# Chef

## Chef简介

### Linux系统中的变量

[root@workstation chef-repo]# export

declare -x EDITOR="vim"

declare -x GEM\_HOME="/root/.chefdk/gem/ruby/2.5.0"

declare -x GEM\_PATH="/root/.chefdk/gem/ruby/2.5.0:/opt/chef-workstation/embedded/lib/ruby/gems/2.5.0"

declare -x GEM\_ROOT="/opt/chef-workstation/embedded/lib/ruby/gems/2.5.0"

declare -x HISTCONTROL="ignoredups"

declare -x HISTSIZE="1000"

declare -x HOME="/root"

declare -x HOSTNAME="workstation"

declare -x LANG="zh\_CN.UTF-8"

declare -x LESSOPEN="||/usr/bin/lesspipe.sh %s"

declare -x LOGNAME="root"

declare -x MAIL="/var/spool/mail/root"

declare -x OLDPWD="/opt/chef-repo/environments"

declare -x PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin"

declare -x PWD="/opt/chef-repo"

declare -x SHELL="/bin/bash"

declare -x SHLVL="1"

declare -x SSH\_CLIENT="192.168.1.1 59846 22"

declare -x SSH\_CONNECTION="192.168.1.1 59846 192.168.1.13 22"

declare -x SSH\_TTY="/dev/pts/0"

declare -x TERM="xterm"

declare -x USER="root"

declare -x XDG\_RUNTIME\_DIR="/run/user/0"

declare -x XDG\_SESSION\_ID="1"

### Chef 工具

是一款自动化服务器配置管理工具，可以对所管理的对象实行自动化配置，如系统管理，安装软件等。Chef 由三大组件组成：Chef Server、Chef Workstation 和 Chef Node。

1、Chef Server

核心服务器，维护了一套配置脚本（Cookbook），与每个被管节点（Chef Node）交互并给出配置指令。

2、Chef Workstation

提供了我们与 Chef Server 交互的接口：我们在 Workstation 上创建定义 Cookbook，并将 Cookbook 上传到 Chef Server 上以保证被管机器能从 Chef Server 上取得最新的配置指令。

3、Chef Node

安装了 chef-client 并注册了的被管理节点，可以是物理机或者虚拟机或者其他对象。Chef Node 每次运行 chef-client 时都会从 Chef Server 端取得最新的配置指令（Cookbook）并按照指令配置自己。一套 Chef 环境包含一个 Chef Server，至少一个 Chef Workstation，以及一到多个 Chef Node。

### Chef 架构说明

一个中心服务器(Chef Server)

Chef将数据存储在CouchDB数据库里面

RabbitMQ和chef-solr等提供搜索的功能

Chef还提供了个图形的用户界面(cher-server-webui)

一至多个Workstation(运行knife工具对Chef进行配置)

Workstation上有一个pem文件,knift利用它作为认证来和chef-server通过REST API进行通信

Workstation将配置(利用Recipe等描述各Client应该如何配置自己)上传到服务器

Workstation和中心服务器可以在同一台机器

一至多个Client（Chef Server管理的节点）

Client上有一个pem文件,chef-client利用它作为认证来和chef-server通过REST API进行通信

当新加一个Client的时候,需要从中心服务器上拷贝validator.pem到新加的Client

它利用这个pem进行注册得到自己的client.pem进行以后的认证Client连到Chef服务器查看如何配置自己,然后进行自我配置

## Chef安装部署

### 1.安装准备

Chef Server

chef-server-core-12.17.33-1.el7.x86\_64.rpmchef-manage-2.5.16-1.el7.x86\_64.rpm

Chef Workstation

chef-workstation-0.1.137-1.el6.x86\_64.rpm

Chef Client

chef-14.2.0-1.el7.x86\_64.rpm

### 2.关闭防火墙、SELinux

# 关闭iptables

chkconfig iptables off

# 关闭ipv6防火墙

chkconfig ip6tables off

#关闭firewall

systemctl stop firewalld.service

systemctl disable firewalld.service

# 关闭SELinux

sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config

### 3.Ntp时间同步

# 不同步会导致，登录认证无法通过

# root/sudo执行

ntpdate ntp.ubuntu.com

设置主机名，域名配置

# Server、Workstation、Nodes均要设置hostname、配置hosts

# 设置Chef Server

hostnamectl set-hostname chefserver

# 设置Workstation

hostnamectl set-hostname chefworkstation

# 设置Node

hostnamectl set-hostname chefnode

# 配置hosts

vim /etc/hosts

192.168.0.31 chefserver

192.168.0.39 chefworkstation

192.168.0.40 chefnode

### 4.Chronyd时间同步

[root@controller ~]# yum -y install chrony  
[root@controller ~]# vim /etc/chrony.conf  
allow 192.168.20.0/24 #打开这行,表示允许谁同步时间  
  
[root@controller ~]# systemctl start chronyd  
[root@controller ~]# systemctl enable chronyd  
[root@controller ~]#chronyc sources

[root@compute ~]# yum install -y chrony  
[root@compute ~]# vim /etc/chrony.conf  
#server 0.centos.pool.ntp.org iburst  
#server 1.centos.pool.ntp.org iburst  
#server 2.centos.pool.ntp.org iburst  
#server 3.centos.pool.ntp.org iburst  
server 192.168.20.11 iburst

#将上面的网络时间服务器注掉,写上这行,表示与controller同步时间

[root@controller ~]# systemctl start chronyd  
[root@controller ~]# systemctl enable chronyd  
[root@controller ~]#chronyc sources

### 5.下载源文件

# 创建目录

mkdir -p /chef/software <https://downloads.chef.io/>

cd /chef/software

# Chef Server [最新稳定版12.17.33]

wget <https://packages.chef.io/files/stable/chef-server/12.17.33/el/7/chef-server-core-12.17.33-1.el7.x86_64.rpm>

# Chef Manage [最新稳定版2.5.16]

wget <https://packages.chef.io/files/stable/chef-manage/2.5.16/el/7/chef-manage-2.5.16-1.el7.x86_64.rpm>

# Chef Workstation [最新稳定版0.1.137]

wget <https://packages.chef.io/files/stable/chef-workstation/0.1.137/el/7/chef-workstation-0.1.137-1.el6.x86_64.rpm>

# Chef Client [最新稳定版14.2.0]

wget <https://packages.chef.io/files/stable/chef/14.2.0/el/7/chef-14.2.0-1.el7.x86_64.rpm>

### 6.系统参数调整

#内核参数调整

vi /etc/sysctl.conf

# 物理内存使用90%再使用swap

vm.swappiness = 10

# 1 表示开启重用，允许将TIME-WAIT sockets重新用于新的TCP连接，默认为0，表示关闭

net.ipv4.tcp\_tw\_reuse = 1

# 1 表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭

net.ipv4.tcp\_tw\_recycle = 1

# 定义了系统中每一个端口最大的监听队列的长度, 对于一个经常处理新连接的高负载web服务环境，默认128太低

net.core.somaxconn = 2048

# 表示如果套接字由本端要求关闭，这个参数决定了它保持在FIN-WAIT-2状态的时间

net.ipv4.tcp\_fin\_timeout = 30

# 该参数决定了, 每个网络接口接收数据包的速率比内核处理这些包的速率快时，允许送到队列的数据包的最大数目，不要设置过大

net.core.netdev\_max\_backlog = 8096

# 控制分配内存行为，不允许overcommit

vm.overcommit\_memory= 2

# 由于系统物理内存和swap内存都为4G

vm.overcommit\_radio=0

# (semmsl semmns semopm semmni)

# semmsl：每个信号量set中信号量最大个数；

# semmns：linux系统中信号量最大个数；

# semopm：semop系统调用允许的信号量最大个数设置，

# 设置成和semmsl一样即可；

# semmni：linux系统信号量set最大个数；

kernel.sem = 500 512000 500 1024

# 生效

sysctl -p

# 关闭允许hugepage可以动态分配

echo never> /sys/kernel/mm/transparent\_hugepage/enabled

# 资源参数调整

# 查看资源参数命令：ulimit -a

#1、open files 调整为65535

ulimit -n 65535

#2、max user processes 调整为 65535

ulimit -u 65535

安装

### 1．Chef Server安装

rpm -ivh chef-server-core-12.17.33-1.el7.x86\_64.rpm

# 2.配置server

chef-server-ctl reconfigure

# 配置成功信息如下

# Starting Chef Client, version 12.21.31

#resolving cookbooks for run list: ["private-chef::default"]

#Synchronizing Cookbooks:

# - private-chef (0.1.1)

# - openssl (8.1.2)

# - runit (4.0.4)

# - packagecloud (0.3.0)

# - yum-epel (2.1.2)

# - compat\_resource (12.19.1)

# - enterprise (0.11.0)

# <...省略若干信息...>

# Chef Client finished, 78/580 resources updated in 57 seconds

# Chef Server Reconfigured!

# 3.安装管理控制台

chef-server-ctl install chef-manage --path /chef/software/chef-manage-2.5.16-1.el7.x86\_64.rpm

# 4.重新配置server

chef-server-ctl reconfigure

# 5.配置manage

# 先按任意键继续再按q键然后输入yes

chef-manage-ctl reconfigure

# 6.创建用户

# 语法：chef-server-ctl user-create USER\_NAME FIRST\_NAME LAST\_NAME EMAIL 'PASSWORD' --filename FILE\_NAME

chef-server-ctl user-create thinkbam Lee PengFei 452734423@qq.com 'thinkbam' --filename /etc/chef/thinkbam.pem

# 7.进入Manage创建组织

# 浏览器输入 https://192.168.0.31

# 创建组织：Thinkbam 简称 ：top

### 2．Chef Workstation安装

# 1.安装

cd /chef/software

rpm -ivh chef-workstation-0.1.137-1.el6.x86\_64.rpm

# 2.验证安装

chef verify

# 3.配置系统默认ruby

echo 'eval "$(chef shell-init bash)"' >> ~/.bash\_profile

source ~/.bash\_profile

which ruby

# 输入一下信息说明安装成功

# /opt/chef-workstation/embedded/bin/ruby

# 4.下载Starter Kit

# 浏览器输入https://192.168.0.31/organizations/top

# 依次点击： "选择top一栏的设置按钮" -> "Starter Kit" -> "Download Starter Kit"

# 5.上传chef-starter.zip(默认已第二种方式，第一种少许多东西)

# mkdir /opt/chef-data

# 上传 chef-starter.zip 至 /opt/chef-data 目录

cd /opt/chef-data

unzip chef-starter.zip

cd chef-repo/

# 6.验证安装

knife ssl fetch

knife ssl check

knife node list

# 5 配置chef-repo(以下为另一种方式)

创建chef-repo

**$** chef generate repo chef-repo

创建.chef目录

**$** mkdir chef-repo/.chef

因为Workstation要与Server交互，需登录Server下载以下文件：

* knife.rb

Administrator > Organizations > Select your organization > Gernerate Knife Config > Save File

* ORGANIZATION-validator.pem

Administrator > Organizations > Select your organization> Reset Validation Key > Download

* USER.pem

Administrator > Users > Select your user name > Reset Key > Download

下载后将这些文件复制到.chef文件夹  
下载证书：

**$** knife ssl fetch

下载的证书保存在.chef/trusted\_certs/目录下。

**验证安装**

执行以下命令查看是否能连接到Chef Server:

**$** cd chef-repo

**$** knife ssl check

### 3．Chef Client安装

# 1.安装

rpm -ivh chef-14.2.0-1.el7.x86\_64.rpm

# 2.校验

chef-client -v

# 输出 Chef: 14.2.0 表示安装成功

# 3.登录Workstation机器，创建新节点（即在目标主机安装 Chef Client 与 Chef Server 相关联）

knife bootstrap 192.168.0.40 --ssh-user root --ssh-password 123456 --node-name chefnode

# 4.登录Server查看节点详情

# https://192.168.0.31 (用户/密码：thinkbam/123456)

安装完成

## Chef之详情

### Chef之使用步骤

ERROR: You must set your EDITOR environment variable or configure your editor via knife.rb #export EDITOR=vim

Chef 环境安装完成以后，我们来看看如何使用这套环境来进行配置管理。总体来说，Chef 的配置过程是：

* 在 Workstation 上定义各个 Chef Client 应该如何配置自己，然后将这些信息上传到 Server 端。
* 每个 Chef Client 连到 Server 查看如何配置自己，然后进行自我配置。

在 Workstation 上使用 Cookbook 来定义配置方法。Cookbook 使用 Ruby 脚本定义对 Chef Client 的各种操作，具体 Cookbook 的写法本文不做叙述。一旦 Cookbook 写好之后，就可以重复使用，可以对多个 Chef Client 进行批量配置。一般从创建 Cookbook 到使用 Cookbook 会包括以下几个过程。

1. 在 Workstation 上创建 Cookbook

使用 knife 命令可以快速创建一个 Cookbook，如：

# knife cookbook create db2

1. 编辑 Cookbook

根据实际需要，编辑 Cookbook 里的 Recipe，可以定义各种对服务器的配置操作，如系统管理，安装软件等。

1. 同步 Cookbook

将在 Workstation 上写的 Cookbook 同步上传到 Server 上，可以通过 upload 命令实现：

# knife cookbook upload db2

此命令将最新的 db2 Cookbook 上传到 Server 端，这样 Client 端就能从 Server 端得到最新的配置指令。

也可以将所有的 Cookbook 都一起上传到 Server 端：

# knife cookbook upload --all

1. 将 Cookbook 添加到要配置的 Node 的 Run List 中，如：

# knife node run\_list add chef-node2 recipe[db2]

此命令将名为 db2 的 Cookbook 下的默认 Recipe（default.rb）添加到名为 chef-node2 的 Node 的 run\_list 中。

也可以指定 Cookbook 里的某个特定 Recipe 添加，如：

# knife node run\_list add chef-node2 recipe[db2::createdb]

此命令就将特定的 Recipe(created.rb)添加到 Node 的 Run List 里。可以通过 knife 的 node show 命令查看某个 Node 的具体信息：

# knife node show chef-node2

此命令可以看到 Node 的 Run List。

查看更详细的 Node 信息可以加上-l 参数：

# knife node show –l chef-node2

1. 运行 Cookbook

在 Chef Node 上直接运行chef-client命令，Chef Client 就会从 Server 端下载最新的配置脚本，然后按照配置脚本配置自己（即脚本运行的过程）。

# knife ssh "name:\*" 'chef-client' --ssh-user root --ssh-password 1

除掉在 Client 端直接运行chef-client命令，也可以在 Workstation 上运行 knife ssh 命令来达到同样的效果。不同的是，在 Client 端运行 chef-client 命令只是对自己一个 Node 进行配置，而在 Workstation 上运行的 knife ssh 命令可以同时对多个 Client 端进行批量配置。

## Chef之CookBook

### Cookbook\_file

### File

## Ruby语法和案例

语法检查：ruby -c /opt/file

### 注释

注释使用”#”，在需要用的地方可以加入转义”\”，添加注释注明有助于阅读

#jake每天早上要吃5片培根

if bacon.strips < 5

buy\_bacon

end

### 变量