

3 things in chef: Files (create, delete, update), Packages(install, update, remove), Services(start, stop)

Two important components of Node: Chef Client & Chef Ohai.

Commands:

sudo su

yum update -y

mkdir cookbook

wget https://packages.chef.io/files/stable/chef-workstation/21.10.640/el/8/chef-workstation-21.10.640-1.el8.x86_64.rpm

yum install -y chef-workstation-21.10.640-1.el8.x86_64.rpm

cd cookbook

chef generate cookbook mycookbook

cd mycookbook

chef generate recipe mycookbook/recipes/recipe2 / chef generate recipe test-recipe

cd ..

vi mycookbook/recipes/recipe2.rb

```
file '/myfile' do
  content 'Hello Welcome to DevOps tool CHEF'
  action :create
end
```

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

file '/myfile2' do
  content 'This is My Second Project code'
  action :create
  owner 'root'
  group 'root'
end
```

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

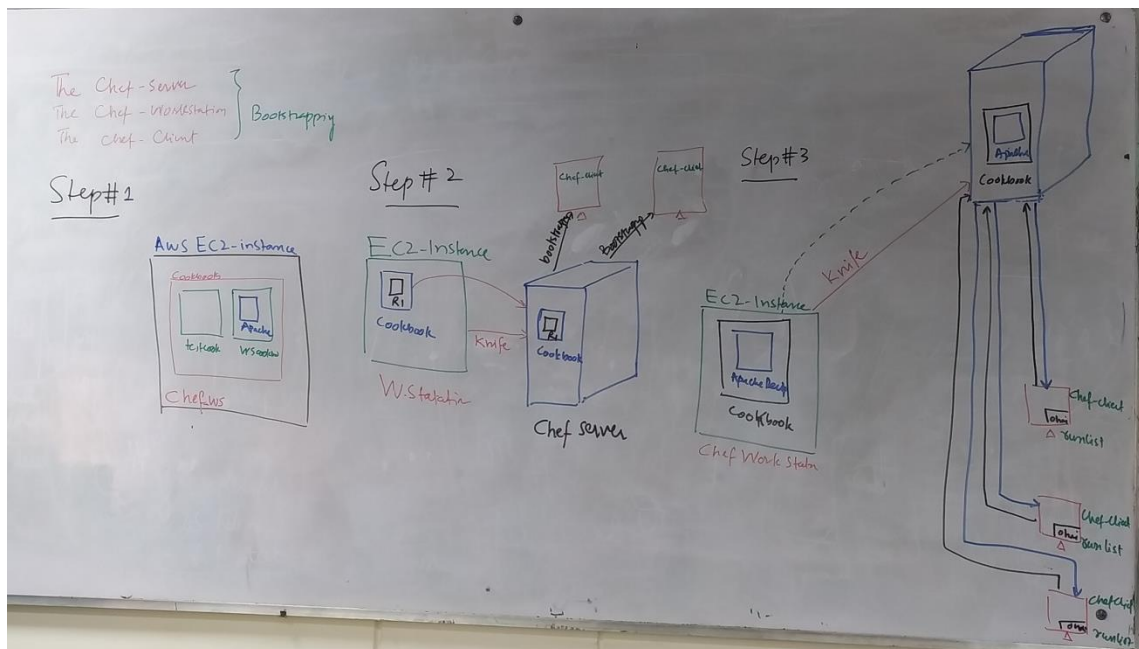
file '/var/www/html/index.html' do
```

```
content 'Welcome to Devops, I m Jonny'
action :create
end
```

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

chef exec ruby -c mycookbook/recipes/test-recipe.rb // For Syntax check

chef-client -zr "recipe[mycookbook::test-recipe]" // For execution



Procedure (for Step 2 & 3):

1. Create an account in Chef Server(chef.manage.io).
2. Attach your Workstation to Chef Server
3. Attach nodes to Chef Server via Bootstrapping. (So that Nodes automatically gets updated from Chef Server)
4. Apply Cookbooks from Server to Node. (auto)

Commands:

Step # 01:

```
Sudo su
Ls
Cd cookbooks
```

Chef generate cookbook apache-cookbook

Ls

cd apache-cookbook

chef generate recipe apache-recipe

vi recipes/apache-recipe.rb

```
package 'httpd' do
```

```
  action :install
```

```
end
```

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

```
  content 'Welcome to Devops, I m Jonny'
```

```
  action :create
```

```
end
```

```
service 'httpd' do
```

```
  action [:enable, :start]
```

```
end
```

chef exec ruby -c recipes/apacherecipe.rb

chef-client -zr "recipe[apache::apacherecipe]"

Copy Public IP of your instance & open it in browser, hopefully this will act as apache server now.

Step # 02:

Sudo su

Ls

Cd cookbooks

Now open Browser search for chef.manage.io & create an account.

We make Chef Cloud as Chef Server.

Go to Chef account, create an organization, click on starter kit & download starter kit, unzip it & find chef-repo in it(We have to upload it to EC-2 Instance to establish connection).

Now download WinSCP software & login with ec-2 credentials & drag & drop the chef-repo folder to Linux.

WinSCP Software Details:

Host name: Public IPv4 DNS of your Instance

Username: ec2-user

Advanced -> authentication -> Upload Private Key File(.ppk) of your instance Key-pair.

Login now.

Drag & Drop Chef repo file from Windows to Linux & close it.

Ls

Cd chef-repo

Cd .chef

Cat config.rb

See https://docs.chef.io/workstation/config_rb/ for more information on knife configuration options

```
current_dir = File.dirname(__FILE__)
log_level      :info
log_location   STDOUT
node_name      "johnnyaslam"
client_key      "#{current_dir}/johnnyaslam.pem"
chef_server_url "https://api.chef.io/organizations/jonny-muet"
cookbook_path   ["#{current_dir}/../cookbooks"]

[root@ip-172-31-83-70 .chef]# knife ssl check
Connecting to host api.chef.io:443
Successfully verified certificates from `api.chef.io'
```

```
knife ssl check
Connecting to host api.chef.io:443
Successfully verified certificates from `api.chef.io'
```

Workstation successfully connected to Chef Server now.

```
mv cookbooks/apacheecb chef-repo/cookbooks
```

```
knife cookbook upload apacheecb
```

Now we will connect Server with Nodes which is called Bootstrapping. Remember that workstation & nodes should be in same region. Every Command needs to be executed in Chef-repo directory. Two actions will be performed while bootstrapping. Adding nodes to Chef Server & Installing chef package.

Create Linux machine as well which will be treated like nodes.

Here is the procedure:

From Network Settings -> Subnet Info -> Select Subnet in same availability zone as that of CHEF-WS.

In advanced details, write this script:

```
#!/bin/bash
sudo su
yum update -y
```

Paste key pair(.pem file) of your nodes to Linux machine via WinSCP in chef-repo folder.

Now goto chef-workstation:

```
[chef-repo] # knife bootstrap [private ip of your instance node] --connection-user ec2-user --sudo -i
[key.pem] -N [name of instance]
[chef-repo] # knife node list
```

Attach recipe in run_list of node

```
knife node run_list set <Node Name> "recipe[cb::recipe-name]"
knife node show <Node Name>
```

To automate the process of auto changes, once changed on server, schedule chef-client command on every second. Crontab is present in Root/etc.

```
vi /etc/crontab
***** root chef-client
```

// Furthermore, if you don't want to do this scheduling everytime on every node. Just write in script in advanced details while creating the instance.

In advanced details, write this script:

```
#!/bin/bash
sudo su
yum update -y
echo "***** root chef-client" >> /etc/crontab
```

END

Errors:

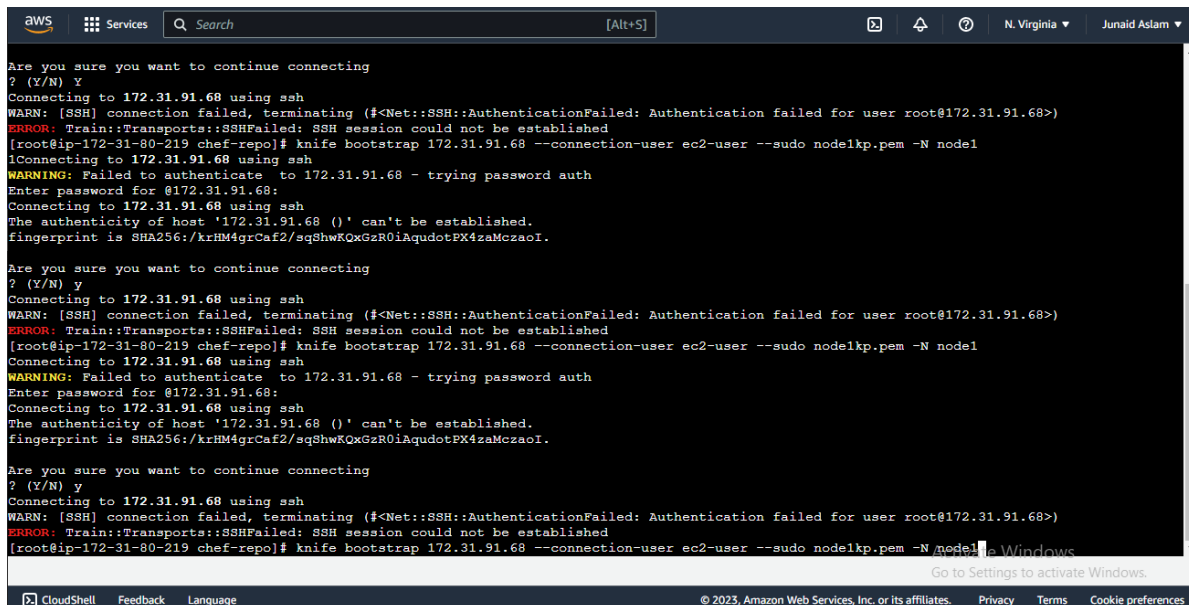
Error # 01:

If you got error while executing chef-generate cookbook command.

If Error is: "libcrypt.so.1: cannot open shared object file: no such file or directory"

Solution: Install libxcrypt package by command (yum install libxcrypt-compat -y)

Error # 02:



Solution:

Chef on Single Machine

61

Write recipe
check the syntax

→ chef exec ruby -c my-co/recipes/testrecipe.rb
o/p Syntax ok

Call the chef client

→ chef-client -zr "recipe[my-co::testrecipe]" ✓

Now

Chef WS + Chef Server + node

① After creating an account on chef-server download the starterkit - `knife` it and keep the file `chef-repo` in your linux instance using WinSCP. Once the starterkit is at the linux instance the connection will automatically be established b/c `chef-repo` directory hold `chef` files which hold the url of chef server.

→ To confirm the connection is established use

② → `] knife ssl check`

O/P Successful verified certificate

③ Now → create a node in Server A-2

Advance details `#!/bin/bash`

`sudo su`

`yes update -y`

④ Bootstrap the node (keep the `key.pem` in `chef-repo`)

`] knife bootstrap <primary ip> --connection-user
dc-user --sudo -i <key.pem> -N node name`

⑤ `] knife node list` (can be verified graphically)

(62)

⑥ Now upload the cookbook in chef server
]# knife cookbook upload my-co
]# knife cookbook list
O/P my-co (can be verified graphically)

⑦ Attach the recipe in the runlist of the node
]# knife node run-list set node1
"recipe[my-co::test-recipe]"
]# knife node show node1
O/P recipe [my-co::test-recipe]

⑧ To my Node1

]# sudo su

]# chef-client

Check by using IP in browser

→ Change some content again call chef-server

⑨ We should not call chef-server again and again

so]# vi /etc/crontab

xxxxxx root=chef-client

⑩ Create Node 2 in same AZ

Advance details

#! /bin/bash

sudo su

yum update -y

echo "xxxxxx root=chef-client" >>
/etc/crontab

Repeat the process of bootstrapping