Automation using Ansible: Spin up a new Linux VM using Ansible playbook.

Ansible is an open-source automation tool for provisioning, application

deployment (WordPress deployment in this case), and configuration management.

Gone are the days of SSH’ing into your server to run a command or hacking

together bash scripts to semi-automate laborious tasks. Whether you’re managing

a single server or an entire fleet, Ansible can not only simplify the process but save

you time. So, what makes Ansible so great?

Ansible is completely agent-less, meaning you don’t have to install any software

on your managed hosts. All commands are run through Ansible via SSH and if

Ansible needs updating you only need to update your single control machine and

not any remote machines. The only prerequisite to running Ansible commands is

to have Python installed on your control machine.

Procedure:

Installation

Step 1: First, ensure that pip is installed.

sudo easy\_install pip

Step 2: Then install Ansible.

sudo pip install ansible

Step 3: Once the installation has completed you can verify that everything

installed correctly by issuing:

ansible --version

Step 4: If you were installing Ansible on Ubuntu the commands would be:

sudo apt update

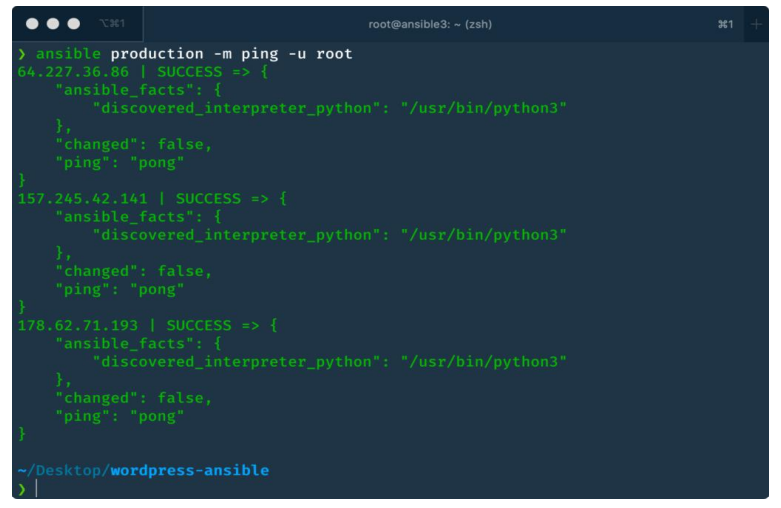
sudo apt install software-properties-common

sudo apt-add-repository --yes --update ppa:ansible/ansible

sudo apt install ansible

Running Commands

ansible production -m ping -u root



Organization of Playbook

ansible.cfg

hosts

provision.yml

roles

nginx

handlers

main.yml

tasks

main.yml

--- - hosts: production

user: root

vars:

username: ashley

password:

$6$rlLdG6wd1CT8v7i$7psP8l26lmaPhT3cigoYYXhjG28CtD1ifILq9KzvA0W0

TH2Hj4.iO43RkPWgJGIi60Mz0CsxWbRVBSQkAY95W0

public\_key: ~/.ssh/id\_rsa.pub

roles:

common

- ufw

- user

- nginx

- php

- mariadb

- wp

-cli

- ssh