skip first step if u have ssh-key generated before

step -1 :-

we need to add our ssh-key on github

for the first time setup process

$ ssh-keygen -t rsa

Press enter

if we don't have id\_rsa.pub key then genereate key and to generate key follow this link

"https://help.github.com/articles/generating-ssh-keys/"

$ ssh-copy-id cca-user@192.155.69.240

now u will be able to login without password

server login

$ ssh cca-user@192.155.69.240

step-2:-

++++++++++++++++++++++To setup rbenv and ruby

$ sudo apt-get update

$ sudo apt-get install curl git-core build-essential zlib1g-dev libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libcurl4-openssl-dev libxml2-dev libxslt1-dev python-software-properties

$ git clone https://github.com/sstephenson/rbenv.git ~/.rbenv

$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc

$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc

$ exec $SHELL

$ type rbenv

$ git clone https://github.com/sstephenson/ruby-build.git ~/.rbenv/plugins/ruby-build

$ rbenv install 2.2.2

$ rbenv global 2.2.2

$ ruby -v

step -3:-

$ gem install bundler

step-4 :

install mysql

$ sudo apt-get update

$ sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql

sudo mysql\_install\_db

sudo /usr/bin/mysql\_secure\_installation

For more information "https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu"

Step -5:-

Install nginx

$ sudo apt-get install nginx

$ ps ax | grep nginx

Edit Nginx config file :-

nano /etc/nginx/nginx.conf

remove all previous code and paste below code

+++++++++++++++++++++++++++++++++++++++

user cca-user;

worker\_processes 4;

pid /var/run/nginx.pid;

events {

worker\_connections 768;

}

http {

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 65;

types\_hash\_max\_size 2048;

server\_name\_in\_redirect off;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

gzip on;

gzip\_disable "msie6";

include /etc/nginx/conf.d/\*.conf;

include /etc/nginx/sites-enabled/\*;

}

$ sudo service nginx start

+++++++++++++++++++++++++++++++++++++++

nano /etc/nginx/sites-enabled/default

remove all previous code and paste below code

++++++++++++++++++++++++++++++++++++++++

upstream app {

server unix:/home/cca-user/unicorn.sock fail\_timeout=0;

}

server {

listen 80;

# Application root, as defined previously

root /home/cca-user/Project\_Zulu/current/public;

#ssl on;

#ssl\_certificate /etc/nginx/ssl/SSL.crt;

#ssl\_certificate\_key /etc/nginx/ssl/YOUR\_APP\_NAME.key;

#server\_name ;

try\_files $uri/index.html $uri @app;

access\_log /var/log/nginx/access.log combined;

error\_log /var/log/nginx/error.log;

location @app {

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header Host $http\_host;

proxy\_redirect off;

proxy\_pass http://app;

#proxy\_set\_header X-Forwarded-Proto https; # <-- don't need this if you're not running SSL

}

error\_page 500 502 503 504 /500.html;

client\_max\_body\_size 4G;

keepalive\_timeout 10;

}

++++++++++++++++++++++++++

in project folder in local system

add unicorn.rb file under config folder

add these lines

# Set your full path to application.

app\_dir = File.expand\_path('../../', \_\_FILE\_\_)

shared\_dir = '/home/cca-user'

# Set unicorn options

worker\_processes 2

preload\_app true

timeout 30

# Fill path to your app

working\_directory app\_dir

# Set up socket location

listen "#{shared\_dir}/unicorn.sock", :backlog => 64

# Loging

stderr\_path "#{shared\_dir}/log/unicorn.stderr.log"

stdout\_path "#{shared\_dir}/log/unicorn.stdout.log"

# Set master PID location

pid "#{shared\_dir}/pids/unicorn.pid"

before\_fork do |server, worker|

defined?(ActiveRecord::Base) and ActiveRecord::Base.connection.disconnect!

old\_pid = "#{server.config[:pid]}.oldbin"

if File.exists?(old\_pid) && server.pid != old\_pid

begin

sig = (worker.nr + 1) >= server.worker\_processes ? :QUIT : :TTOU

Process.kill(sig, File.read(old\_pid).to\_i)

rescue Errno::ENOENT, Errno::ESRCH

# someone else did our job for us

end

end

end

after\_fork do |server, worker|

defined?(ActiveRecord::Base) and ActiveRecord::Base.establish\_connection

end

before\_exec do |server|

#ENV['BUNDLE\_GEMFILE'] = "#{app\_dir}/Gemfile"

end

+++++++++++++++++++++++++++++++++++++

open ur gemfile in projectfolder next add below gems

gem 'unicorn'

gem 'mina'

gem 'mina-sidekiq', :require => false

gem 'mina-unicorn', :require => false

run bundle install in ur terminal

$ bundle install

then run mina init command in terminal

$ mina init

Created config/deploy.rb.

open deploy.rb file

paste the below content

require 'mina/bundler'

require 'mina/rails'

require 'mina/git'

require 'mina\_sidekiq/tasks'

require 'mina/unicorn'

require 'mina/rbenv' # for rbenv support. (http://rbenv.org)

# require 'mina/rvm' # for rvm support. (http://rvm.io)

set :domain, '192.155.69.240'

set :deploy\_to, '/home/cca-user/Project\_Zulu'

set :repository, 'git@github.com:juanmramirez/Project\_Zulu.git'

set :branch, 'deploy'

set :user, 'cca-user'

set :forward\_agent, true

set :port, '22'

set :term\_mode, nil

set :unicorn\_pid, "/home/cca-user/pids/unicorn.pid"

set :bundle\_gemfile, "#{deploy\_to}/#{current\_path}/Gemfile"

# For system-wide RVM install.

# set :rvm\_path, '/usr/local/rvm/bin/rvm'

# Manually create these paths in shared/ (eg: shared/config/database.yml) in your server.

# They will be linked in the 'deploy:link\_shared\_paths' step.

set :shared\_paths, ['config/database.yml', 'log', 'config/secrets.yml']

# Optional settings:

# set :user, 'foobar' # Username in the server to SSH to.

# set :port, '30000' # SSH port number.

# set :forward\_agent, true # SSH forward\_agent.

# This task is the environment that is loaded for most commands, such as

# `mina deploy` or `mina rake`.

task :environment do

queue 'export LANGUAGE=en\_US.UTF-8'

queue 'export LANG=en\_US.UTF-8'

queue 'export LC\_ALL=en\_US.UTF-8'

queue %{

echo "-----> Loading environment"

#{echo\_cmd %[source ~/.bashrc]}

}

invoke :'rbenv:load'

end

# Put any custom mkdir's in here for when `mina setup` is ran.

# For Rails apps, we'll make some of the shared paths that are shared between

# all releases.

task :setup => :environment do

queue! %[mkdir -p "#{deploy\_to}/#{shared\_path}/log"]

queue! %[chmod g+rx,u+rwx "#{deploy\_to}/#{shared\_path}/log"]

queue! %[mkdir -p "#{deploy\_to}/#{shared\_path}/config"]

queue! %[chmod g+rx,u+rwx "#{deploy\_to}/#{shared\_path}/config"]

queue! %[touch "#{deploy\_to}/#{shared\_path}/config/database.yml"]

queue %[echo "-----> Be sure to edit '#{deploy\_to}/#{shared\_path}/config/database.yml'."]

queue! %[touch "#{deploy\_to}/shared/config/secrets.yml"]

queue %[echo "-----> Be sure to edit 'shared/config/secrets.yml'."]

queue! %[mkdir -p "#{deploy\_to}/shared/pids/"]

queue! %[chmod g+rx,u+rwx "#{deploy\_to}/shared/pids"]

queue! %[mkdir -p "/home/cca-user/pids"]

queue! %[chmod g+rx,u+rwx "/home/cca-user/pids"]

queue! %[mkdir -p "/home/cca-user/log"]

queue! %[chmod g+rx,u+rwx "/home/cca-user/log"]

end

desc "Deploys the current version to the server."

task :deploy => :environment do

deploy do

# Put things that will set up an empty directory into a fully set-up

# instance of your project.

invoke :'git:clone'

invoke :'deploy:link\_shared\_paths'

invoke :'bundle:install'

invoke :'rails:db\_migrate'

invoke :'rails:assets\_precompile'

invoke :'deploy:cleanup'

to :launch do

invoke :'unicorn:restart'

end

end

end

run mina setup in terminal

$ mina setup

$ ssh cca-user@192.155.69.240

$ nano Project\_Zulu/shared/config/database.yml

add below code

++++++++++++++++++++++++++++++++++++

default: &default

adapter: mysql2

encoding: utf8

pool: 5

username: root

password: root

socket: /var/run/mysqld/mysqld.sock

production:

<<: \*default

database: enake\_production

++++++++++++++++++++++++++++

Adding secrets keys :

$ nano Project\_Zulu/shared/config/secrets.yml

production:

secret\_key\_base: ee61d626098b933c46572c25df6da3418665baea929c680576268f37de719cb1b870d7af102eedf7021565004b0dd27d45204e349d381e34a46b13a710c$

(TO generate secret key on production Run - "rake secret" with in Project\_Zulu directory)

run mina deploy under project folder

$ mina deploy