-org-validator.pem ~/.chef
```

## Autre façon de procéder pour l'installation du référentiel ( plus rapide ) :

Le kit de démarrage (Chef starter kit) n'est qu'une archive contenant la structure du répertoire pour effectuer des travaux de développement sur le référentiel Chef. Il contient les paramètres du livre de recettes et la configuration de knife.

Vous pouvez simplement le télécharger et copier l'archive sur votre référentiel Chef.

L'image ci-dessous montre la page de téléchargement du kit depuis l'interface utilisateur Chef installée par Chef-manage.



The screenshot shows the Chef Manage web interface. The browser address bar displays `https://10.12.2.127/organizations/example/getting_started`. The navigation menu includes 'Nodes', 'Reports', 'Policy', and 'Administration'. The user is signed in as 'Sarath Pillai'. The main content area displays a 'Thank you for choosing Chef!' message and a 'Follow these steps to be on your way to using Chef' section. A 'Download Starter Kit' button is highlighted with a red arrow. Below this, the 'What's next?' section provides links to 'Chef Documentation', 'Browse Community Cookbooks', 'Contact Support', and 'More Resources'.

Une fois téléchargé il faut décompresser l'archive chef-starter.zip comme suit :

```
unzip chef-starter.zip
```

La commande ci-dessus créera un répertoire appelé chef-repo dans l'emplacement actuel. Ce répertoire contient tous les paramètres de base pour interagir avec le serveur chef du poste de travail.

Il a également un répertoire de livre de recettes à l'intérieur, où tous les livres de cuisine (cookbooks) seront stockés et ensuite transférés vers le serveur Chef.

```
root@chef-dk:/home/ubuntu/chef-repo# ls -la
total 32
drwxr-xr-x 5 root  root  4096 Jun 30 02:45 .
drwxr-xr-x 5 ubuntu ubuntu 4096 Jun 30 02:45 ..
drwxr-xr-x 4 root  root  4096 Jun 30 03:01 .chef
drwxr-xr-x 4 root  root  4096 Jun 30 02:59 cookbooks
-rw-r--r-- 1 root  root   495 Jun 30 02:30 .gitignore
-rw-r--r-- 1 root  root  2341 Jun 30 02:30 README.md
drwxr-xr-x 2 root  root  4096 Jun 30 02:30 roles
```

Il existe un répertoire appelé ".chef" comme indiqué ci-dessus. Ce répertoire .chef contient la clé privée pour l'utilisateur. En outre, ce répertoire ".chef" contient un fichier de configuration de knife appelé **knife.rb**.

```
root@chef-dk:/home/ubuntu/chef-repo/.chef# ls -l
total 16
-rw-r--r-- 1 root root  423 Jun 30 02:30 knife.rb
-rw-r--r-- 1 root root 1678 Jun 30 02:30 admin.pem
```

## Le fichier de configuration knife.rb

Le fichier **knife.rb** doit être créé dans le répertoire ~/.chef du référentiel chef-dk à partir de laquelle on va écrire les recettes Chef et il doit être de la forme :

```
current_dir = File.dirname(__FILE__)
log_level    :info
log_location STDOUT
node_name    'node_name'
client_key   '#{current_dir}/USER.pem'
validation_client_name 'chef-validator'
validation_key   '#{current_dir}/ORGANIZATION-validator.pem'
```

```
chef_server_url      'https://chef.exemple.org/organizations/ORG_NAME'  
cache_type           'BasicFile'  
cache_options( :path => "#{ENV['HOME']}/.chef/checksums" )  
cookbook_path        ['#{current_dir}/../cookbooks']
```

**client\_key**: spécifie le chemin du fichier de clé privée associé à l'utilisateur. Cette clé privée sera utilisée pour authentifier le poste de travail auprès du serveur chef.

**chef\_server\_url**: ceci est l'URL complète du serveur chef avec le chemin de l'organisation (nous avons créé une organisation lors de notre précédent tutoriel d'installation du serveur chef d'entreprise. Le nom de notre organisation est "exemple-org")

**cookbook\_path**: c'est le chemin absolu où les livres de recettes seront stockés sur le poste de travail. Comme knife synchronisera les livres de recettes sur le serveur, il téléchargera également les livres de recettes, créera de nouveaux livres de recettes, etc.

**validation\_client\_name**: c'est le nom d'utilisateur associé à l'organisation. Nous avons créé un utilisateur nommé "admin" plus haut.

```
root@chef-dk:/home/ubuntu/chef-repo# knife ssl fetch  
WARNING: Certificates from chef.exemple.com will be fetched and placed in your trusted_cert  
directory (/home/ubuntu/chef-repo/.chef/trusted_certs).
```

```
Knife has no means to verify these are the correct certificates. You should  
verify the authenticity of these certificates after downloading.
```

```
Adding certificate for chef.exemple.com in /home/ubuntu/chef-repo/.chef/trusted_certs/chef_exemple_org.crt
```

La commande ci-dessus interagit avec le serveur chef (en utilisant l'URL définie dans le fichier knife.rb) et prend les certificats SSL du serveur. Ce certificat SSL est ensuite stocké dans un nouveau répertoire appelé trusted\_certs.

Une fois que les certificats SSL y sont ajoutés et que **knife** les a approuvés, vous pouvez ensuite vérifier la connectivité au serveur Chef en utilisant la commande ci-dessous.

```
root@chef-dk:/home/ubuntu/chef-repo# knife ssl check  
Connecting to host chef.exemple.com:443  
Successfully verified certificates from `chef.exemple.org'
```

Si les choses sont correctement configurées vous devriez pouvoir voir la sortie ci-dessus. Maintenant que tout fonctionne, la commande knife peut effectuer toutes les opérations sur le serveur Chef en utilisant l'utilisateur que nous avons créé (qui est défini à l'intérieur de knife.rb) et l'organisation associée à cet utilisateur.

Bootstrapper un nœud n'est rien d'autre que le travail d'installation et de configuration de chef-client sur un serveur qui doit être automatisé par l'outil Chef. L'installation et la configuration d'un nœud avec l'agent chef commencera par le téléchargement de toutes les recettes Chef nécessaires depuis le serveur en passant dans un premier temps par une connexion ssh depuis le référentiel Chef vers le client. Elle s'effectue avec une commande **knife** simple comme indiqué ci-dessous:

```
knife bootstrap 10.12.2.23
    --ssh-user info
    --ssh-password 'password'
    --node-name node1.exemple.org
```

On peut aussi utiliser la commande suivante pour utiliser les clefs SSH :

```
knife bootstrap 10.12.2.23
    --ssh-user ubuntu --sudo
    --identity-file privatekey.pem
    --node-name node1.exemple.org
```

La commande **chef-client** s'exécute depuis le nœud et permet son installation et sa configuration. Le client Chef télécharge depuis le serveur Chef toutes les recettes Chef nécessaires et les exécute. Le processus d'exécution est différent de celui d'avec la commande **knife bootstrap**.

## Comment créer et télécharger un livre de recettes sur le serveur Chef depuis le référentiel ?

La deuxième opération que le référentiel fait est de créer un livre de recettes qui constitue le principal bloc de configuration de base qui sera appliqué sur les nœuds.

Vous pouvez créer un livre de recettes squelette **test** en utilisant la commande ci-dessous:

```
knife cookbook create test
```

**Résultat de la commande :**

```
chef
** Creating cookbook test
** Creating README for cookbook: test
** Creating CHANGELOG for cookbook: test
** Creating metadata for cookbook: test
```

N'oubliez pas le fait que les commandes knife doivent être effectuées à partir du répertoire chef-repo (car il nécessite un répertoire .chef, qui possède le fichier de paramètres knife.rb)

Pour faire la mise à jour du cookbook **test** depuis le référentiel Chef (là où on a installé le paquetage chef-dk) vers le serveur Chef il faut utiliser la commande suivante:

```
knife cookbook upload test
```

**Résultat de la commande :**

```
Uploading installation_logiciels [0.1.0]  
Uploaded 1 cookbook.
```

## Exemple d'exécution de commande chef-client sur le noeud :

```
root@test:~# chef-client
Starting Chef Client, version 13.3.42
resolving cookbooks for run list: ["installation_logiciels::install"]
Synchronizing Cookbooks:
  - installation_logiciels (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 36 resources
Recipe: installation_logiciels::auto-upgrade
  * template[/etc/apt/apt.conf.d/20auto-upgrades] action create (up to date)
  * template[/etc/apt/apt.conf.d/50unattended-upgrades] action create (up to date)
  * file[/etc/apt/apt.conf.d/50unattended-upgrades.ucf-dist] action delete (skipped due to only_if)
  * file[/etc/apt/apt.conf.d/20unattended-upgrades.ucf-dist] action delete (skipped due to only_if)
  * template[/etc/cron.daily/auto-upgrade] action create (up to date)
  * execute[unattended-upgrades] action run
    - execute /usr/bin/unattended-upgrades
Recipe: installation_logiciels::chef-client
  * remote_file[/tmp/chef_13.3.42-1_amd64.deb] action create (up to date)
  * script[On checke la version de Chef et on met à jour si besoin] action run
    - execute "bash" "/tmp/chef-script20170829-16982-1xlczp6"
Recipe: installation_logiciels::install-bib
  * linux_user[root] action modify (up to date)
  * linux_user[info] action modify (up to date)
  * linux_user[bib] action modify (up to date)
  * template[/etc/apt/apt.conf] action create (up to date)
  * template[/etc/init.d/iptables] action create (up to date)
  * template[/etc/lightdm/lightdm.conf] action create (up to date)
  * directory[/etc/firefox] action create (up to date)
  * template[/etc/firefox/syspref.js] action create (up to date)
  * template[/etc/ssh/sshd_config] action create (up to date)
  * service[ssh] action stop
    - stop service service[ssh]
  * service[ssh] action start
    - start service service[ssh]
  * script[Creation du repertoire /home/bib] action run
```

```
- execute "bash" "/tmp/chef-script20170829-16982-fq8cb8"
```

```
Recipe: installation_logiciels::cron-purge-bib
```

```
* template[/root/purge-home-bib] action create (up to date)
```

```
* template[/etc/cron.daily/purge] action create
```

```
- update content in file /etc/cron.daily/purge from 656a38 to 46978a
```

```
--- /etc/cron.daily/purge 2017-08-28 14:21:17.771458249 +0200
```

```
+++ /etc/cron.daily/.chef-purge20170829-16982-yz93yr 2017-08-29 09:10:49.778906952 +0200
```

```
@@ -1,4 +1,4 @@
```

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/X11R6/bin
```

```
-00 18 * * * root /root/purge-home-bib > /dev/null 2>&1
```

```
+00 23 * * * root /root/purge-home-bib > /dev/null 2>&1
```

```
Recipe: installation_logiciels::ethernet_sleep
```

```
* template[/etc/pm/sleep.d/ethernet_sleep] action create (up to date)
```

```
Recipe: installation_logiciels::fusioninventory
```

```
* apt_package[fusioninventory-agent] action install (up to date)
```

```
* template[/etc/fusioninventory/agent.cfg] action create (up to date)
```

```
* template[/etc/ssl/certs/pharasma_math_u-psud_fr.crt] action create (up to date)
```

```
Recipe: installation_logiciels::cron-fusion
```

```
* template[/etc/default/fusioninventory-agent] action create (up to date)
```

```
Recipe: installation_logiciels::fusioninventory
```

```
* execute[0n lance fusioninventory-agent] action run
```

```
- execute /usr/bin/fusioninventory-agent
```

```
Recipe: installation_logiciels::ntp-bib
```

```
* apt_package[ntp] action install (up to date)
```

```
* apt_package[ntp] action install (up to date)
```

```
* template[/etc/ntp.conf] action create (up to date)
```

```
* template[/etc/default/ntp] action create (up to date)
```

```
* execute[systemctl enable ntp] action run
```

```
- execute systemctl enable ntp
```

```
Recipe: installation_logiciels::rc-local-bib
```

```
* template[/etc/rc.local] action create (up to date)
```

```
Recipe: installation_logiciels::install-bib
```

```
* service[ntp] action stop
```

```
- stop service service[ntp]
```

```
* service[ntp] action start
  - start service service[ntp]
* service[iptables] action stop
  - stop service service[iptables]
* service[iptables] action start
  - start service service[iptables]
```

Running handlers:

Running handlers complete

Chef Client finished, 12/39 resources updated in 08 seconds

**Remarque** : Les fichiers modifiés par Chef sont automatiquement copiés dans le répertoire **/var/chef/backup** du noeud.

Le fichier de log **chef-stacktrace.out** contenant les traces de l'exécution de Chef sont dans le répertoire **/var/chef/cache** du noeud.

Sur le client Chef, nous avons un répertoire **/etc/chef** dans lequel il y a un fichier **client.rb** qui contient les informations liées au client :

```
log_location      STDOUT
chef_server_url   ""https://chef.exemple.org/organizations/Exemple-org
validation_client_name "exemple-org-validator"
node_name        "test"
trusted_certs_dir "/etc/chef/trusted_certs"
```

Ces informations permettent au client Chef de communiquer avec le serveur en SSL. On peut désactiver cela en rajoutant l'option **ssl\_verify\_mode:verify\_none**