Task1

I use Vagrant file to create virtual machines. Since I use hyper-v as vm.provider I need to update hosts file and install ssh key manually because Vagrant does not work with Hyper-V networks

Node1: 192.168.0.108

Node2: 192.168.0.108

Control:

Commands:

Vagrant up

vagrant ssh control

sudo vim /etc/hosts # add node1 and node2 hosts

#update ssh keys

ssh-keygen -b 2048 -t rsa -f /home/vagrant/.ssh/id\_rsa -q -N ""

echo -e "Host node\*\n StrictHostKeyChecking no\n UserKnownHostsFile /dev/null\n User vagrant\n" > .ssh/config

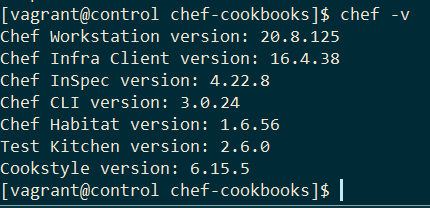
chmod 600 .ssh/config

sshpass -p 'vagrant' ssh-copy-id -i /home/vagrant/.ssh/id\_rsa.pub vagrant@node1

sshpass -p 'vagrant' ssh-copy-id -i /home/vagrant/.ssh/id\_rsa.pub vagrant@node2

Connect to control machine vagrant ssh control

* Download Chef binaries: wget <https://packages.chef.io/files/stable/chef-workstation/20.8.125/el/7/chef-workstation-20.8.125-1.el7.x86_64.rpm>
* Install Chef : sudo yum install chef-workstation-20.8.125-1.el7.x86\_64.rpm
* Check Chef version: chef -v

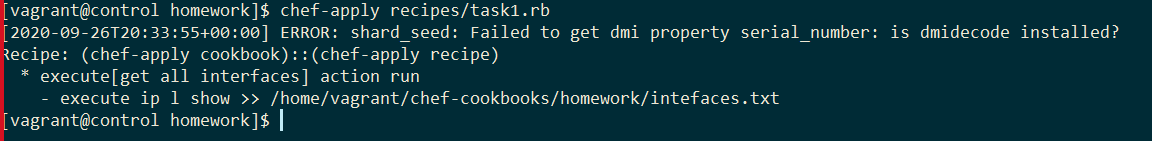


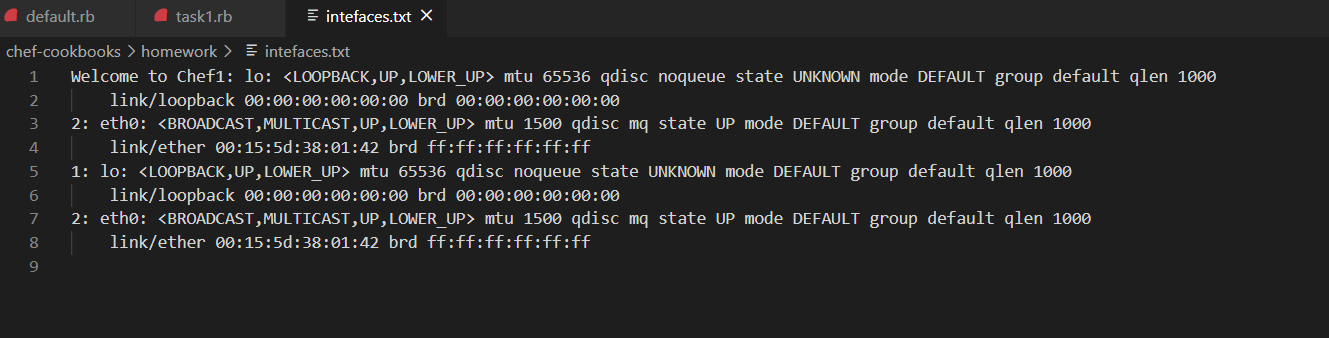
Create task1.rb

execute "get all interfaces" do

    command "ip l show >> /home/vagrant/chef-cookbooks/homework/intefaces.txt"

end



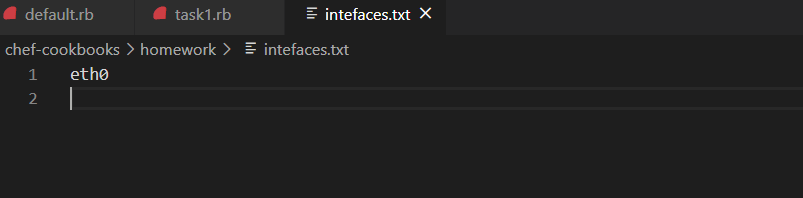


Or

execute "get all interfaces" do

    command "ip addr show | awk '/inet.\*brd/{print $NF}' >> /home/vagrant/chef-cookbooks/homework/intefaces.txt"

end



Task2

Create new recipe install\_httpd.rb

package 'httpd' do

    package\_name 'httpd'

end

service "httpd" do

  action [ :start, :enable ]

end

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

  source 'index.html.erb'

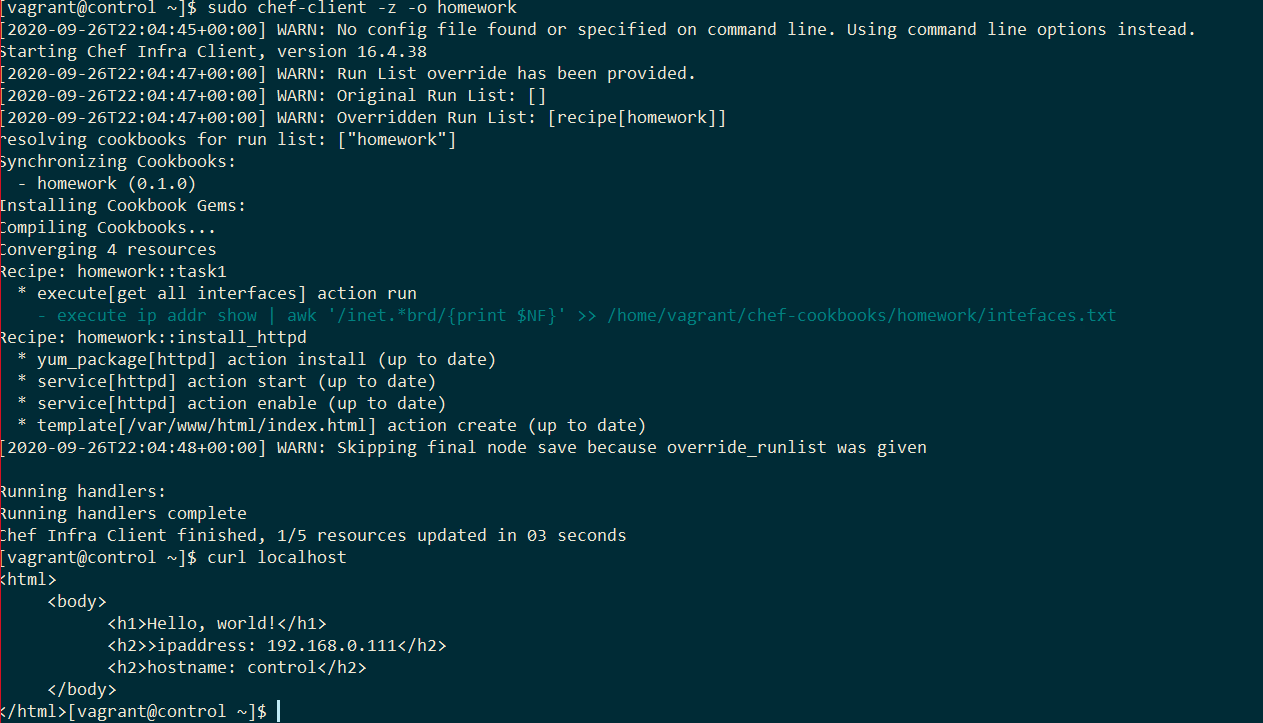
  owner 'apache'

  group 'apache'

  mode '0755'

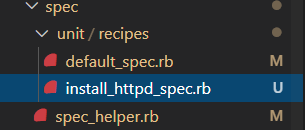
  notifies :restart, 'service[httpd]', :immediately

end



Task 3

Create unit tests



Install\_httpd\_spec.rb

require 'spec\_helper'

describe 'homework::install\_httpd' do

  context 'When all attributes are default, on an unspecified platform' do

    let(:chef\_run) do

      runner = ChefSpec::ServerRunner.new

      runner.converge(described\_recipe)

    end

    it 'converges successfully' do

      expect { chef\_run }.to\_not raise\_error

    end

  end

end

default\_spec.rb

#

# Cookbook:: homework

# Spec:: default

#

# Copyright:: 2020, The Authors, All Rights Reserved.

require 'spec\_helper'

describe 'homework::default' do

  context 'When all attributes are default, on an unspecified platform' do

    let(:chef\_run) do

      runner = ChefSpec::ServerRunner.new

      runner.converge(described\_recipe)

    end

    it 'converges successfully' do

      expect { chef\_run }.to\_not raise\_error

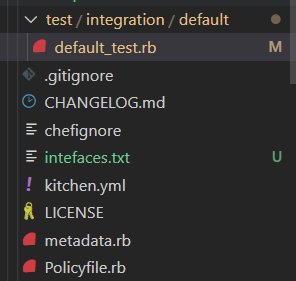
    end

  end

end

Task4

Create integration test



require 'spec\_helper'

describe 'homework::default' do

  describe port(80) do

    it { should be\_listening }

  end

  describe command('curl http://localhost') do

    its(:stdout) { should contain('hostname: control') }

  end

end