CHEF

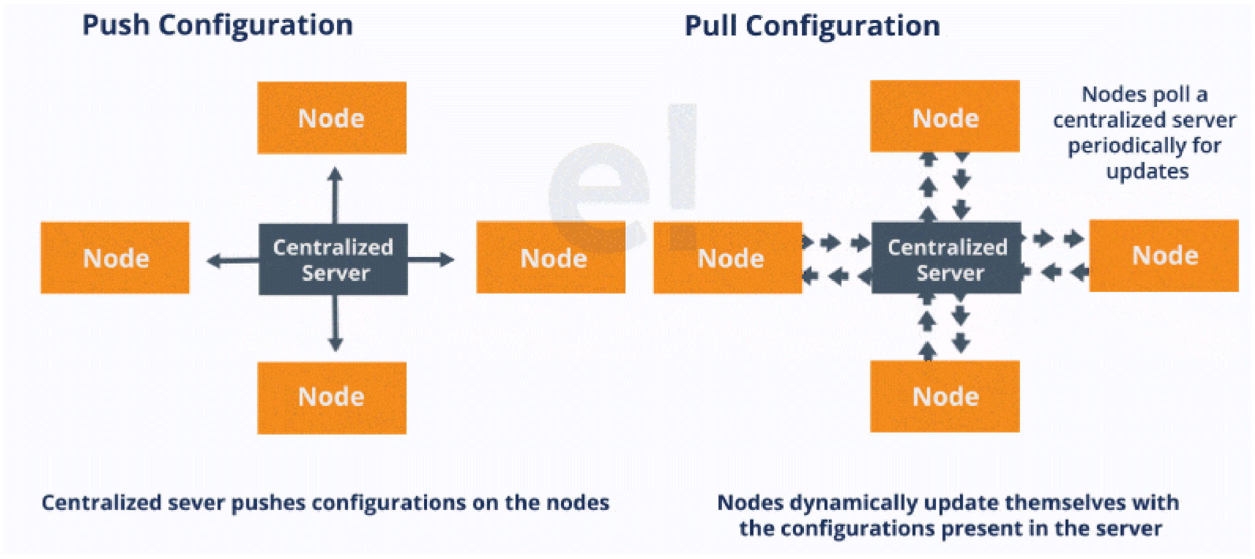
Configuration Management Tool

Version: 23.7.1042

Download Chef Workstation form :
https://www.chef.io/downloads/tools/workstation?os=amazon

```
wget
https://packages.chef.io/files/stable/chef-workstation/23.7.1042/amazon/2/chef-workstati
on-23.7.1042-1.el7.x86_64.rpm
```

Chef is used to automating the process of infrastructure provisioning. The Chef tool helps in speeding up the deployment process and software delivery. Being a DevOps tool it helps in streamlining the configuration task and managing the company's server

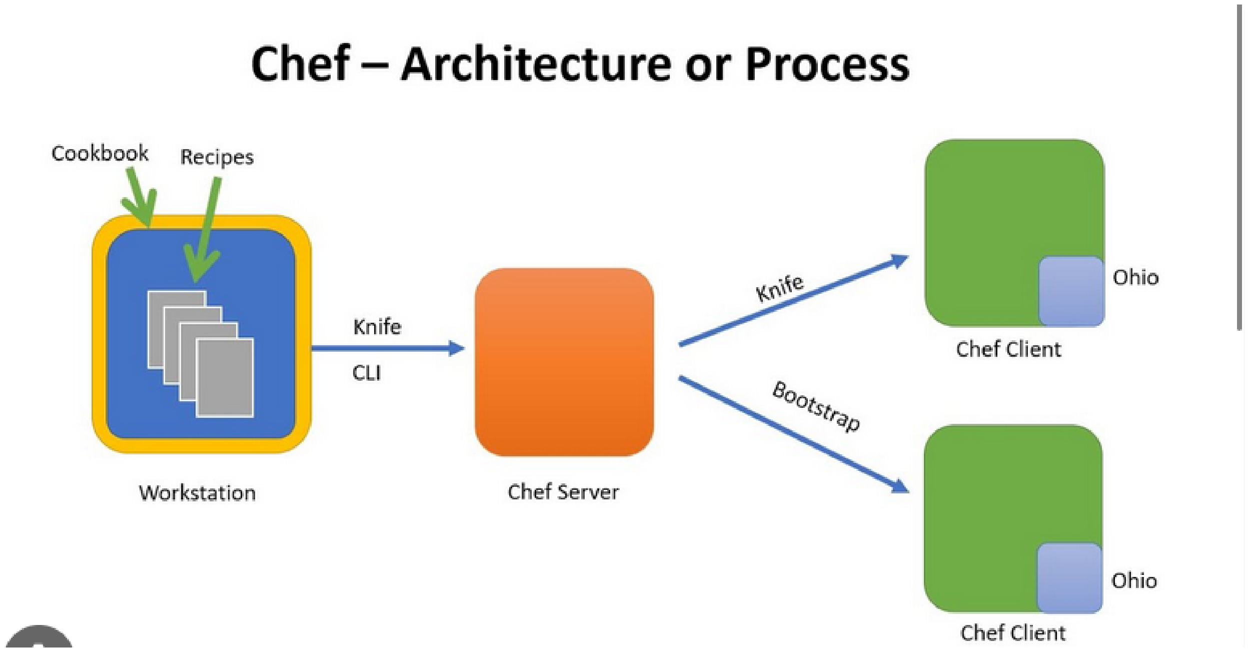

Push Configuration
Pull Configuration
Node
Node
Centralized Server
Node
Node
Node
Node
Centralized Server
Node
Node
Nodes poll a centralized server periodically for updates
Centralized sever pushes configurations on the nodes
Nodes dynamically update themselves with the configurations present in the server

- ➢ Chef is company and the name of a Configuration Management Tool written in Ruby and Erlang.
- ➢ Founded by Adam Jacobs in 2009.
- ➢ Actual Name was “Marionette” Later renamed to Chef.
- ➢ On April 2, 2019 the company announced that all their products are now open source under the Apache 2.0 License.
- ➢ Chef is used by Facebook, AWS OpsWorks, HP Public Cloud etc.

- ➢ Chef is an administration tool whatever system admins used to do manually, Now we are automating all those task by using Chef.
- ➢ Configuration Management- It is a Method through which we automate admin task.
- ➢ Configuration Management Tool turns your code into Infrastructure.
- ➢ So your code should be repeatable, testable and versionable.

Advantages of CM Tool

- ➔ Complete Automation
- ➔ Increase uptime
- ➔ Improve Performance
- ➔ Ensure Compliance
- ➔ Prevent Errors
- ➔ Reduce Cost



Components of Chef

★ **WorkStation:** Workstations are personal computers or virtual servers where all configuration code is created, tested or changed.

- ★ DevOps Engineer actually sits here and writes codes. This code is called a recipe. A collection of recipes is known as a cookbook.
- ★ Workstation communicates with the chef server using a knife.
- ★ Knife is a command line tool that uploads the cookbook to the server.

- ★ **Chef-Server:** The chef-server is a middle-man between workstation and the nodes.
- ★ All cookbooks are stored here.
- ★ Server may be hosted local or remote.

- ★ **Node:** Nodes are the systems that require the configuration.
- ★ Ohai fetch the current state of the nodes it is located in.
- ★ Node communicates with the chef-server using the chef-client.
- ★ Each mode can have a difference.
- ★ Chef-client is installed on every node.

- ★ **Knife:** Tool to establish communication among workstation, server and nodes. Knife is a command line tool that runs on a workstation.

- ★ **Chef-Client:** Tool runs on every Chef node to pull code from chef-server.
- ★ Chef-Client will gather current system configuration.
- ★ Download the desired system configuration from the chef-server.
- ★ Configure the node such that it adheres to the policy.

- ★ **Ohai:** Maintain the current state information of chef-code.

- ★ **Idempotency:** Tracking the state of system resources to ensure that the changes should not reapply repeatedly.

- ★ **Chef-Supermarket:** Where you get custom code.

How to Create Cookbook & Recipe

- ➢ First of all create one Linux Machine on AWS
- ➢ Now use putty and access the machine.
- ➢ Login as : ec2-user
  ```
  # sudo su
  # yum update -y
  ```

- ➢ To Download Workstation
- ➢ Visit **https://www.chef.io/downloads/tools/workstation?os=amazon**
- ➢ **Fill the form to start download**
- ➢ Once downloading start, stop the downloading and copy the downloadable link
  https://packages.chef.io/files/stable/chef-workstation/23.7.1042/amazon/2/chef-workstation-23.7.1042-1.el7.x86_64.rpm

- ➢ Now go to Linux Machine.

```
# wget < url >
```

```
wget
https://packages.chef.io/files/stable/chef-workstation/23.7.1042/amazon/2/chef-workstation-23.7.1042-1.el7.x86_64.rpm
```

---- **These two packages need to install on both chef workstation and chef node**

```
# yum install libcrypt* -y
# yum install dmidecode -y

# ls
```

- ➢ It shows chef packages.

```
# yum install <chef-workstation> -y
#  yum install chef-workstation-23.7.1042-1.el7.x86_64.rpm -y
# which chef
# chef --version
```

O/P->

```
exit status 127
Chef Workstation version: 23.7.1042
Test Kitchen version: 3.5.0
Cookstyle version: 7.32.2
Chef Infra Client version: 18.2.7
Chef InSpec version: 5.22.3
Chef CLI version: 5.6.12
Chef Habitat version: 1.6.652
```

Cookbook

Cookbook is a collection of recipes and some other files and folders.
Inside cookbook:

- ➔ Chefignore : like .gitignore
- ➔ Kitchen.yml : for testing cookbook
- ➔ Metadata.rb : name, version, author etc of cookbook
- ➔ Readme.md : Information about usage of cookbook
- ➔ Recipe : Where you write code
- ➔ Spec : For unit test
- ➔ Test : For integration test

Commands for test-cookbook:

```
# which chef
# mkdir cookbooks
# ls
# cd  cookbooks
# chef generate cookbook <cookbook name>
```

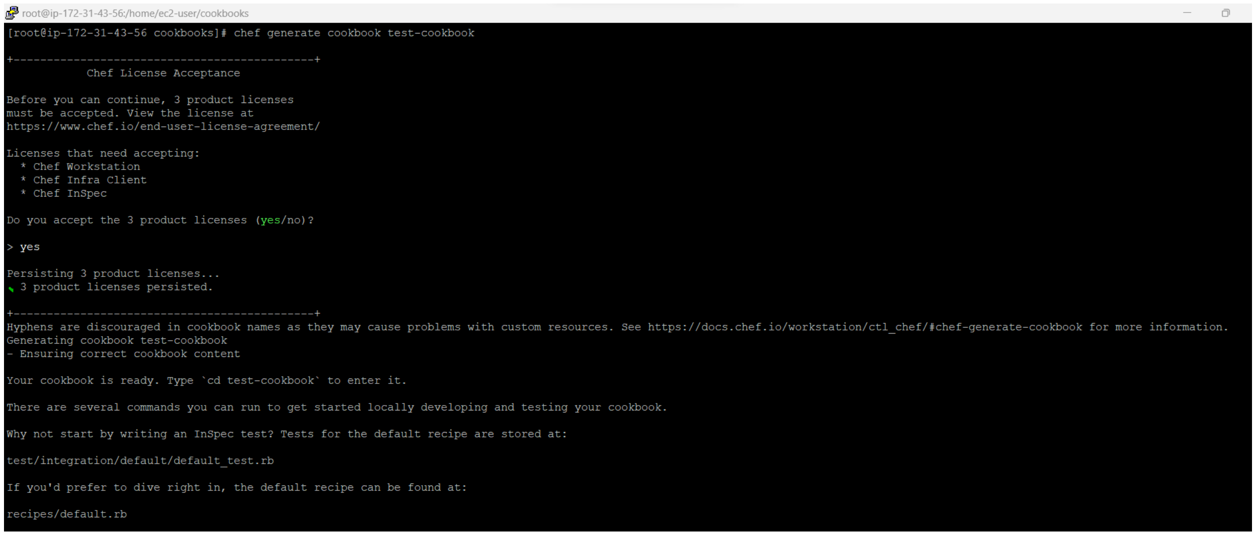
Example: chef generate cookbook test-cookbook
If you get this Error: /opt/chef-workstation/embedded/bin/ruby: error while loading shared libraries: libcrypt.so.1: cannot open shared object file: No such file or directory
Solution below:

```
[root@ip-172-31-39-232 cookbooks]# yum install libcrypt* -y
```

Try again now:

```
# chef generate cookbook <cookbook name>
```

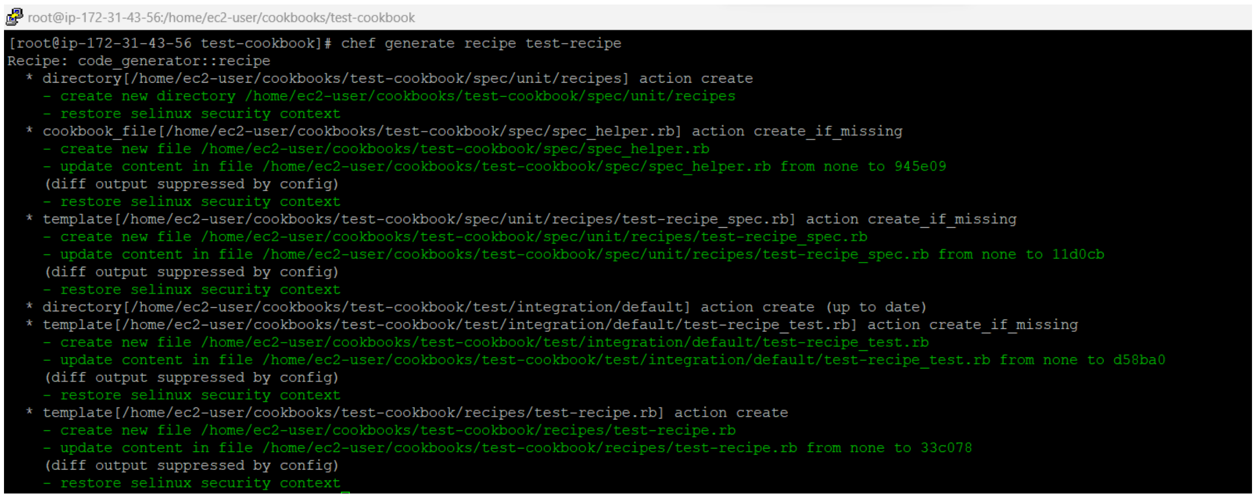
Example: chef generate cookbook test-cookbook



```
[root@ip-172-31-43-56 cookbooks]# chef generate cookbook test-cookbook

+---------------------------------------------+
            Chef License Acceptance

Before you can continue, 3 product licenses
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

Licenses that need accepting:
  * Chef Workstation
  * Chef Infra Client
  * Chef InSpec

Do you accept the 3 product licenses (yes/no)?

> yes

Persisting 3 product licenses...
✔ 3 product licenses persisted.

+---------------------------------------------+
Hyphens are discouraged in cookbook names as they may cause problems with custom resources. See https://docs.chef.io/workstation/ctl_chef/#chef-generate-cookbook for more information.
Generating cookbook test-cookbook
- Ensuring correct cookbook content

Your cookbook is ready. Type `cd test-cookbook` to enter it.

There are several commands you can run to get started locally developing and testing your cookbook.

Why not start by writing an InSpec test? Tests for the default recipe are stored at:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb
```

# yum install tree -y
# tree
# cd test-cookbook
chef generate recipe <recipe-name>

Example: chef generate recipe test-recipe



```
[root@ip-172-31-43-56 test-cookbook]# chef generate recipe test-recipe
Recipe: code_generator::recipe
  * directory[/home/ec2-user/cookbooks/test-cookbook/spec/unit/recipes] action create
    - create new directory /home/ec2-user/cookbooks/test-cookbook/spec/unit/recipes
    - restore selinux security context
  * cookbook_file[/home/ec2-user/cookbooks/test-cookbook/spec/spec_helper.rb] action create_if_missing
    - create new file /home/ec2-user/cookbooks/test-cookbook/spec/spec_helper.rb
    - update content in file /home/ec2-user/cookbooks/test-cookbook/spec/spec_helper.rb from none to 945e09
    (diff output suppressed by config)
    - restore selinux security context
  * template[/home/ec2-user/cookbooks/test-cookbook/spec/unit/recipes/test-recipe_spec.rb] action create_if_missing
    - create new file /home/ec2-user/cookbooks/test-cookbook/spec/unit/recipes/test-recipe_spec.rb
    - update content in file /home/ec2-user/cookbooks/test-cookbook/spec/unit/recipes/test-recipe_spec.rb from none to 11d0cb
    (diff output suppressed by config)
    - restore selinux security context
  * directory[/home/ec2-user/cookbooks/test-cookbook/test/integration/default] action create (up to date)
  * template[/home/ec2-user/cookbooks/test-cookbook/test/integration/default/test-recipe_test.rb] action create_if_missing
    - create new file /home/ec2-user/cookbooks/test-cookbook/test/integration/default/test-recipe_test.rb
    - update content in file /home/ec2-user/cookbooks/test-cookbook/test/integration/default/test-recipe_test.rb from none to d58ba0
    (diff output suppressed by config)
    - restore selinux security context
  * template[/home/ec2-user/cookbooks/test-cookbook/recipes/test-recipe.rb] action create
    - create new file /home/ec2-user/cookbooks/test-cookbook/recipes/test-recipe.rb
    - update content in file /home/ec2-user/cookbooks/test-cookbook/recipes/test-recipe.rb from none to 33c078
    (diff output suppressed by config)
    - restore selinux security context
```

# tree
cd ..
# vim test-cookbook/recipes/test-recipe.rb

```
file '/myfile' do
content 'Welcome to Technical Guftgu'
action :create
```

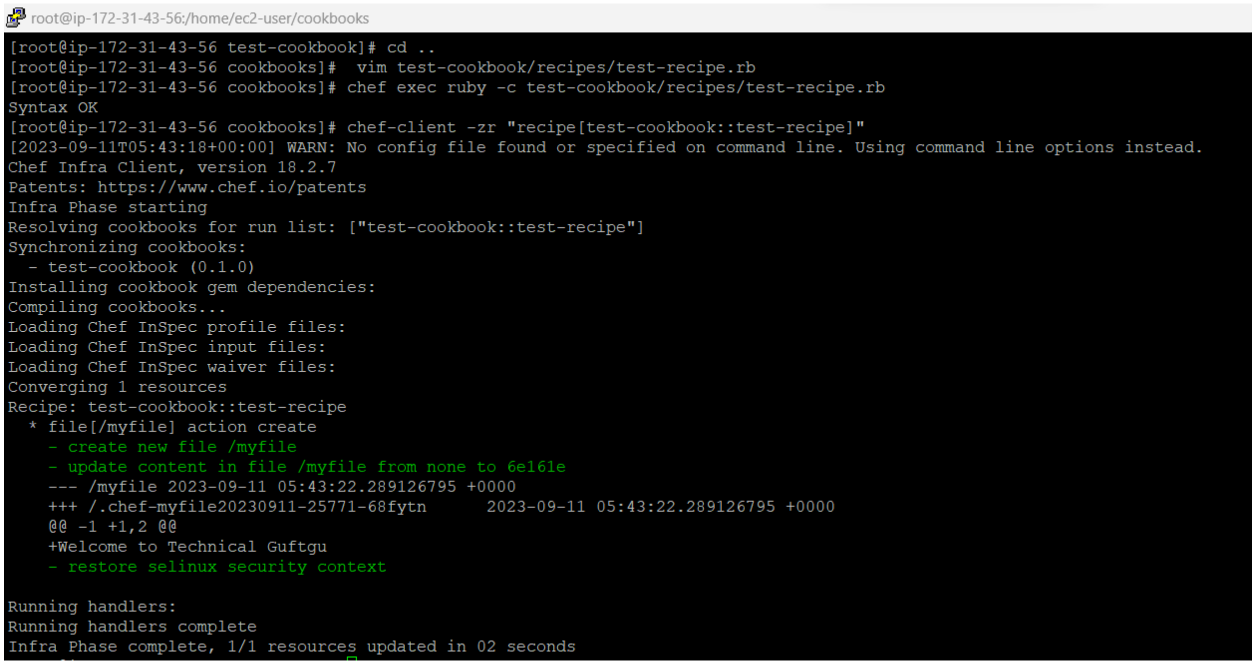
end

chef exec ruby -c test-cookbook/recipes/test-recipe.rb
o/p -> Syntax OK

chef-client -zr "recipe[test-cookbook::test-recipe]"

If error: [2023-09-08T11:04:40+00:00] ERROR: shard_seed: Failed to get dmi property serial_number: is dmidecode installed?

Solution:
# yum install dmidecode -y



```
root@ip-172-31-43-56:/home/ec2-user/cookbooks
[root@ip-172-31-43-56 test-cookbook]# cd ..
[root@ip-172-31-43-56 cookbooks]#  vim test-cookbook/recipes/test-recipe.rb
[root@ip-172-31-43-56 cookbooks]# chef exec ruby -c test-cookbook/recipes/test-recipe.rb
Syntax OK
[root@ip-172-31-43-56 cookbooks]# chef-client -zr "recipe[test-cookbook::test-recipe]"
[2023-09-11T05:43:18+00:00] WARN: No config file found or specified on command line. Using command line options instead.
Chef Infra Client, version 18.2.7
Patents: https://www.chef.io/patents
Infra Phase starting
Resolving cookbooks for run list: ["test-cookbook::test-recipe"]
Synchronizing cookbooks:
  - test-cookbook (0.1.0)
Installing cookbook gem dependencies:
Compiling cookbooks...
Loading Chef InSpec profile files:
Loading Chef InSpec input files:
Loading Chef InSpec waiver files:
Converging 1 resources
Recipe: test-cookbook::test-recipe
  * file[/myfile] action create
    - create new file /myfile
    - update content in file /myfile from none to 6e161e
    --- /myfile 2023-09-11 05:43:22.289126795 +0000
    +++ /.chef-myfile20230911-25771-68fytn      2023-09-11 05:43:22.289126795 +0000
    @@ -1 +1,2 @@
    +Welcome to Technical Guftgu
    - restore selinux security context

Running handlers:
Running handlers complete
Infra Phase complete, 1/1 resources updated in 02 seconds
```

ls /
o/p -> myfile

######## Create & Write Second Recipe ########

# cd test-cookbook
# chef generate recipe recipe2
cd ..
vim test-cookbook/recipes/recipe2.rb

package 'tree' do
action :install
end

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

# chef-client -zr "recipe[test-cookbook::recipe2]"
# cat /myfile2
# yum remove tree -y
# chef-client -zr "recipe[test-cookbook::recipe2]"
```

######## Deploying an Apache Server ############

```
# chef generate cookbook apache-cookbook
# cd apache-cookbook
# chef generate recipe apache-recipe
# tree
# cd ..
# vim apache-cookbook/recipes/apache-recipe.rb


package 'httpd' do
action :install
end

file '/var/www/html/index.html' do
content 'Welcome to Technical Guftgu'
action :create
end


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

# chef exec ruby -c apache-cookbook/recipes/apache-recipe.rb
o/p -> Syntax OK
```

```
# chef-client -zr "recipe[apache-cookbook::apache-recipe]"
```

Resources

Resource: It is the basic component of a recipe used to manage the infrastructure with different kinds of states. There can be multiple resources in a recipe, which will help in configuring and managing the infrastructure.
for e.g. ->

Package: Manages the package on a node.
Service: Manages the services on a node.
User: Manages the users on the node.
Group: Manages groups.
Template: Manages the files with embedded Ruby template.
Cookbook-file: Transfers the files from the files subdirectory in the cookbook to a location on the node.
File: Manages the content of a file on the node.
Execute: Executes a command on the node.
Cron: Edits an existing cron file on the node.
Directory: Manages the directory on the node.

Chef Attributes

- ➢ Attribute is a key value pair which represents specific details about a node.
- ➢ Attribute used by chef-client.

- ➢ Why do we use attribute->
  1. To determine the current state of the node.
  2. What the state of the node was at the end of the previous chef-client run.
  3. What the state of the node should be at the end of the current chef-client run.
- ➢ Types of Attribute:
  1. default (Least Priority)
  2. force-default
  3. normal
  4. override
  5. force-override

6. automatic (Highest Priority)

- ➢ Who define attribute
    1. Node
    2. Cookbook
    3. Roles
    4. Environment
    5. Recipe

Note: Attributes defined by ohai have the highest priority, followed by attribute defined in a recipe then attribute defined in an attribute file.

Lab Commands:

```
$ sudo su
# ohai
# ohai ipaddress
# ohai memory/total
# ls
# cd cookbooks/
# ls
O/p -> apache-cookbook
# cd apache-cookbook/
# tree
# chef generate recipe recipe3
# cd ..
# vim apache-cookbook/recipes/recipe3.rb
```

```
file '/basicinfo' do
content "This is 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
```

```
# chef exec ruby -c apache-cookbook/recipes/recipe3.rb
# chef-client -zr "recipe[apache-cookbook::recipe3]"
# ls /
# cat /basicinfo
```

Linux commands in chef recipe

```
# sudo su
# cd cookbooks
# ls
# vim test-cookbook/recipes/test-recipe.rb


execute "run a script" do
        command <<-EOH
        mkdir /rajputdir
        touch /rajputfile
        EOH
end

# chef-client -zr "recipe[test-cookbook::test-recipe]"
# ls /

# vim test-cookbook/recipes/test-recipe.rb

user "rajput" do
action :create
end

# chef-client -zr "recipe[test-cookbook::test-recipe]"
# vim test-cookbook/recipes/test-recipe.rb
group "technicalguftgu" do
action :create
members "rajput"
append true
end
# chef-client -zr "recipe[test-cookbook::test-recipe]"
# cat /etc/group
```

We run chef-client to apply recipes to bring nodes into desired state. This process is known as ‘Convergence’.

What is runlist?

To run the recipe in a sequence order that is mentioned in a run list. With this process, we can run multiple recipes, but the condition is, there must be only one recipe from one cookbook.

```
# chef-client -zr
"recipe[test-cookbook::test-recipe],recipe[apache-cookbook::apache-recipe]"
```

How to include recipe

- ➔ To call recipes/recipes from another recipe with in same cookbook.
- ➔ To run multiple recipes from the same cookbook.
- ➔ Here comes the default recipe into action.
- ➔ We can run any no of the recipes with this command, but all must be from same cookbook.

```
# vim test-cookbook/recipes/default.rb

include_recipe "test-cookbook::test-recipe"
include_recipe "test-cookbook::recipe2"

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

Now we are combining the previous two concepts, so that we can run multiple recipes from multiple cookbooks simultaneously.

```
# chef-client -zr "recipe[test-cookbook::default],recipe[apache-cookbook::default]"
```

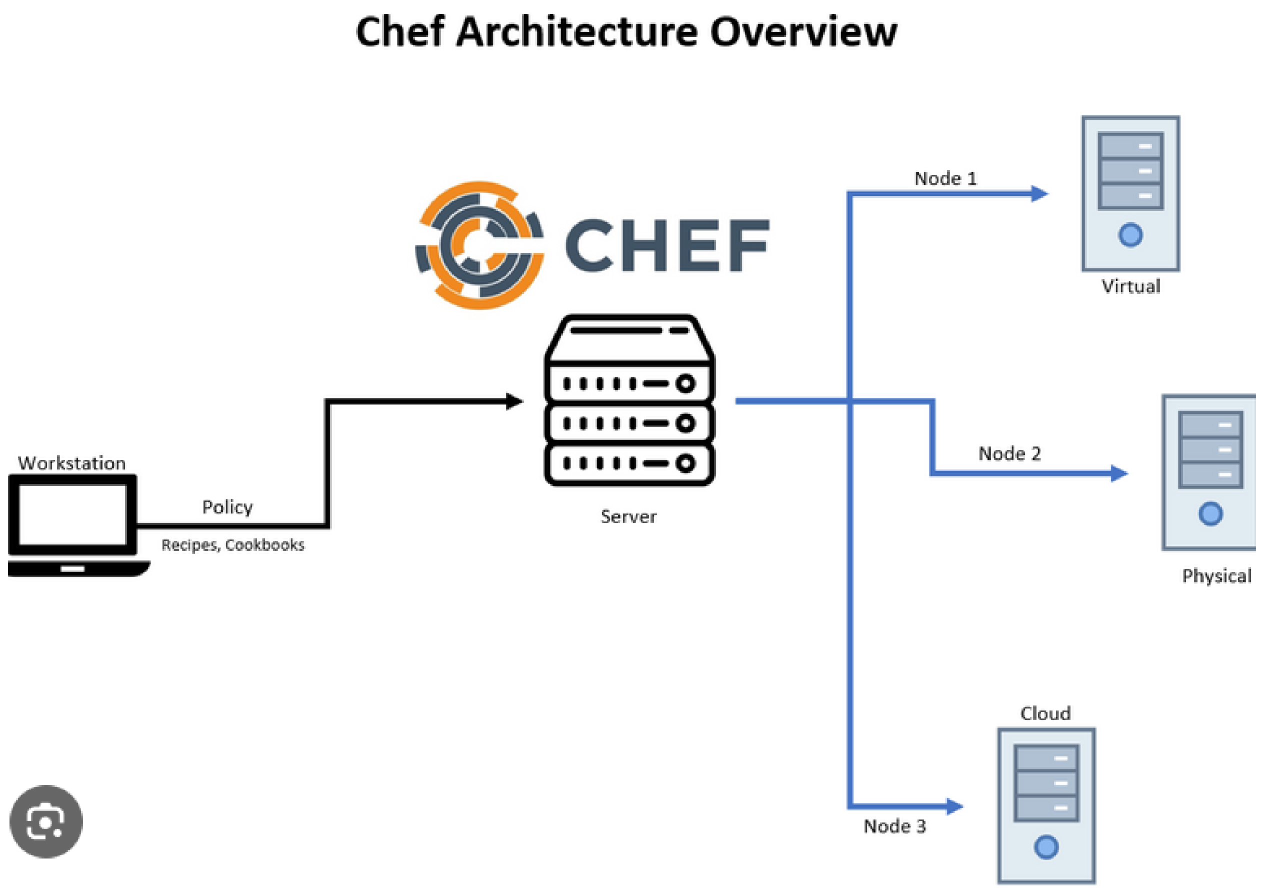
or

```
# chef-client -zr "recipe[test-cookbook],recipe[apache-cookbook]"
```

Getting started with Chef-Server & Nodes

Chef Server URL: https://manage.chef.io/login

UserName: jaiswal-chef
Pwd:



Chef-Server is going to be a mediator for the code or cookbooks.

- ➢ Firstly, Create one account in chef-server.
- ➢ Then, Attach your workstation to the chef-server.
- ➢ Now, upload your cookbooks from workstation to chef-server.
- ➢ Now, attach nodes to chef-server via bootstrap process.
- ➢ Apply cookbooks from chef-server to nodes.

Login into Amazon Linux Machine (Workstation)

```
# sudo su
# cd cookbooks
# ls
o/p-> apache-cookbook
        test-cookbook
```

> Now, open google chrome -> search manage.chef.io > create an account.
> Go to chef account -> click on organization -> starter kit -> download starter kit
> Open the download content -> unzip -> chef-repo
> Now download ‘WinScp’ -> login with ec2 credentials
> Now Drag & Drop ‘chef-repo’ folder from local machine to Linux (workstation)

Now, Open workstation in AWS Again

```
# ls
# cd ..
# ls
o/p->   chef-repo
        cookbooks
# cd chef-repo
# ls -a
o/p-> .chef cookbooks role
# cd .chef
# ls
# o/p-> config.rb technicalguftgu.pem
# cat config.rb
```

You will get the url of chef server

```
# cd ..
# knife ssl check
o/p-> Connecting to host api.chef.io:443
Successfully verified certificates from `api.chef.io'
```

Bootstrap Nodes

Attaching a node to the chef server is called Bootstrapping (Both workstation and node should be in the same AZ).

Now, onwards you have to be inside ‘chef-repo’ directly to run any command.
Two actions will be done while bootstrapping.

1. Adding node to chef-server.
2. Installing chef package.

Very important to create new organization

Create one linux machine (Node1), launch in same AZ.
Advanced details.

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

Now go to the chef-workstation.

```
# knife bootstrap node-private-ip --ssh-user ec2-user --sudo -i node1.pem -N node1
```

Example below:

```
# knife bootstrap 172.31.33.216 --ssh-user ec2-user --sudo -i node1.pem -N node1
```

{ paste node1.pem in chef-repo folder}

```
# knife node list
```

{ to see bootstrapped node}

```
# mv cookbooks/test-cookbook/ chef-repo/cookbooks/
# mv cookbooks/apache-cookbook/ chef-repo/cookbooks/
# rm -fr cookbooks
# cd chef-repo/
# cd cookbooks
# ls
apache-cookbook  chefignore  starter  test-cookbook
```

```
# knife cookbook upload apache-cookbook
Uploading apache-cookbook [0.1.0]
Uploaded 1 cookbook.
```

Now, check where cookbook is uploaded or not

```
# knife cookbook list
apache-cookbook   0.1.0
```

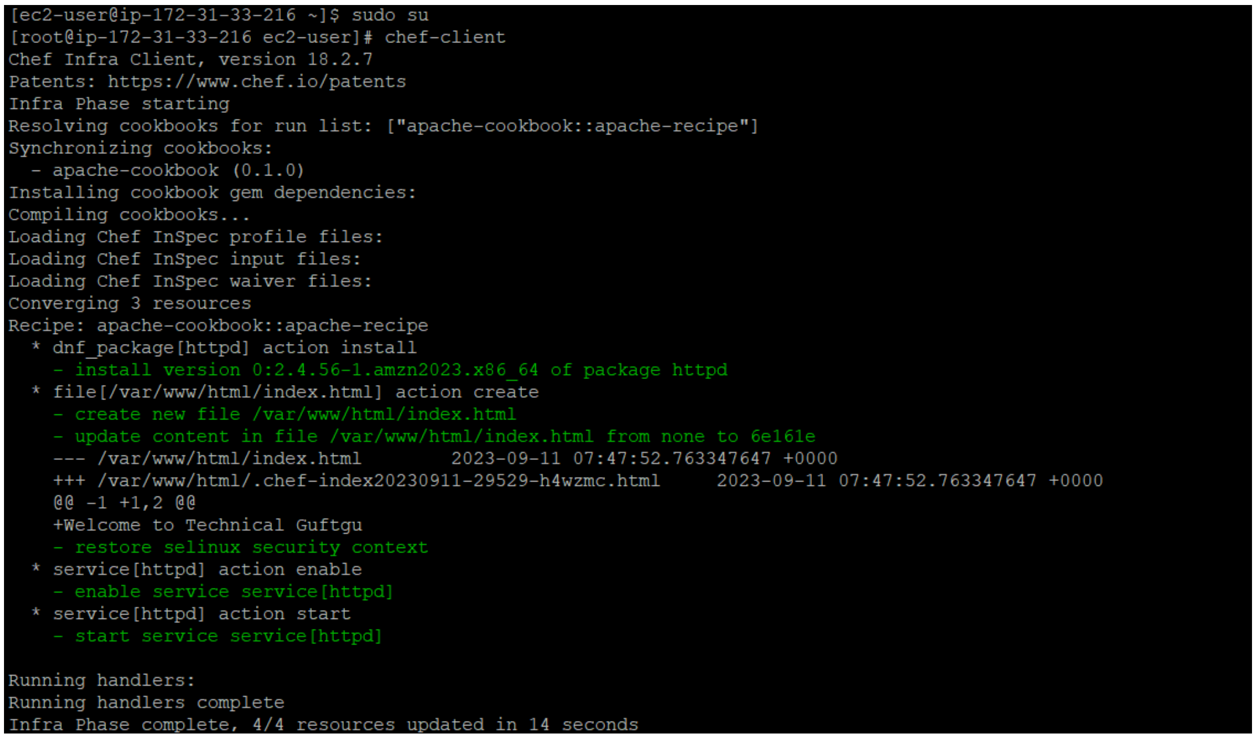
Now, We will attach the recipe, which we would like to run on node

```
# knife node run-list set node1 "recipe[apache-cookbook::apache-recipe]"
node1:
  run_list: recipe[apache-cookbook::apache-recipe]
```

# knife node show node1

```
Node Name:   node1
Environment: _default
FQDN:        ip-172-31-33-216.ap-south-1.compute.internal
IP:      13.233.111.128
Run List:    recipe[apache-cookbook::apache-recipe]
Roles:
Recipes:
Platform:    amazon 2023
Tags:
```

Now, Take access of **node1** with the help of putty
# sudo su
# chef-client



```
[ec2-user@ip-172-31-33-216 ~]$ sudo su
[root@ip-172-31-33-216 ec2-user]# chef-client
Chef Infra Client, version 18.2.7
Patents: https://www.chef.io/patents
Infra Phase starting
Resolving cookbooks for run list: ["apache-cookbook::apache-recipe"]
Synchronizing cookbooks:
  - apache-cookbook (0.1.0)
Installing cookbook gem dependencies:
Compiling cookbooks...
Loading Chef InSpec profile files:
Loading Chef InSpec input files:
Loading Chef InSpec waiver files:
Converging 3 resources
Recipe: apache-cookbook::apache-recipe
  * dnf_package[httpd] action install
    - install version 0:2.4.56-1.amzn2023.x86_64 of package httpd
  * file[/var/www/html/index.html] action create
    - create new file /var/www/html/index.html
    - update content in file /var/www/html/index.html from none to 6e161e
    --- /var/www/html/index.html        2023-09-11 07:47:52.763347647 +0000
    +++ /var/www/html/.chef-index20230911-29529-h4wzmc.html     2023-09-11 07:47:52.763347647 +0000
    @@ -1 +1,2 @@
    +Welcome to Technical Guftgu
    - restore selinux security context
  * service[httpd] action enable
    - enable service service[httpd]
  * service[httpd] action start
    - start service service[httpd]

Running handlers:
Running handlers complete
Infra Phase complete, 4/4 resources updated in 14 seconds
```

Now all files would be updated, go to the browser, paste the public ip of node1, you will get a webpage.

Now to some change in apache file on workstation:

```
# vim cookbooks/apache-cookbook/recipes/apache-recipe.rb
```

```
# knife cookbook upload apache-cookbook
Uploading apache-cookbook [0.1.0]
Uploaded 1 cookbook.
```

Now, Again go to node1 & configure cron job

```
# vim /etc/crontab

* * * * * root chef-client

:wq!
```

Now, go to workstation and do changes and upload the cookbook

```
# vi cookbooks/apache-cookbook/recipes/apache-recipe.rb
# knife cookbook upload apache-cookbook
```

Now go to the browser and check with node1 public ip.

Now, Create one more linux machine (Nide2)
Advance details

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

Now go to workstation and run bootstrap command
Now attach cookbook to node2 runlist
Now check in the browser, node2 shows webpage

Commands to delete & clean Chef-Server:

– To See list of cookbooks which are present in chef-server

```
# knife cookbook list
```

– To Delete cookbook from chef-server

```
# knife cookbook delete < cookbook name> -y
```

# knife cookbook delete apache-cookbook -y

– To see list of nodes which are present in chef-server
# knife node list

– To delete nodes from chef-server
# knife node delete <node name> -y
# knife node delete node1 -y

– To see list of clients which are present in chef-server
# knife client list
# knife client delete <client name> -y
# knife client delete csws

– To see list of roles which are present in chef-server
# knife role list

– To delete roles from chef-server
# knife role delete < role name> -y
# knife role delete apache-role -y

Chef Role

# cd chef-repo
# cd roles
ls
# vi devops.rb
name "devops"
description "Webserver role"
run_list "recipe[apache-cookbook::apache-recipe]"

Now, Comeback to chef-repo
# cd ..

Now upload the role on chef-server
# knife role from file roles/devops.rb

```
o/p-> Updated Role devops
```

If you want to see the role created

```
# knife role list
o/p-> devops.rb
```

Now create two instance as node1 & node2 in same AZ as Chef-Workstation
With user data in advanced:

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

Now bootstrap the node

```
# knife bootstrap <node private ip> --ssh-user ec2-user --sudo -i node-key.pem -N
node1
```

```
# knife bootstrap <node private ip> --ssh-user ec2-user --sudo -i node-key.pem -N
node2
```

Now, connect these nodes to role.

```
# knife node list
# knife node run_list set node1 "role[devops]"

node1:
  run_list: role[devops]
```

```
# knife node run_list set node2 "role[devops]"

node2:
  run_list: role[devops]
```

```
# knife node show node1
o/p-> runlist - role[devops]

# knife cookbook upload apache-cookbook
```

Now, check public-ip of any node in browser

```
# cat cookbooks/apache-cookbook/recipes/recipe1.rb

# vi roles/devops.rb
name "devops"
description " web server role"
run_list "recipe[apache-cookbook::recipe1]"

# knife role from file roles/devops.rb
```

Now, take access of any node via putty & check
Now, again go to workstation

```
# vi roles/devops.rb
name "devops"
description " web server role"
run_list "recipe[apache-cookbook]"

# knife role from file roles/devops.rb
# knife cookbook upload apache-cookbook

# vi roles/devops.rb
name "devops"
description " web server role"
run_list "recipe[apache-cookbook]", "recipe[test-cookbook]"
```

Now, upload this role to server

```
# knife role from file roles/devops.rb
# knife cookbook upload test-cookbook

# vi cookbooks/test-cookbook/recipes/test-recipe.rb
%w (httpd mariadb-server unzip git vim) each do | p |
package p do
action :install
end
```

```
end

# knife cookbook upload test-cookbook
```

Now, go inside any node and search git, if you will get git inside node. It means working properly.

Notes Created By: Prince Jaiswal (princejaiswal0007@gmail.com)