Server Side

1. Install ssh on ubuntu
2. Make your user a sudo user by *sudo usermod -aG sudo “yourusername”*
3. First install rvm for ubuntu. Look at rvm github for help.
   1. Once installed. Install ruby 2.5.5 by *rvm install 2.5.5*
   2. Once ruby is installed install bundle by *gem install bundler*
4. Install nodejs by *sudo apt-get install -y nodejs && sudo ln -sf /usr/bin/nodejs /usr/local/bin/node*
5. Next install Git
   1. Add the global config to git: git config –global user.name, etc.
6. Then generate a ssh key for github by *ssh-keygen -t rsa -C “youremail”*
   1. Then go to /home/username/.ssh/id\_rsa.pub and copy whats inside.
   2. Go to github and under your username go to setting and go to ssh keys and enter your create a new ssh key. Paste what you copied and give it a name. Save and create.
7. Clone your code where you want by going to directory and using git clone the ssh version instead of the http
8. When you clone. Switch to your branch so that it can have your code and not masters
9. Navigate to code and change code to production by *echo ‘export RAILS\_ENV=production’ >> ~/.bashrc*
   1. Then type: *Source ~/.bashrc*
10. Install mysql2 server and client
    1. Create a new user with password so that you can use for the database
    2. Use: *GRANT ALL PRIVILEGES ON \*.\* TO 'username'@'localhost' IDENTIFIED BY 'password';*
11. Make sure that the database.yml has the username and password for the production database.
12. Then create a secret key by bundle exec rake secret. Copy what it outputs and make a file called secrets.yml by *nano config/secrets.yml* 
    1. Write: production:

*secret\_key\_base: Paste what you copied*

1. Next create the database by *rails db:create*
2. Then migrate by *rails db:migrate*
   1. If you get an error It might be because one of migration is not up before the others. I think it was of polymorphism ones. That means you will have to migrate that one first and then migrate the rest.
3. Then precompile the assets by *bundle exec rake assets:precompile*
4. Install apache2
5. Install passenger module by *sudo passenger-install-apache2-module*
   1. Follow the instructions
6. You will need the path to ruby so type *passenger-config about ruby-command*
   1. Copy the path that passenger uses for ruby and you will use it for the apache file
7. Change apache enabled-sites by *sudo nano /etc/apache2/sites-enabled/000-default.conf* to

<VirtualHost \*:80>

# Tell Apache and Passenger where your app's 'public' directory is

DocumentRoot DirectoryOfYourCode/public

PassengerRuby /The path that you copied for ruby on step 16.

<Directory DirectoryOfYourCode/public>

Allow from all

Options -MultiViews

Require all granted

</Directory>

</VirtualHost>

1. When you are done then restart apache2 and it should work. Go the IP address of the VM and it should display the page. If not you can check whats going on by typing *tail -f log/production.log* 
   1. That will give you an error log that you can check.

App side

1. Make sure you have the passenger gem in gemfile
2. Make sure that you have msyql2 gem in gemfile
3. Make sure you have uglifier gem installed
4. Go to the config/environments/production.rb and change *config.assets.js\_compressor = :uglifier* to *config.assets.js\_compressor = Uglifier.new(harmony: true)* 
   1. If you don’t do this it won’t precompile the javascripts