Guide to Deploy on Vagrant and Ansible

1. Setting up the Vagrant Virtual Box

   When setting up to utilize Vagrant and Ansible for the purpose of deployment, one must first do a few steps to initialize the virtual machine.

   Download Vagrant and Virtual Box
   Vagrant: https://www.vagrantup.com/downloads.html
   VirtualBox: https://www.virtualbox.org/wiki/Downloads

   Once the software's have been downloaded you can now begin to create your virtual machine.
   Create a new directory where you wish to access your Vagrant virtual machine.
   Through the terminal, navigate to that directory and
   create the Vagrant virtual machine
   -#vagrant init hashicorp/precise64

   This will create a VagrantFile, where you can then start up your virtual machine
   -#vagrant up

   Once the virtual machine has been booted up you can then access the virtual machine through the terminal using
   -#vagrant ssh

   Setting up Password-less Authentication

   To set up password-less authentication, we need to create keys to be used rather than the actual password.

   -#ssh-keygen –t rsa

   This function will create a pair of keys within the ~/.ssh/ directory, unless specified otherwise. Move to the directory that the keys that were created in by:
   -#cd .ssh

   Verify the all the files are present by using:
   -# ls –l

   Make sure to check if the file authorized_keys exists, if not then create it by using:
   -#touch authorized_keys

   Copy the contents within id_rsa.pub into authorized_keys file using:
   -#cat id_rsa.pub >> authorized_keys

   To test that you have successfully configured a password-less system use:
   -#ssh vagrant@localhost
   If done successfully, then you should not of have to use your password.

Exit out of the ssh session by using
-#exit

Setting up Ansible

Before we install Ansible, we first need to first install a few prerequisite.

-#sudo apt-get update

This is used to update Ubuntu to the latest version, installing critical functions that are required by Ansible

-#sudo apt-get install ubuntu-desktop

If we run the virtual box from VirtualBox, then it only displays the terminal. We wish to have a GUI as well so that we can see our website.

Once we have those two installed, then we can move on to Installing and configuring Ansible.
-#sudo-apt-add-repository ppa: ansible/ansible
-#sudo-apt-get install ansible

These two functions will create a repository and then install Ansible to the virtual machine.
Move to the Ansible directory

-#cd /etc/ansible

Here we need to edit the host file.
-#sudo nano hosts

Edit the content so it reads

[php]
localhost ansible_ssh_user=vagrant

Ctrl-O to save and Ctrl-X to quit

Ansible Playbook is the method Ansible uses to configure and deploy a program by using scripts (YAML language) to automate the process.

Edit the playbook by using

-#sudo nano php.yml

copy the contents below into the playbook (Not including the ---)

```
---------------------------------------
- hosts: php
sudo: yes

tasks:
 - name: install packages
          apt: name={{ item }} update_cache=yes state=latest
          with_items:
          - git
          - mcrypt
          - php5-cli
          - php5-curl
          - php5-fpm
          - php5-intl
 #        - php5-json
          - php5-mcrypt
          - php5-sqlite
          - sqlite3
          - apache2
          - libapache2-mod-php5


         - name: Clone Git Repository
            git: >
           dest=/var/www/html
           repo=https://github.com/Mandeep/Task4.git
           update=yes
         sudo: yes
```

What this does is install the functions that are needed to deploy the website, such as Git, PHP and Apache2.

Near the end, the Clone Git Repository purpose is to download the repository at https://github.com/mandeep/Task4.git, into the /var/www/html directory. This directory is the root directory for the Apache2 server, meaning that any php or html file in this directory can be accessed using Apache2 through https://localhost. The update function means that if any changes are made to the Github repository, then the new files will replace the old ones in the destination. Finally the sudo: yes function allows the Playbook to have root access to the directory, incase there is any permission requirements.

To execute the playbook and to open the webbrowser to see your project use:
-#ansible-playbook php.yml --ask-sudo-pass; xdg-open http://localhost