Hosted Chef-Server is used for this assignment.

If using Vagrant setup, then it would be good thing to mention the IP address for each machine access.

Config.vm.network “private\_network”, ip: “192.165.32.15” – [This is a reference, change ip addr as needed]

Also, if you want to check the tomcat working on your local machine then have forwarding ports.

**Let’s Get Started**

Used multiple VM’s for creating the environment,

WorkStation - VM with Ubuntu 16.04

Node 1 – VM with Ubuntu 16.04

Node 2 – VM with CentOS 7.3

In both the nodes we need to permit the root login access,

$ cd /etc/sshd

$ vi sshd\_config = [after opening the file change PermitRootLogin as Yes, and if it is commented then uncomment]

--Restart Service sshd

$ service sshd restart – [Ubuntu]

$ systemctl restart sshd – [CentOS]

Download Starter Kit and get that to the root folder [Use filezilla to connect to the workstation and put it in vagrant default shared folder, then move it to the “~/”(root folder)]

$ mv chef-starter.zip ~/

$ unzip chef-starter.zip = [will give you the chef-repo folder]

$ cd chef-repo

Now install chef in workstation

$ curl -L <https://www.opscode.com/chef/install.sh> | bash

Now install Chef Development Kit on Workstation

Get the package from the website for respective Operating System and install it.

$ sudo dpkg -i <package\_name> = [Ubuntu]

$ sudo apt-get install -f

$ rpm -ivh <package\_name> = [CentOS]

Now bootstrap the node

$ knife bootstrap <ip\_addr> -x root -P <passwd> -N <node\_name> == [specify the ip addr of node which you want to bootstrap]

Creating/Adding cookbooks to the chef repo

<PLEASE FIND THE FILE IN THE GIT REPO> - <https://github.com/mukhasir/Chef_Assignment>

🡺Please read the last lines of README.md file

$ chef generate cookbook <cb\_name>

$ knife cookbook upload <cb\_name>

Add the recipes to the run list of the nodes as needed

$ knife node run\_list add <node\_name> “recipe[<cookbookname>]” = [<cookbookname> when we have default.rb file present else the file name we have]

Add roles in workstation

Create a folder in chef-repo named “roles” and then have the <role\_name>.rb file in that.

$ mkdir /chef-repo/roles

$ cd /roles

In <role\_name>.rb file create with this items

$ vi appserver.rb

Name “”

Description “”

run\_list “role[<role\_name>]”,“recipe[<cookbook\_name>]”

Create a databag folder in chef-repo

Then create a folder which would act as databag with name “user” if you want to store user info.

$ knife data\_bag create <databag\_name>

Go into the users folder

$ cd /users

Create a .json file with the information needed.

Admin.json

{

“id”:”admin”,

“comment”:”admin-user”,

“uid”:”2005”,

“gid”:0,

“home”:”/home/admin”

“shell”:”/bin/bash”

”password”:”123456”

}

$ knife data\_bag from file <databad\_name> <.json>

Now goto the chef-repo folder and create a new cookbook “localusers” which will read the user from databag and create it.

$ chef generate cookbook localusers

$ cd cookbooks/localusers/recipes/

$ vi default.rb

search(:users, “\*:\*”).each do| data|

user data[“id”] do

comment data[“comment”]

uid data[“uid”]

gid data[“gid”]

home data[“home”]

shell data[“shell”]

end

end

$ knife cookbook upload “recipe[localusers]”

If we have added it to the role, or define a run\_list existing recipe then it will get executed.

Finally if we do “chef-client” on the nodes at each corresponding level, it will be converging with chef-server (hosted).