**CHEF**

* **What is chef?**

Chef is a configuration management tool...

* **What is configuration management tool?**

It is maintain number (100 and 1000's) of machines with automation

Infrastructure as a code

* **What it maintain?**
* It maintain the system resources such as
* Install an application
* Configuration
* Updates
* Maintain and so on...
* **Why configuration management tool?**
* Reduce management complexity
* Save time
* **Configuration management tools**
* Puppet---->2005 pull
* Chef------>2008 pull
* Ansible---->2011 push

**Infrastructure as a code**

* Chef insures each node compiles with policy.
* Policy determined by the configurations in each nodes runlist.
* Reduce management complexity through abstraction.
* Your policy states what state each resource should be in.
* **What is resource?**

Resource is a piece of system and its desired state.

File--->create, delete

User---->Crete, delete

Package-->install/not

* **Managing complexity**

1. Organization
2. Environment
3. Roles
4. Nodes
5. Recipes
6. Cookbooks
7. Runlist
8. **Organization:**

It is independent tenant.

**e.g.:** companies, department.

1. **Environment**

Environment reflect your patterns and workflow, and can be used to model the life stages of your applications.

* 1. Development env
  2. Testing env
  3. Staging env
  4. Production env

1. **Roles:**

Roles represents the type of servers in your infrasture.

* 1. DB server
  2. Webserver
  3. Lb. server

1. **Nodes:**

Nodes represents the servers in your infrasture, it could be a physical server or virtual server.

Could also be network hardware’s, switches, routes, etc...

1. **Recipe:**

Recipe is a collection of resources

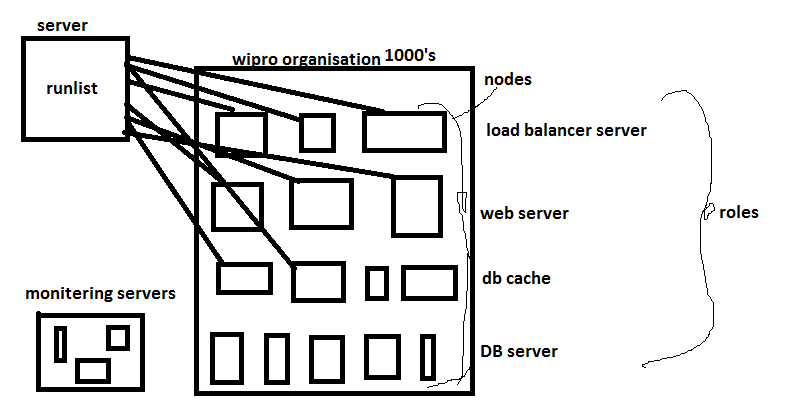
File, application

1. **Cookbooks:**

Collection of recipes and templates and custom resources...

1. **Runlist**

Runlist is a collection of cookbooks, and every machine has its own runlist...



Wipro--organization

Production/testing --environment

Roles --servers like dB, database

Nodes --hostmechines

Recipes --collection of resources like php apache /install

Cookbooks--collection of recopies ex: file creation, application install

Runlist -- collection of cookbooks

* **Advantages:**

1. More scalability
2. Fully automated

* **Dis advantage**

1. Hard to learn

* **chef architecture:**

3 -tier architecture

Workstation------->chef-server<----------nodes

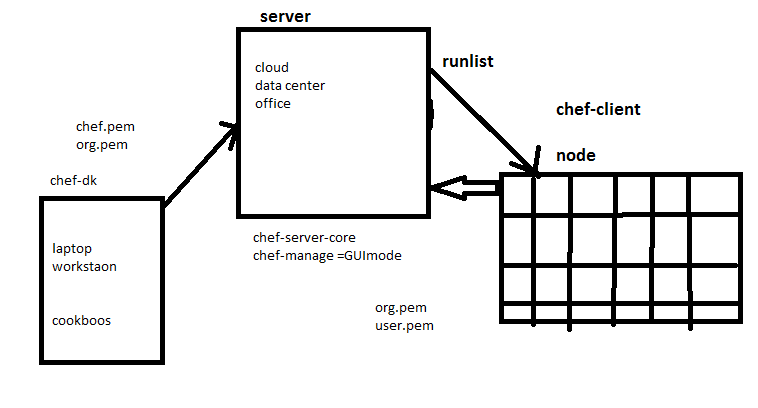
Webserver

DB server

Cache sever

Monitoring server

Email server



* **Chef-server setup**

**Pre-requisite:**

**Step-1**

1. Vm (1 GB ram, 15hdd.1cpu, 64bit Linux)

2. Packages

Chef-server-core (core)

Chef-manage (GUI)

**Step-2:**

Change host name

**Centos6.x centos 7.x**

/etc/sysconfig/network sudo vi /etc/hostname

chefserver.technical.com (FQDN)

**Step-3**

Update the host file

# sudo vi /etc/hosts

**Edit**: ipaddress chefserver hostname

Ip hostname (fullqualified-domain-name)

**Step-4**

* Download above packages
* Bourse: CHEF server DOWNLOAD
  + Copy URL
  + Chef manage
  + Copy URL

**Step-5**

Install the above packages

**#wget copy URL**

**#rpm-ivh chef-server-core...**

**#rpm-ivh chef-manage....**

**To configure**

**#chef-server-ctl reconfigure**

**#chef-manage-ctl reconfigure --accept-license**

**#chef-server-ctl status**

**#chef-server-ctl restart**

**Step**-6

#stop firewall (or) open port 443

Go to browser https:// <chef server ip>

400 error= server is stopped

502 error =request is not reached to server i.e. restart the server

**Step-7**

How to create USER AND organization?

**User:**

#Chef-server-ctl user-create <username> <first name> <middle name (optional)> <last name> "<user Email address>" "<password>" -f <username. Pem>

**Pem---> private enhanced mail.**

**Organization:**

#chef-server-ctl org-create <org short name> "<org full name>" -a <username> -f <orgshort-validator.pem>

* **Chef -workstation**

Steps to install and configuration of chef workstation:

**Step-1**

Get the new vm (1 GB ram, 1cpu, 10 GB Hdd)

**Step-2**

Download and install chefdk (workstation)

**Step-3**

Setup appropriate hostname

Chef-workstation.technicaladda.com

**Step-4**

Update/etc/host file = VI /etc/hosts

<Serverip> <server hostname>

<Workstation ip> <workstation host name>

**Step-5**

Create a user for system with root privileges

Login to that user

Create project directory

#mkdir <techadda>

#cd techadda

**Step-6**

Create a structure and configure "knife.rb" file

#mkdir .chef

#cd .chef

#knife configure

**Step-7**

# knife ssl fetch (fetch the ssl certificate from chef server)

#Knife ssl check (check connection)

**Step-8**

# knife user list

#knife client list

* **Chef-workstation setup in windows (optional)**

Chef server --> 1.5 G

Chef workstation ----> 1G

Chef node ----> 1G approx.

**Linux ---> hostname, hosts**

**Windows --> C:\Windows\System32\drivers\etc\hosts**

* **Chef-node setup using bootstrap method:**
* **Bootstrap process:**

1) It will install chef-client in the node.

2) It will collect the info about node and upload to the chef-server.

3) It will register the node number under chef-server.

4) It will run initial cookbooks (if any).

* **Pre-requisites:**

1) Install a new system (VM).

2) Create a user with sudo (root) access. (Wheel group)

# usermod –aG wheel username

3) Setup the hostname correctly (/etc/hostname). (e.g.: chefnode.techadda.com)

#sudo vi /etc/hostname

4) update the hosts (/etc/hosts) file with chef-server ip, workstation ip and node ip.

#sudo vi /etc/hosts

<Server ip> <server hostname>

<Workstation ip> <workstation hostname>

<node ip> <node hostname>

#ping

**NOTE:** Date and Time must be the same in all the 3 machines.

Now login into workstation and update hosts file with node ip

Windows --> C:\Windows\System32\drivers\etc\hosts

<Node ip> <node hostname>

* **Command for chef-node setup:**

**(From workstation)**

- Knife bootstrap <nodeip> -x <username> -P <password> --sudo

- Knife bootstrap 192.168.43.38 -x chefadmin -P chefadmin --sudo

* **Cookbooks**

**Q) What is cookbook?**

Cookbook is a collection of recipes. Cookbook contain recipes, templates, files, custom resources, etc

**1) Create a cookbook**

**2) Write a recipe file.**

**3) Test a cook book.**

**4) Upload cookbook to server.**

**5) Update node runlist.**

**6) Execute chef-client on the node.**

**Q) What are the commands to create a cookbook?**

# knfe cookbook create <cookbook name> [old version]

It will create full directory structure. It is not associated with git.

It will create cookbook in cookbooks directory.

#chef generate cookbook <cookbook name> [new version]

It will create partial directory structure. It is associate with git.

It will create cookbook in current directory. So, you need to go to cookbooks directory.

* **Old version**

attributes

changelog.md

definations

files

|\_

default

libraries

metadata.rb

README.md

recipes

|\_

default.rb

resources

templates

|\_default

* **New version**

Berksfile

chefignore

metadata.rb

README.md

recipes

|\_\_\_default.rb

spec

|\_\_\_spec\_helper.rb

|\_\_\_unit

|\_\_recipes

|\_\_default.spec.rb

test

|\_recipes

|\_default-test.rb

**1) File\_test**

1) Create a cookbook

2) Write a recipe file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#knife cookbook create file\_test (or) chef generate cookbook file\_test (new version)

#VI techadda/cookbooks/file\_test/recipes/default.rb

resource\_type 'name' do

Parameters

Parameters

Action

End

file '/tmp/dummy.txt' do

owner 'root'

group 'root'

mode 0644

content 'this is a dummy file'

action :create

end

#knife cookbook test file\_test (or) cookstyle test file\_test

# knife cookbook upload file\_test

#knife cookbook list

# knife node run\_list add <node name> file\_test

#knife node show <node name> to display node information

# execute "chef-client" in chef node

#sudo chef-client

#cat dummy.txt

**Ohai process:**

It collects the data about your system (nodes) and upload to the server

#ohai | less

**2) Cookbook for webserver (APACHE):**

**Web:**

1) Install the package (httpd)

2) Start the httpd service

3) Enable httpd service

4) Create index.html page

5) Check and confirm.

* **Cookbook steps:**

1) Create a cookbook.

2) Write a recipe file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#knife cookbook create web\_server (or) chef generate cookbook web\_server

# vi techadda/cookbooks/web\_server/recipes/default.rb

package 'httpd' do

action: install

end

service 'httpd' do

action [: start , :enable]

end

file '/var/www/html/index.html' do

owner 'root'

group 'root'

mode 0644

content '<h1>hello technical guys welcome</h1>'

action: create

end

#knife cookbook test web\_server (or) cookstyle test web\_server

#knife cookbook upload web\_server

#knife cookbook list

#knife node run\_list add <node name> web\_server

#execute "chef-client" in chef node

**3) Cookbook for cronie:**

**Q) What is cronjob?**

We can schedule the activities in the machine.

E.g.: backup script, commands, tasks.

**#crontab -e** -->edit the cron schedule

**# crontab -l** -->to see the current schedule

package : cronie

service: crond

file: /var/spool/cron/<username>

There are six fields in this file

MIN HR DOM MONTH DOW <command>

(0-59) (0-23) (1-31) (1-12) (0-7) command

30 01 \* \* \* /home/ramki/etcbackup.sh (daily backup)

00 02 \* \* 1 /home/ramki/backup.sh (weekly backup)

00 03 01 \* \* /home/ramki/backup.sh (monthly backup)

(9,19,29,39,49,59) \* \* \* \* /home/ramki/backup.sh (every 9,19,29,39,49,59 min backup)

\*/10 \* \* \* \* /home/ramki/backup.sh (every 10 min)

**Cookbook steps:**

1) Create a cookbook.

2) Write a recipe file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#Knife cookbook create cron\_test (or) chef generate cookbook cron\_test

#vi techadda/cookbooks/cron\_test/recipes/default.rb

Package 'cronie' do

action: install

end

service 'crond' do

action [: start, :enable]

end

file '/var/spool/cron/root' do

owner 'root'

group 'root'

mode 0644

content '\*/10 \* \* \* \* /bin/chef-client'

action :create

end

#knife cookbook test cron\_test (or) cookstyle test cron\_test

#knife cookbook upload cron\_test

#knife cookbook list

#knife node run\_list add <node name> cron\_test

#execute "chef-client" in chef node

**4) Chef Attributes:**

Chef attributes are variables, using this attributes we can find below info.

In short, it will help us to get the information about node and it's state.

E.g.: RAM, hostname, ipaddress, packages.

**Centos, ubuntu**

Two types of attributes are their

i) User defined -->user will define it

ii) System defined --> chef will define it

**Q) Where you can define the attributes?**

i) In the node itself.

ii) In the cookbook (attributes, recipe)

iii) In the role.

iV) In the environment.

**Q) What are different types of attributes?**

* 1. Default ->lowest precedence
  2. Force\_default
  3. Normal
  4. Override
  5. Force\_override
  6. Automatic -> highest precedence

**Use less definition in recipe file.**

**E.g.:** package\_name = "httpd"

Service \_name = "httpd"

package package\_name do

action :install

end

* **attributes in cookbook:**

We can define attributes in attributes directory, and recipe directory

default ["package\_name"] = "cronie" node ["package\_name"]

1) Create a cookbook.

2) Write a recipe file and attribute file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#knife cookbook create attribute\_test (or) chef generate cookbook attribute\_test

# mkdir techadda/cookbooks/attribute\_test/attributes

# vi techadda/cookbooks/attribute\_test/attributes/default.rb

case node['platform’]

when "centos","rhel"

default['pkg\_name']="httpd"

default["ser\_name"] = "httpd"

default["doc\_root"] = "/var/www/html"

when "ubuntu","debian"

default["pkg\_name"] = "apache2"

default["ser\_name"] = "apache2"

default["doc\_root"] = "/var/www/"

end

#vi techadda/cookbooks/attribute\_test/recipes/default.rb

package node['pkg\_name'] do

action :install

end

service node['ser\_name'] do

action [:start, :enable]

end

file "#{node['doc\_root']}/welcome.html" do

owner 'root'

group 'root'

mode '0644'

content 'hey guys'

action :create

end

#knife cookbook test attribute\_test (or) cookstyle test attribute\_test

#knife cookbook upload attribute\_test

#knife cookbook list

#knife node run\_list add <node name> attribute\_test

#execute "chef-client" in chef node

**5) Templates:**

Q)what is template?

Template is a script with .erb extension

.erb --> embedded ruby

Using this script we can create a static files.

Source destination

----------------------------------

index.html.erb ---->index.html

httpd.conf.erb ---->httpd.conf

1) Create a cookbook.

2) Write a recipe file and template file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#knife cookbook create template test (or) chef generate cookbook template\_test

# mkdir techadda/cookbooks/template\_test/templates/default/

# vi techadda/cookbooks/attribute\_test/templates/default/index.html.erb

<html>

<h1>welcome to technical adda my hostname is <%=node['hostname']%></h1>

<h2>my server platform is <%=node['platform']%></h2>

<h3>server has total memory <%=node['memory']['total']%> </h3>

#vi techadda/cookbooks/template\_test/recipes/default.rb

#template "/var/www/html/info.html" do

owner 'root'

group 'root'

mode '0644'

source 'index.html.erb'

action :create

end

#knife cookbook test template\_test (or) cookstyle test template\_test

#knife cookbook upload template\_test

#knife cookbook list

#knife node run\_list add <node name> template\_test

#execute "chef-client" in chef node

**6) Cookbook for creating user:**

**Q) Who is user?**

Who can utilize the system resources?

**Target:** create a user in chef node.

Information need to create a user

username: user01

password : <password>

groupname : <group name>

home : /home/<username>

shell : /bin/bash

**Q) What is my password structure?**

**Password layers:**

$encrypt\_type$salt\_name$actual\_pass

$encrypt\_type = 6 types of sha1

md5=1

crc=2

sha256=5

sha512=6

$salt\_name=$kgh556 (random unique number system generated)

$actual\_pass=$8md5nvef985

**Q) How to generate an encrypted password?**

#openssl passwd -<type> -salt <salt name> <actual\_passwd>

# openssl passwd -1 -salt batman redhat

1) Create a cookbook.

2) Write a recipe file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#knife cookbook create user\_test (or) chef generate cookbook user\_test

#vi techadda/cookbooks/template\_test/recipes/default.rb

group 'user01' do

action :create

end

user 'user01' do

password '$1$batman$nev555'

group 'user01'

shell '/bin/bash'

home '/home/user01'

manage\_home true

action :create

end

#knife cookbook test user\_test (or) cookstyle test user\_test

#knife cookbook upload user\_test

#knife cookbook list

#knife node run\_list add (node name) user\_test (or) knife node edit (node name)

#execute "chef-client" in chef node

**6) Databags:**

Data bags are used to maintain the system sensitive information like passwords.

**Q) What are databags?**

- Databags are global variables.

- It stores the data in JSON format.

- It will always store on the server.

- It stored in index format for faster searching.

**Q) Why we need databags?**

- To store secret (or) sensitive information like password, private key.

**Q) What are the steps to handle databags?**

a) Create an empty databags.

b) Write a son format file and put some data.

c) Import the data from the file to databags.

d) See the list of current databags.

e) Modify the recipe to collect data from databags.

#Knife data bag list

#knife data bag create (data bag name)

#vi user02\_password.json

{

"id" : "user02",

"password" : "$1$batman$254niefb"

}

#knife data bag from file (data bag name) (file.json)

**Cookbook steps:**

1)create a cookbook.

2) Write a recipe file.

3) Test a cook book.

4) Upload cookbook to server.

5) Update node run\_list.

6) Execute chef-client on the node.

#knife cookbook create user\_test (or) chef generate cookbook user\_test

#vi techadda/cookbooks/template\_test/recipes/default.rb

userpass=data\_bag\_item('user02\_passwd','user02')

group 'user02' do

action :create

end

user 'user02' do

password userpass['password']

group 'user02'

shell '/bin/bash'

home '/home/user02'

manage\_home true

action :create

end

#knife cookbook test user\_test (or) cookstyle test user\_test

#knife cookbook upload user\_test

#knife cookbook list

#knife node run\_list add (node name) user\_test (or) knife node edit (node name)

#execute "chef-client" in chef node

* **Roles in chef:**

**Q) What is a role?**

It is a type of server like db, web, and application.

It is a collection of cookbooks and attributes.

* **Commands to create a role**

#knife role create webserver

#knife role edit webserver (place the cookbooks in role)

#knife node edit chefnode.techadda.com (instead of cookbooks place the role role[role\_name])

#knife[:editor]="vim"

#Apache, php, mysql (LAMP )

**Supermarket in chef:**

#knife supermarket --help

# knife supermarket download mysql 8.5.1