Snipe.yml

inventory.yml

snipeit.j2

inventory.yml

[my\_hosts] ## slave ##

13.232.150.195 ansible\_ssh\_private\_key\_file=/home/ubuntu/snipeit/slave11.pem ansible\_user=ubuntu

snipeit.j2

server {

listen 80;

listen [::]:80;

server\_name {{ snipeit\_domain }};

root {{ root\_path }};

index index.php index.html index.htm;

access\_log /var/log/nginx/snipeit.saur.com.access.log;

error\_log /var/log/nginx/snipeit.saur.com.error.log;

location / {

try\_files $uri $uri/ /index.php$is\_args$args;

}

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass unix:/var/run/php/{{ php\_fpm }}.sock;

include fastcgi\_params;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_intercept\_errors on;

}

}

snipe.yml

---

- name: Set up Snipeit Application

hosts: my\_hosts

become: yes

vars:

snipeit\_db\_name: snipeit\_db

snipeit\_db\_user: snipeit\_user

snipeit\_db\_password: password

snipeit\_domain: 13.232.150.195

php\_fpm: php8.1-fpm

root\_path: /var/www/html

app\_path: /var/www/snipeit

tasks:

- name: update and upgrade packages

ansible.builtin.apt:

update\_cache: true

upgrade: yes

- name: Install NGINX, MySQL

ansible.builtin.apt:

name: "{{ packages }}"

state: present

vars:

packages:

- nginx

- mysql-server

- python3-pip

- name: Add APT repository universe

ansible.builtin.apt\_repository:

repo: "deb http://archive.ubuntu.com/ubuntu {{ ansible\_distribution\_release }} universe"

state: present

- name: Install PHP and PHP FPM

ansible.builtin.apt:

name: "{{ packages }}"

state: present

vars:

packages:

- php-fpm

- php-mysql

- name: Create a new site by copying snipeit.saur.com to etc/nginx/sites-available/

ansible.builtin.template:

src: snipeit.j2

dest: /etc/nginx/sites-available/snipeit.saur.com

- name: creating stats for symbolic link

ansible.builtin.stat:

path: /etc/nginx/sites-enabled/snipeit.saur.com

register: symbolic\_link

- name: creating symbolic link /etc/nginx/sites-available/snipeit.saur.com and /etc/nginx/sites-enabled/

ansible.builtin.command: sudo ln -s /etc/nginx/sites-available/snipeit.saur.com /etc/nginx/sites-enabled/

when: not symbolic\_link.stat.exists

- name: unlink /etc/nginx/sites-enabled/default

ansible.builtin.file:

path: /etc/nginx/sites-enabled/default

state: absent

notify:

- Restart nginx service

- testing php using info.php

- name: secure mariadb

become: yes

ansible.builtin.expect:

command: mysql\_secure\_installation

responses:

'Enter current password for root': ''

'Set root password': 'n'

'Remove anonymous users': 'y'

'Disallow root login remotely': 'y'

'Remove test database': 'y'

'Reload privilege tables now': 'y'

timeout: 1

register: secure\_mariadb

failed\_when: "'... Failed!' in secure\_mariadb.stdout\_lines"

- name: Install PyMySQL

ansible.builtin.pip:

name: PyMySQL

- name: Create Snipe-IT database

community.mysql.mysql\_db:

name: "{{ snipeit\_db\_name }}"

state: present

login\_unix\_socket: /var/run/mysqld/mysqld.sock

- name: Create Snipe-IT database user

community.mysql.mysql\_user:

name: "{{ snipeit\_db\_user }}"

password: "{{ snipeit\_db\_password }}"

priv: "{{ snipeit\_db\_name }}.\*:ALL"

host: localhost

login\_unix\_socket: /var/run/mysqld/mysqld.sock

state: present

- name: restart mysql

ansible.builtin.systemd:

name: mysql

state: restarted

- name: creating a stat for snipeit

ansible.builtin.stat:

path: "{{ app\_path }}"

register: snipeit

- name: cloning snipe-it from git

ansible.builtin.git:

repo: https://github.com/snipe/snipe-it

dest: "{{ app\_path }}"

clone: true

when: not snipeit.stat.exists

- name: copy .env.example to .env

ansible.builtin.copy:

src: /var/www/snipeit/.env.example

dest: /var/www/snipeit/.env

remote\_src: true

- name: update .env file

ansible.builtin.lineinfile:

path: "{{ app\_path }}/.env"

regexp: "{{ item.regexp }}"

line: "{{ item.line }}"

with\_items:

- { regexp: '^DB\_DATABASE=', line: "DB\_DATABASE={{ snipeit\_db\_name }}" }

- { regexp: '^DB\_USERNAME=', line: "DB\_USERNAME= {{ snipeit\_db\_user }}"}

- { regexp: '^DB\_PASSWORD=', line: "DB\_PASSWORD= {{ snipeit\_db\_password }}"}

- { regexp: '^APP\_URL=', line: "APP\_URL=http://{{ snipeit\_domain }}" }

- name: set owner permissions

ansible.builtin.file:

path: "{{ app\_path }}/"

state: directory

recurse: yes

owner: www-data

group: www-data

- name: Set directory permissions

ansible.builtin.file:

path: "{{ item }}"

state: directory

recurse: yes

owner: www-data

group: www-data

mode: 0755

loop:

- "{{ app\_path }}/storage"

- "{{ app\_path }}/public/uploads"

- name: creating stats for composer.phar

ansible.builtin.stat:

path: /usr/local/bin/composer

register: composer

- name: Download Composer

ansible.builtin.shell: sudo curl -sS https://getcomposer.org/installer | php && sudo mv composer.phar /usr/local/bin/composer

when: not composer.stat.exists

- name: installing necessary packages

ansible.builtin.apt:

name: "{{ packages }}"

state: present

vars:

packages:

- git

- unzip

- php

- php-curl

- php-mysql

- php-gd

- php-ldap

- php-zip

- php-mbstring

- php-xml

- php-bcmath

- php-tokenizer

- name: composer install

community.general.composer:

command: install

working\_dir: "{{ app\_path }}"

prefer\_source: true

no\_dev: true

vars:

ansible\_become: yes

ansible\_become\_user: root

environment:

COMPOSER\_ALLOW\_SUPERUSER: 1

- name: php key-generation

ansible.builtin.command: sudo php artisan key:generate --force

args:

chdir: "{{ app\_path }}"

- name: Update Nginx site root

ansible.builtin.lineinfile:

path: /etc/nginx/sites-available/snipeit.saur.com

regexp: '^(\s\*)root(\s+)/var/www/html;'

line: "root {{ app\_path }}/public;"

notify:

- Restart nginx service

handlers:

- name: Restart nginx service

ansible.builtin.service:

name: nginx

state: restarted

- name: testing php using info.php

ansible.builtin.command: sudo bash -c "echo -e '<?php\nphpinfo();\n?>' > /var/www/html/info.php"

#####