Magento 2 Deployment on Stage and Production Server

1. Install Vagrant on your personal MAC system.
2. Create a directory with name “magento\_dev\_env”.
3. Create a directory with name “dev\_env”. This directory is mapped to /var/www/html directory of VM machine using the Vagrantfile
4. Copy Vagrantfile from shared location and paste in “magento\_dev\_env” directory.
5. Create local VM using **vagrant up** command
6. Vagrantfile script will create VM with apache2, php, mysql, composer.
7. Connect to VM machine by issue **vagrant ssh** command from magento\_dev\_env directory
8. Setup date on VM using command “dpka-reconfigure tzdata”
9. Edit the 0000-default.conf file and add following lines

<Directory “/var/www/html”>

AllowOverride All

</Directory>

1. Edit /etc/apache2/apache2.conf and change User & Group to vagrant user
2. edit /etc/php/7.1/apache2/php.ini and change below value\

memory\_limit = 2G

Upload\_max\_filesize = 100M

Max\_execution\_time = 360

Date.timezone = Asia/Dubai

Opcache.save.comments = True

1. restart apache2 services
2. create database withname magento2 with user magento2 have full rights.
3. Now download magento2 using composer on your personal MAC system in directory /magento\_dev\_env/dev\_env/ using the following command

composer create-project --repository=https://repo.magento.com/ magento/project-community-edition=2.3.1 .

1. Enter then user name & password copy from magento marketplace user profile 🡪 access key
2. Install magento2 on local VM machine using IP Address 192.168.10.10. Open browser on your MAC System and enter IP 192.168.10.10 , setup will start, follow the screen instruction
3. remove the .gitignore file and copy .gitignore file from share location studyguides
4. now initialize git repository in dev\_env directory **using git** init command
5. add all file repository using **commad git add \* --force** and commit
6. now create a repository on github with name magento2store
7. copy the and paste in local repository to link local repo with remote repo
8. push all the code to new repository using **git push –u origin master** command
9. make sure magento site is working locally
10. now install Capistrano on local VM machine by following steps
    1. update the VM using command sudo apt-get update
    2. sudo apt-get install build-essential
    3. sudo apt-get install curl
    4. gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB
    5. \curl -sSL [https://get.rvm.io](https://get.rvm.io/) | bash -s stable
    6. source ~/.rvm/scripts/rvm
    7. rvm install 2.5.3
    8. rvm use 2.5.3
    9. rvm use 2.5.3 –default
    10. ruby -v
    11. above commad from a – j will install ruby on system
11. gem install capistrano-magento2 this command will install Capistrano on local VM
12. go to magento2 root directory and create file with name “Gemfile” using command **touch Gemfile** and add following code

source ‘https://rubygems.org’

gem ‘capistrano-magento2’

1. run the following commad from root directory **bundle install**
2. go to magento2 root directory and create director with name “tools”
3. inside “tools” directory create another directory with name “cap”
4. issue command **mkdir -p tools/cap** to get 26&27 results
5. move to “cap”directory and run cap install command.
6. Run the command **cap install**
7. This will create directory structure
8. Now edit the Capfile and add below code

require "capistrano/magento2/deploy"

require "capistrano/magento2/pending”

1. Now edit the config/deploy.rb file and add below code

set :application, "magento\_store"

set :repo\_url, [git@github.com:rizwanwakil2018/magento2store.git](mailto:git@github.com:rizwanwakil2018/magento2store.git)

set :magento\_deploy\_languages, ['en\_US', 'en\_GB']

set :magento\_auth\_public\_key, '6d304bfc5ac8e8cdac4c60c14247ca39'

set :magento\_auth\_private\_key, '65efe00a6ff3fcaf29d8ac13aef70531'

set :keep\_releases, 3

1. Now edit the config/deploy/staging.rb file and add below code

server '34.76.138.14', user: 'vagrant', roles: %w{app db web}

set :deploy\_to, '/var/www/html/staging'

set :branch, proc { `git rev-parse --abbrev-ref staging`.chomp }

1. Now edit the config/deploy/production.rb file and add below code

server '34.76.138.14', user: 'vagrant', roles: %w{app db web}

set :deploy\_to, '/var/www/html/production'

set :branch, proc { `git rev-parse --abbrev-ref master`.chomp }

1. Now generate ssh key on VM machine and add to remote repository by using below command
   1. ssh-keygen -t rsa -b 4096 -C "vagrant"
2. install gcloud on local VM using below command

curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

curl <https://sdk.cloud.google.com> | bash

exec –l $SHELL

gcloud init

1. connect to production server from local VM using command

gcloud compute ssh magento2

1. now copy ssh .pub file from local VM to remote production server using command

gcloud compute scp ~/.ssh/id\_rsa.pub [vagrant@magento2:~/.ssh/](mailto:vagrant@magento2:~/.ssh/)

1. now open id\_rsa.pub file on local VM machine , copy the content , open the github 🡪 setting🡪SSH key and create new and paste the content
2. make sure you can connect from local VM to Production server using commad

ssh [vagrant@34.76.138.14](mailto:vagrant@34.76.138.14)

1. make sure to copy the id\_rsa.pub content and open Production VM in google console and create a SSH key and paste the content
2. now make sure using below command you can connect to repository from local VM

ssh –T [git@github.com](mailto:git@github.com)

1. setting up staging and production server in Google Cloud
   1. login to google cloud console
   2. create a VM
   3. go to VPC network and change IP to static copy the VM id address to

safe location (34.76.138.14)

* 1. generate ssh key on production server using command

ssh-keygen -t rsa -b 4096 -C "vagrant"

* 1. open the id\_rsa.pub and copy content and paste in github ssh/deploy area

1. now make sure using below command you can connect to repository from production server

ssh –T [git@github.com](mailto:git@github.com)

1. install composer & git on Production server using command

sudo apt-get install curl git

sudo curl -sS https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer

1. now connect to Production server and install the apache, php & mysql using command

install apache using sudo apt-get install apache2

Setup date on VM using command “dpka-reconfigure tzdata”

Edit the 0000-default.conf file and add following lines

<Directory “/var/www/html”>

AllowOverride All

</Directory>

install php using commad

sudo apt-get –y update

sudo apt-get -y install software-properties-common

sudo add-apt-repository ppa:ondrej/php

sudo apt-get –y update

sudo apt-get -y install php7.1

sudo apt-get -y install libapache2-mod-php7.1 php7.1-common php7.1-gmp php7.1-curl php7.1-soap php7.1-bcmath php7.1-intl php7.1-mbstring php7.1-xmlrpc php7.1-mcrypt php7.1-mysql php7.1-gd php7.1-xml php7.1-cli php7.1-zip

sudo apt-get -y install zip unzip php7.0-zip

sudo apt-get -y install php7.1-xdebug phpunit

sudo a2enmod rewrite

sudo service apache2 restart

Edit /etc/apache2/apache2.conf and change User & Group to vagrant user

edit /etc/php/7.1/apache2/php.ini and change below value\

memory\_limit = 2G

Upload\_max\_filesize = 100M

Max\_execution\_time = 360

Date.timezone = Asia/Dubai

Opcache.save.comments = True

restart apache2 services

1. Install mysql on Production server

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

sudo service mysql start

sudo mysql\_secure\_installation

1. Take database backup on local VM using command

mysqldump –u root –p magento2 > file\_name.sql

transfer this file to production server using command

gcloud compute scp file\_location\file\_name magento2:~

create database with same name and user permission in production server then restore the databases from backup file using command

mysql –u root –p magento2 < file\_name.sql

make sure to change the url in core\_config\_data table

update core\_config\_data set value="http://34.76.138.14/production/current/pub" where config\_id=2;

1. now move to /var/www/html/production directory and create follow directory structure

mkdir –p share/app/etc

touch share/app/etc/env.php

copy content from local VM mean from development system and paste in env.php