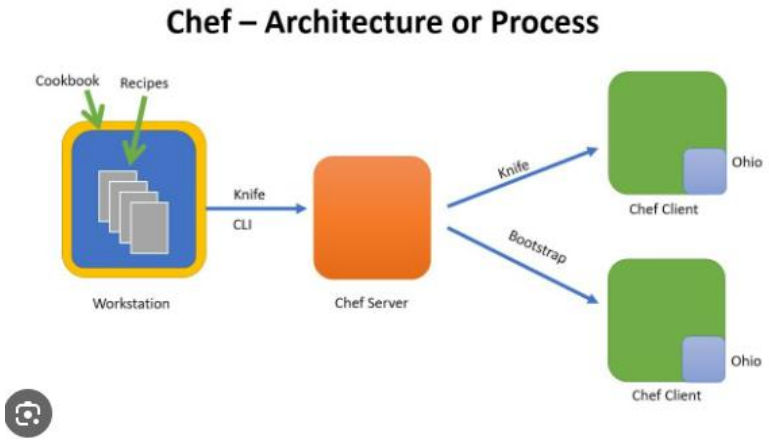
CHEF

Chef is a configuration management tool.

It is a pull based configuration management tool unlike Ansible which is push based in nature.

It turns our code into Infrastructure hence, its Infrastructure As Code[IAC].

Chef Architecture



Chef has following components.

(a)-Workstation-They are personal machine or virtual servers , where configuration codes are created,Codes are called Recipes and collection of recipes is called cookbook.

Workstation communicate with Chef Server through Knife.

(b)Chef-Server-It is stored between Workstation and Node.

All Cookbooks are stored here. Servers may be stored locally or remote.

(c) Nodes-They are systems which require configuration.

OHAI fetches current state of node.

Node communicate with Chef Server with help of Chef-Client

CHEF

Download & install **Chef** and create **Cookbook, Recipes**

- Wget <chef download link>[Install Chef Workstation for Linux]
- mkdir cookbooks
- cd cookbooks/
- chef generate cookbook **Ashok**-cookbook
- cd **Ashok**-cookbook
- yum install tree -y
- tree
- chef generate recipe **Ashok**-recipe **cd ..**
- vi **Ashok-cookbook/recipes/test-recipe.rb**

I + Enter then <paste code>

```
file '/myfile' do
content  'Welcome
Ashok     Anupam'
action
:create end
```

enter+esc+:wq

Chef exec ruby -c Ashok-cookbook/recipes/Ashok-recipe.rb (check the syntax)

- Syntax **OK**
- (run the chef client)
- **Chef-client -zr "recipe[Ashok-cookbook::Ashok-recipe]"**
- **Cat /**myfile(xyz) (also try **ls /**) (to check inside the file)

Apache server:

[cookbooks]#chef generate cookbook apache-cookbook
#cd apache-cookbook
#chef generate recipe apache-recipe #cd
..
#vi Apache-cookbook/recipes/apache-recipe.rb

I + Enter then <paste code>

```
package 'httpd'
do action :install end

file '/var/www/html/index.html' do
content 'Welcome Ashok Anupam'
action :create end

service 'httpd' do  action
[:enable, :start] end
```

#Chef-client -zr "recipe[Apache-cookbook::Apache-recipe]"

CHEF

Now ping public IP address to see content on apache website

ATTRIBUTES:

What is this: Attributes is a key value pair which represent a specific detail about node.

Who used? Chef client

Why used? To determine

- current state of node
- what was the state of the node at the end of previous chef client run?
- What should be the state of the node at the end of current chef client will run?

Types:	**Priority**
1. Default	1st maximum
2. Force-default	2nd more
3. Normal	3rd may be
4. Override	4th less
5. Force override	5th very less
6. Automatic	6th minimum

Who defines Attributes?

Ans: (Node, Cookbooks, Roles, Environment**)**

**(attribute defined by Ohai have highest priority)

sudo su
ohai
ohai ipaddress
ohai memory/total # ohai cpu/0/mhz
ls
cd cookbooks
cd Apache-cookbook
Chef generate recipe recipe10 # cd ..
vi **apache-cookbook/recipes/recipe10.rb**

[+ Enter then <paste code>]

File '/besicinfo' do Content "this id to get Attributes HOSTNAME: #{node['hostname']} IPADDRESS: #{node['ipaddress']} CPU: #{node['cpu']['0']['mhz']} MEMORY: #{node['memory']['total']}" owner 'root' group 'root' action :create end Esc+:wq

chef exec ruby -c apache-cookbook/recipes/recipe10.rb

chef-client -zr "recipe[apache-cookbook::recipe10]" (call the client)

SEE OUTPUT ATTRIBUTES

Insert Linux commands

[cookbooks]# vi Ashok-cookbook/recipes/ABC-recipe.rb
I + Enter then <paste code>

Execute "run a script" do Command <<-EOH ----> EOH = End of here/hunk (now can write non ruby)
Mkdir /Ashokdir touch /Anupamfile EOH End Enter+Esc+:wq

CHEF

Create user

vi Ashok-cookbook/recipes/ABC-recipe.rb

User "Anupam" do Action :create End Enter+Esc+:wq

(now run the recipe,call chef-client)

chef-client -zr "recipe[Ashok-cookbook::ABC-recipe]"

Create group

vi Ashok-cookbook/recipes/ABC-recipe.rb

Group "Devops group" do Action :create Members 'Alok' Append true End Enter+Esc+:wq (now check the recipe)

Chef exec ruby -c Ashok-cookbook/recipes/ABC-recipe.rb
syntax OK (now run the recipe)
chef-client -zr "recipe[Ashok-cookbook::abc-recipe]"
cat /etc/group (also try ls /) (to check the group)

RUNLIST: To run the recipe in a sequence order , we mention that in a run list. With this process we can run multiple recipes but the condition is, they must be from one cookbook.

***(chef client calling default recipes from Ashok-cookbook & Apache-cookbook together) ***

[cookbooks]# chef-client -zr "recipe[Ashok-cookbook::default],recipe[Apache-cookbook::default]"

Include Recipe: To call recipes/recipe from another recipe with in same cookbook.

To run multiple recipes from same cookbook. We can run any numbers of recipes with include command but all must be from same cookbook. Here including recipes with default recipe in Ashok-Cookbook.

[cookbooks]# vi Ashok-cookbook/recipes/default.rb then
<paste code>

Include_recipe "ABC-cookbook::ABC-recipe"
Include_recipe "ABC-cookbook::XYZ-recipe"
Include_recipe "ABC-cookbook::PQR-recipe"Esc+:wq (call the chef client)

#chef-client -zr "recipe[Ashok-cookbook::default]"

BOOTSTRAP

Connect workstation to chef server to node using chef-repo, bootstrap

Chef server is going to be mediator between the code and cookbooks.

Bootstrapping is Attaching a node to chef server, while using Bootstrap command, both workstation and node should be in same AZ.

CHEF

Two actions will be done while bootstrapping 1. adding node to chef server 2. installing chef package.

Chef-repo It would be the main directory inside it you have to run any commands, it is also having cookbooks).

Create chef manage account by “manage.chef.io” and download starter kit. Go to download and extract file chef-repo, after extracting we got more files inside chef-repo are (.chef ,cookbooks ,gitignore, README.md, roles)

For sending chef-repo file to Linux machine we use the software called WinSCP. Drag Chef-repo from left window and drop

Sudo su

ls

chef-repo chef-workstation-20.7.96-1.e17.x86_64.rpm cookbooks nodes

cd chef-repo

ls-a

. .. .chef cookbooks .gitignore README.md roles

cd .chef/

[.chef]# ls/

config.rb kritikaji.pem (organization name is kritikaji)

cat config.rb

Inside config.rb you will get chef_server_url https://api.chef.io/organizations/kritikaji

knife ssl check (to check workstation is connected with chef server ?)

Create Linux machine (Node1) same AZ of workstation with new security group and new key pair name

node1-key, save Private IP for further knife bootstrap commands.(SSH & HTTPs)

Attach Advance details

[#!/bin/bash
Sudo su
Yum update -y]

With the help of WinSCP please transfer downloaded node1-key.pem to Chef-repo for bootstrap command

Now go to chef workstation and execute Bootstrap command to attach node1 to chef-server.

[chef-repo] # **knife bootstrap 172.31.44.20 --ssh-user ec2-user --sudo -i node1.pem -N Node1**
(Y for YES/NO)

CHEF

Node has been connected to server and node package has installed

```
[chef-repo] # knife node list

node1 ❼(showing result node1 means node1 has been connected to server)
```

Moving and delete cookbooks in chef-repo to avoid cookbooks confusion:

mv cookbooks/apache-cookbook chef-repo/cookbooks

mv cookbooks/Ashok-cookbook chef-repo/cookbooks

ls [didn't get any cookbook, all empty]

cd chef-repo

ls [get (cookbooks node1-key.pem README.md roles)]

apache-cookbook ,chefignore starter, Ashok-cookbook (got inside the chef-repo-cookboook)

It means both cookbooks have been moved to Chef-repo Cookboooks from

Upload apache-cookbook to chef-server:

[Chef-repo] # knife cookbook upload apache-cookbook

knife cookbook list (confirm uploading?)

apache-cookbook

Now we will attach the recipe on node1 which we would like to run on node1, by this command

[chef-repo]# knife Node1:node set node1 "recipe[apache-cookbook::apache-recipe] run_list: recipe[apachecookbook::apache-recipe]

[chef-repo] # knife node show node1 (get so many info including recipes in run_list)

Now access the Node1

sudo su
chef-client

This chef-client implement the code (inside the recipe) on server Automatically Now back to workstation and edit the recipe:

[Chef-repo]# vi cookbooks/apache-cookbook/recipes/apache-recipe.rb

CHEF

"Update recipe"

Upload apache-cookbook to chef-server

[chef-repo] # knife cookbook upload apache-cookbook

Now go to the node1 and call chef client # chef-client
You can see all updated content, also you can ping node1's public IP and see change.

Now see how can we automate this process:
Go to node1
[ec2-user] # vi /etc/crontab
* * * * * root chef-client Esc+:wq */n (HR DAY MONTH YEAR WEEK)

With the help of this command automation will start no need to call the chef client again=2 Chef-client command execute periodically according to "*/n * * * * crontab method"

Now see full automation:

Create one more linux machine Node2 *(we also can use existing key of node1 for node2 creation)

Attach Advance details [#!/bin/bash
Sudo su Yum
update -y
echo"* * * * *root chef-client">> etc/crontab]

Now back to workstation and run Bootstrap command
[chef-repo]# knife bootstrap 172.31.10.120 --ssh-user ec2-user --sudo -i node- 2key.pem -N node2 (Y for YES/NO)

Node has been connected to server and node package has been installed

Now Attach the Recipe to node2 run_list

[chef-repo]# knife node run_list set node2 "recipe[apache-cookbook::apache-recipe]"

then for check ping the IP of node2 and see webpage.

How to see Delete everything from inside chef-server:

To see list of client present inside chef-server,To delete clients

[chef-repo] #knife client list

knife client delete clientname -y

To see cookbook list,To delete cookbook

[chef-repo] # knife cookbook list

#knife cookbook delete cookbookName -y

CHEF

To see Role list To delete Role

[chef-repo] # knife role list

#knife cookbook delete roleName -y

To see Node list,To delete Node

[chef-repo] # knife node list

#knife cookbook delete nodeName -y

How to create ROLE:

[chef-repo]# ls
.chef roles
cd roles/
[roles] #ls #starter.rb
[roles] # vi Engineer.rb *(this is the command to create role name Engineer)
Name "Engineer"
Description "webserver role"
run_list "recipe[apache-cookbook::apache-recipe]" ESC+:wq

Now back to chef-repo # cd .. and upload the role on chef server

[chef-repo] # knife role from file roles/Engineer.rb

If you want to see the created role

knife role list

o/p: Engineer

Now create 4 instances (1,2,3,4) by one IMA on same availability zone as of workstation with new security group sg-1 with SSH +HTTP.

Attach Advance details [#!/bin/bash
Sudo su Yum
update -y
echo"* * * * *root chef-client">> etc/crontab]

Now Bootstraps the nodes 1,2,3,4 one by one

#[chef-repo]# knife bootstrap 172.31.10.121 --ssh-user ec2-user --sudo -i node-1key.pem -N node1 #[chef-repo]# knife bootstrap 172.31.10.122 --ssh-user ec2-user --sudo -i node-1key.pem -N node2

#[chef-repo]# knife bootstrap 172.31.10.123 --ssh-user ec2-user --sudo -i node-1key.pem -N node3

CHEF

#[chef-repo]# knife bootstrap 172.31.10.124 --ssh-user ec2-user --sudo -i node-1key.pem -N node4

Now connect these nodes to roles one by one.

knife node run-list set node1 "role[Engineer]"
Node1:Run_list:role[Engineer]
(similarly for rest 3 nodes)

knife node run-list set node2 "role[Engineer]"
knife node run-list set node3 "role[Engineer]"
knife node run-list set node4 "role[Engineer]"

UPLOAD cookbook to server

knife cookbook upload apache-cookbook

Now we can check public IP of any node on webserver, every node will behave like server cause, now cookbook has been uploaded despite of uploading different recipes, all recipes have uploaded together inside role by cookbok.

Now we are doing changes in recipe

vi cookbooks/apache-cookbook/recipes/apache-recipe.rb

Content change to "I Love India" ESC+:wq

Now see if Boss need changes, said do work on another recipe (recipe10)

#cat cookbooks/apache-cookbook/recipes/recipe10.rb

Paste code update recipe and go to the role in workstation

vi roles/Engineer.rb

vi Engineer.rb
Name "Engineer"
Description "webserver role" run_list "recipe[apache-cookbook::apache-recipe]" update apache-recipe to recipe10 in role run_list "recipe[apache-cookbook::recipe10]"
ESC+:wq

*for update in recipe we can create user and file by these commands below
#user"Ashok"
#file "Anupamfile"

now upload role to server
[chef-repo] # knife role from file roles/Engineer.rb
Again, go to the workstation

vi roles/Engineer.rb
Name "Engineer"

CHEF

Description "webserver role" (change last line only apache-cookbook in role) run_list "recipe[apache-cookbook]" ESC+:wq

now upload role to server

[chef-repo] # knife role from file roles/Engineer.rb

Do not mention any recipe just upload only cookbook for all recipes, will update automatically on server

knife cookbook upload apache-cookbook

Now we are adding 2 cookbooks in roles

vi roles/Engineer.rb

Name "Engineer"
Description "webserver role"
run_list "recipe[apache-cookbook]","recipe[Ashok-cookbook]" esc+:wq

now upload role to server

[chef-repo] # knife role from file roles/Engineer.rb

Do not forget to upload Ashok-cookbook on server otherwise role will not perform properly

knife cookbook upload Ashok-cookbook

Boss need changes again but this time in Ashok-recipe

Chef-repo]# vi cookbooks/apache-cookbook/recipes/Ashok-recipe.rb

%W (httpd mariadb-server unzip git vim) .each do |p| Package p do
Action :install end
end esc+:wq
knife cookbook upload Ashok-cookbook

Go to inside any node and search git by using command

which git after 1 minute execute again same command and you will see output

/bin/git it means working properly