**QUESTION**

1. Buat vagrant virtualbox dan buat user 'awan' dengan password 'buayakecil'.

**Konfigurasi Vagrantfile**

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

# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure

# configures the configuration version (we support older styles for

# backwards compatibility). Please don't change it unless you know what

# you're doing.

VAGRANT\_COMMAND = ARGV[0]

Vagrant.configure(2) do |config|

if VAGRANT\_COMMAND == "ssh"

config.ssh.username = "awan"

config.ssh.password = "buayakecil"

end

# The most common configuration options are documented and commented below.

# For a complete reference, please see the online documentation at

# https://docs.vagrantup.com.

# Every Vagrant development environment requires a box. You can search for

# boxes at https://vagrantcloud.com/search.

config.vm.box = "hashicorp/precise64"

# Disable automatic box update checking. If you disable this, then

# boxes will only be checked for updates when the user runs

# `vagrant box outdated`. This is not recommended.

# config.vm.box\_check\_update = false

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine. In the example below,

# accessing "localhost:8080" will access port 80 on the guest machine.

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

config.vm.network "forwarded\_port", guest: 443, host: 8443

# Create a private network, which allows host-only access to the machine

# using a specific IP.

config.vm.network "private\_network", ip: "192.168.33.10"

# Create a public network, which generally matched to bridged network.

# Bridged networks make the machine appear as another physical device on

# your network.

config.vm.network "public\_network", ip: "10.151.36.255"

# Share an additional folder to the guest VM. The first argument is

# the path on the host to the actual folder. The second argument is

# the path on the guest to mount the folder. And the optional third

# argument is a set of non-required options.

config.vm.synced\_folder "src/", "/var/www"

# Provider-specific configuration so you can fine-tune various

# backing providers for Vagrant. These expose provider-specific options.

# Example for VirtualBox:

#

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

# Display the VirtualBox GUI when booting the machine

# vb.gui = true

#

# Customize the amount of memory on the VM:

vb.memory = "1024"

vb.cpus = 2

end

#

# View the documentation for the provider you are using for more

# information on available options.

# config.ssh.username="awan"

# config.ssh.password = "buayakecil"

# config.ssh.insert\_key = false

# Enable provisioning with a shell script. Additional provisioners such as

# Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the

# documentation for more information about their specific syntax and use.

# config.vm.provision "shell", inline: <<-SHELL

# sudo apt-get update

# sudo apt-get install -y apache2

# SHELL

config.vm.provision "shell", path: "bootstrap.sh"

end

**Konfigurasi bootstrap.sh**

#!/usr/bin/env bash

apt-get update

apt-get install -y expect

expect /var/www/user.exp

**Konfigurasi user.exp**

GNU nano 2.5.3 File: src/user.exp

#!/usr/bin/expect -f

spawn sudo adduser awan

expect "password"

send "buayakecil\r"

expect "Retype new UNIX password"

send "buayakecil\r"

set timeout 1

expect {Full Name []: }

send "\r"

expect {Room Number []: }

send "\r"

expect {Work Phone []: }

send "\r"

expect {Home Phone []: }

send "\r"

expect {Other []: }

send "\r"

expect "Is the information correct? \[Y/n\] "

send "\r"

1. Buat vagrant virtualbox dan lakukan provisioning install Phoenix Web Framework

**Konfigurasi Vagrantfile**

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

# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure

# configures the configuration version (we support older styles for

# backwards compatibility). Please don't change it unless you know what

# you're doing.

Vagrant.configure(2) do |config|

# The most common configuration options are documented and commented below.

# For a complete reference, please see the online documentation at

# https://docs.vagrantup.com.

# Every Vagrant development environment requires a box. You can search for

# boxes at https://vagrantcloud.com/search.

config.vm.box = "hashicorp/precise64"

# Disable automatic box update checking. If you disable this, then

# boxes will only be checked for updates when the user runs

# `vagrant box outdated`. This is not recommended.

# config.vm.box\_check\_update = false

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine. In the example below,

# accessing "localhost:8080" will access port 80 on the guest machine.

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

config.vm.network "forwarded\_port", guest: 443, host: 8443

# Create a private network, which allows host-only access to the machine

# using a specific IP.

config.vm.network "private\_network", ip: "192.168.33.10"

# Create a public network, which generally matched to bridged network.

# Bridged networks make the machine appear as another physical device on

# your network.

config.vm.network "public\_network", ip: "10.151.36.255"

# Share an additional folder to the guest VM. The first argument is

# the path on the host to the actual folder. The second argument is

# the path on the guest to mount the folder. And the optional third

# argument is a set of non-required options.

config.vm.synced\_folder "src/", "/var/www"

# Provider-specific configuration so you can fine-tune various

# backing providers for Vagrant. These expose provider-specific options.

# Example for VirtualBox:

#

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

# Display the VirtualBox GUI when booting the machine

# vb.gui = true

#

# Customize the amount of memory on the VM:

vb.memory = "1024"

vb.cpus = 2

end

#

# View the documentation for the provider you are using for more

# information on available options.

# config.ssh.username="awan"

# config.ssh.password = "buayakecil"

# config.ssh.insert\_key = false

# Enable provisioning with a shell script. Additional provisioners such as

# Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the

# documentation for more information about their specific syntax and use.

# config.vm.provision "shell", inline: <<-SHELL

# sudo apt-get update

# sudo apt-get install -y apache2

# SHELL

config.vm.provision "shell", path: "bootstrap.sh"

end

**Konfigurasi bootstrap.sh**

#!/usr/bin/env bash

sudo apt-get update

wget https://packages.erlang-solutions.com/erlang-solutions\_1.0\_all.deb

sudo dpkg -i erlang-solutions\_1.0\_all.deb

sudo apt-get update

sudo apt-get install -y esl-erlang elixir

mix local.hex

sudo apt-get update

sudo apt-get install -y curl

curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -

sudo apt-get update

sudo apt-get install -y nodejs

sudo apt-get install -y postgresql postgresql-client

mix archive.install https://github.com/phoenixframework/archives/raw/master/phoenix\_new.ez --force

**Run again in Vagrant:**

mix archive.install https://github.com/phoenixframework/archives/raw/master/phoenix\_new.ez --force

1. Buat vagrant virtualbox dan lakukan provisioning install:
2. php
3. mysql
4. composer
5. nginx

Setelah melakukan provioning, clone <https://github.com/fathoniadi/pelatihan-laravel.git> pada folder yang sama dengan vagrantfile di komputer host. Setelah itu sinkronisasi folder pelatihan-laravel host ke vagrant ke /var/www/web dan jangan lupa install vendor laravel agar dapat dijalankan. Setelah itu setting root document nginx ke /var/www/web. webserver VM harus dapat diakses pada port 8080 komputer host dan mysql pada vm dapat diakses pada port 6969 komputer host

**STEP-BY STEP**

Membuat folder untuk inisiasi Project Vagrant

mkdir vagrant \_\_3

cd vagrant \_\_3

Melakukan Inisiasi Project Vagrant

vagrant init

Menambah **Vagrant Box**

vagrant box add "ubuntu/xenial64"

Melakukan konfigurasi sesuai isi dari Vagrantfile

nano Vagrantfile

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

# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure

# configures the configuration version (we support older styles for

# backwards compatibility). Please don't change it unless you know what

# you're doing.

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

# The most common configuration options are documented and commented below.

# For a complete reference, please see the online documentation at

# https://docs.vagrantup.com.

# Every Vagrant development environment requires a box. You can search for

# boxes at https://vagrantcloud.com/search.

config.vm.box = "ubuntu/xenial64"

# Disable automatic box update checking. If you disable this, then

# boxes will only be checked for updates when the user runs

# `vagrant box outdated`. This is not recommended.

# config.vm.box\_check\_update = false

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine. In the example below,

# accessing "localhost:8080" will access port 80 on the guest machine.

# NOTE: This will enable public access to the opened port

config.vm.network "forwarded\_port", guest: 80, host: 8084

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine and only allow access

# via 127.0.0.1 to disable public access

# config.vm.network "forwarded\_port", guest: 80, host: 8080, host\_ip: "127.0.0.1"

# Create a private network, which allows host-only access to the machine

# using a specific IP.

config.vm.network "private\_network", ip: "192.168.0.4"

# Create a public network, which generally matched to bridged network.

# Bridged networks make the machine appear as another physical device on

# your network.

# config.vm.network "public\_network"

# Share an additional folder to the guest VM. The first argument is

# the path on the host to the actual folder. The second argument is

# the path on the guest to mount the folder. And the optional third

# argument is a set of non-required options.

config.vm.synced\_folder "pelatihan-laravel/", "/var/www/web"

# Provider-specific configuration so you can fine-tune various

# backing providers for Vagrant. These expose provider-specific options.

# Example for VirtualBox:

#

# config.vm.provider "virtualbox" do |vb|

# # Display the VirtualBox GUI when booting the machine

# vb.gui = true

#

# # Customize the amount of memory on the VM:

vb.memory = "1024"

vb.cpus = “2”

end

#

# View the documentation for the provider you are using for more

# information on available options.

# Enable provisioning with a shell script. Additional provisioners such as

# Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the

# documentation for more information about their specific syntax and use.

# config.vm.provision "shell", inline: <<-SHELL

# apt-get update

# apt-get install -y apache2

# SHELL

config.vm.provision "shell", path: "bootstrap.sh"

end

Membuat file bootstrap.sh

nano bootstrap.sh

!/usr/bin/env bash

apt-get update

#install mysql server dan client

apt-get install -y mysql-server mysql-client

#install php5 & php5-fpm dan keperluan instalasi composer

apt-get install -y curl php5 php5-fpm php5-cgi php5-cli git

#install nginx

apt-get install -y nginx

Menjalankan Vagrant sekaligus melakukan **Provisioning**

vagrant up -- provision

Masuk ke virtual Machine

vagrant ssh

Mengecek mysql dapat diakses atau tidak, dengan mengakses mysql dengan **root access**:

mysql -u root -p status

Cek status mysql

service mysql status

Untuk menghentikan servis mysq:

service mysql stop

sudo nano /etc/nginx/sites-available/default

*lakukan konfigurasi seperti pada baris berikut*

# uncommand

listen 80; ## listen for ipv4; this line is default and implied

listen [::]:80 default ipv6only=on; ## listen for ipv6

#mengganti root folder directory pada nginx

#root /usr/share/nginx/www;

root /var/www/web; --- direktori shared folder

#menambahkan index.php pada line index

index index.php index.html index.htm;

#menambahkan baris-baris berikut

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

try\_files $uri =404;

#uncommand

fastcgi\_pass unix:/var/run/php5-fpm.sock;

fastcgi\_index index.php;

include fastcgi\_params;

*membuat file index.php pada direktori berikut*

nano /usr/share/nginx/www/index.php

<?php

echo "Hello from 192.168.0.4"

phpinfo();

?>

melakukan konfigurasi www.conf

sudo nano /etc/php5/fpm/pool.d/www.conf

#uncommand

user = www-data

group = www-data

#mengganti listen to php5-fpm.sock

listen = /var/run/php5-fpm.sock

#uncommand

listen.owner = www-data

listen.group = www-data

melakukan service restart pada nginx dan php5-fpm

sudo service nginx restart

sudo service php5-fpm restart

Mengganti Ownership dari folder .composer :

sudo su

chown -R vagrant /home/vagrant/.composer/

*folder tersebut hanya dapat diubah oleh root dan dilarang menggunakan sintaks terkait composer dengan root access*

**Instalasi Laravel pada Project**

composer global require "laravel/installer=~1.1"

**Install Composer**

curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin –filename=composer

**Mengkonfigurasi firewall untuk block akses Server Apache**

sudo ufw app list

sudo ufw deny "Apache Full"

**Melakukan clone pelatihan-laravel pada direktori computer host :**

git clone https://github.com/fathoniadi/pelatihan-laravel.git

**Menghapus Apache**

sudo apt-get autoremove

or

sudo apt-get remove apache2\*

4. Buat vagrant virtualbox dan lakukan provisioning install:

1. Squid proxy
2. Bind9

Membuat folder untuk inisiasi Project Vagrant

mkdir vagrant \_\_4

cd vagrant \_\_4

Melakukan Inisiasi Project Vagrant

vagrant init

Menambah **Vagrant Box**

vagrant box add "ubuntu/xenial64"

Melakukan konfigurasi sesuai isi dari Vagrantfile

nano Vagrantfile

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

# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure

# configures the configuration version (we support older styles for

# backwards compatibility). Please don't change it unless you know what

# you're doing.

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

# The most common configuration options are documented and commented below.

# For a complete reference, please see the online documentation at

# https://docs.vagrantup.com.

# Every Vagrant development environment requires a box. You can search for

# boxes at https://vagrantcloud.com/search.

config.vm.box = "ubuntu/xenial64"

# Disable automatic box update checking. If you disable this, then

# boxes will only be checked for updates when the user runs

# `vagrant box outdated`. This is not recommended.

# config.vm.box\_check\_update = false

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine. In the example below,

# accessing "localhost:8080" will access port 80 on the guest machine.

# NOTE: This will enable public access to the opened port

config.vm.network "forwarded\_port", guest: 80, host: 8085

# Create a forwarded port mapping which allows access to a specific port

# within the machine from a port on the host machine and only allow access

# via 127.0.0.1 to disable public access

# config.vm.network "forwarded\_port", guest: 80, host: 8080, host\_ip: "127.0.0.1"

# Create a private network, which allows host-only access to the machine

# using a specific IP.

config.vm.network "private\_network", ip: "192.168.0.5"

# Create a public network, which generally matched to bridged network.

# Bridged networks make the machine appear as another physical device on

# your network.

# config.vm.network "public\_network"

# Share an additional folder to the guest VM. The first argument is

# the path on the host to the actual folder. The second argument is

# the path on the guest to mount the folder. And the optional third

# argument is a set of non-required options.

config.vm.synced\_folder "pelatihan-laravel/", "/var/www/web"

# Provider-specific configuration so you can fine-tune various

# backing providers for Vagrant. These expose provider-specific options.

# Example for VirtualBox:

#

# config.vm.provider "virtualbox" do |vb|

# # Display the VirtualBox GUI when booting the machine

# vb.gui = true

#

# # Customize the amount of memory on the VM:

vb.memory = "1024"

vb.cpus = “2”

end

#

# View the documentation for the provider you are using for more

# information on available options.

# Enable provisioning with a shell script. Additional provisioners such as

# Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the

# documentation for more information about their specific syntax and use.

# config.vm.provision "shell", inline: <<-SHELL

# apt-get update

# apt-get install -y apache2

# SHELL

config.vm.provision "shell", path: "bootstrap.sh"

end

Membuat file bootstrap.sh

nano bootstrap.sh

apt-get update

# Instalasi squid3

apt-get install -y squid3

# Instalasi bind9

apt-get install -y bind9

Menjalankan Vagrant sekaligus melakukan **Provisioning**

vagrant up -- provision

Masuk ke virtual Machine

vagrant ssh

Ubuntu Version

lsb\_release –a

Menjalankan service bind9 dan squid3

sudo service bind9 start & service squid3 start

**CLOUD COMPUTING**

**Nama Kelompok**

Reinhart Caesar - 5115100132

Joshua Resamuel - 5115100172