Untuk melakukan automation pada saat melakukan vagrant up, maka ada beberapa hal yang harus diubah dalam file Bootstrap.sh dan VagrantFile. Hal yang akan diubah menjadi automation pada server adalah membuat machine, menerima SSH, menjalankan playbook dan juga melalukan update file pada beberapa machine

**Update Bootstrap.sh**

#!/usr/bin/env bash

#

# install ansible (http://docs.ansible.com/intro\_installation.html)

apt-get -y install software-properties-common

apt-add-repository -y ppa:ansible/ansible

apt-get update

apt-get -y install ansible

# copy examples into /home/vagrant (from inside the mgmt node)

cp -a /vagrant/examples/\* /home/vagrant

chown -R vagrant:vagrant /home/vagrant

# configure hosts file for our internal network defined by Vagrantfile

cat >> /etc/hosts <<EOL

# vagrant environment nodes

10.0.15.15 mgmt

10.0.15.16 loadbalancer

10.0.15.21 web1

10.0.15.22 web2

10.0.15.23 web3

10.0.15.24 web4

10.0.15.25 web5

10.0.15.26 web6

10.0.15.27 web7

10.0.15.28 web8

10.0.15.29 web9

EOL

cat >>/etc/ansible/hosts <<EOL

[webserver]

web1 ansible\_ssh\_pass=vagrant ansible\_ssh\_user=vagrant

web2 ansible\_ssh\_pass=vagrant ansible\_ssh\_user=vagrant

[lb]

loadbalancer ansible\_ssh\_pass=vagrant ansible\_ssh\_user=vagrant

EOL

cat >> /etc/ansible/ansible.cfg <<EOL

[defaults]

host\_key\_checking = False

EOL

sudo apt-get -y install sshpass

sudo apt-get -y install apache2-utils

ssh-keyscan -H web2 >> /home/vagrant/.ssh/known\_hosts

ssh-keyscan -H web1 >> /home/vagrant/.ssh/known\_hosts

ssh-keyscan -H loadbalancer >> /home/vagrant/.ssh/known\_hosts

#ssh-keyscan loadbalancer web2 web1 >> /root/.ssh/known\_hosts

cd /vagrant/

sshpass -p vagrant ansible-playbook apache.yml --ask-pass

sshpass -p vagrant ansible-playbook haproxy.yml --ask-pass

**New Bootstrap2.sh**

cd /etc/haproxy/

sudo mv haproxy.cfg haproxy.cfg.old

sudo wget https://raw.githubusercontent.com/josephgunawan97/vagrant\_ansible\_multi\_server/master/haproxy.cfg

sudo service haproxy reload

**Update VagrantFile**

# Defines our Vagrant environment

#

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

(1..2).each do |i|

config.vm.define "web#{i}" do |node|

node.vm.box = "trusty64"

node.vm.hostname = "web#{i}"

node.vm.network :private\_network, ip: "10.0.15.2#{i}"

node.vm.network "forwarded\_port", guest: 80, host: "808#{i}"

node.vm.provider "virtualbox" do |vb|

vb.memory = "512"

end

end

end

# create load balancer

config.vm.define :lb do |lb\_config|

lb\_config.vm.box = "trusty64"

lb\_config.vm.hostname = "loadbalancer"

lb\_config.vm.network :private\_network, ip: "10.0.15.16"

lb\_config.vm.network "forwarded\_port", guest: 80, host: 8080

lb\_config.vm.provider "virtualbox" do |vb|

vb.memory = "1024"

end

lb\_config.vm.provision "update",type:"shell",path: "bootstrap2.sh"

end

# create mgmt node

config.vm.define :mgmt do |mgmt\_config|

mgmt\_config.vm.box = "trusty64"

mgmt\_config.vm.hostname = "mgmt"

mgmt\_config.vm.network :private\_network, ip: "10.0.15.15"

mgmt\_config.vm.provider "virtualbox" do |vb|

vb.memory = "512"

end

mgmt\_config.vm.provision :shell, path: "bootstrap.sh"

end

# create some web servers

# https://docs.vagrantup.com/v2/vagrantfile/tips.html

end