yaml

---

- hosts: webservers

become: true

tasks:

- name: Install dependencies

apt:

name: ['python3-pip', 'python3-dev']

state: present

- name: Install Flask

pip:

name: flask

state: present

- name: Copy Flask app

copy:

src: application2.py

dest: /home/ubuntu/application2.py

- name: Run Flask app

command: nohup python3 /home/ubuntu/application2.py > /dev/null 2>&1 &

- hosts: HAproxy

become: true

tasks:

- name: Install HAproxy

apt:

name: haproxy

state: present

- name: Configure HAproxy

copy:

src: haproxy.cfg

dest: /etc/haproxy/haproxy.cfg

- name: Restart HAproxy

systemd:

name: haproxy

state: restarted

This playbook installs the necessary dependencies on the web servers, copies the Flask app to the servers, and runs the app. It also installs HAproxy on the HAproxy server, copies the HAproxy configuration file, and restarts HAproxy.

---

- hosts: all

gather\_facts: yes

- name: Configuring HAproxy loadbalancer

hosts: HAproxy

become: yes

vars:

ui\_pass: "{{ lookup('file', './uipassword') }}"

tasks:

- name : sudo wait

become: yes

shell: while sudo fuser /var/lib/dpkg/lock >/dev/null 2>&1; do sleep 5; done;

- name: Installing HAproxy

apt:

name: haproxy

state: present

update\_cache: true

- name: gather server ip addresses

setup:

filter: ansible\_default\_ipv4.address

- name: copy files haproxy.cfg

template:

src: haproxy.cfg.j2

dest: "/etc/haproxy/haproxy.cfg"

notify:

- restart haproxy

- name: install nginx, snmp, snmp-mibs-downloader

apt:

update\_cache: yes

name: [ 'nginx', 'snmp', 'snmp-mibs-downloader' ]

state: latest

- name: copy snmp conf file

template:

src: snmp.conf

dest: "/etc/snmp/snmp.conf"

- name: copy nginx config files

template:

src: nginx.conf.j2

dest: "/etc/nginx/nginx.conf"

- name: nginx start

service:

name: nginx

state: restarted

handlers:

- name: restart haproxy

service:

name: haproxy

state: restarted

- hosts: webservers

become: yes

tasks:

- name: install pip

apt:

name: python3-pip

state: present

- name: install flask

pip:

executable: pip3

name: flask

state: latest

- name: install gunicorn

pip:

executable: pip3

name: gunicorn

state: latest

- name: createnew directory

file:

path: "/home/flask-app/"

state: directory

- name: copy app.py to webservers

template:

src: "application2.py"

dest: "/home/flask-app/app.py"

- name: start flask app

shell: |

cd /home/flask-app/

gunicorn --bind 0.0.0.0:80 app:app&

- name: install snmpd

apt:

name: snmpd

state: latest

- name: cpy snmpd conf file

template:

src: snmpd.conf.j2

dest: "/etc/snmp/snmpd.conf"

- name: run snmpd

service:

name: snmpd

state: restarted

- hosts: HAproxy

become: yes

tasks:

- name: test HAproxy 1

uri:

url: "http://localhost/"

return\_content: yes

register: html\_content

- debug:

msg: "{{ html\_content.content }}"

- name: test HAproxy 2

uri:

url: "http://localhost/"

return\_content: yes

register: html\_content

- debug:

msg: "{{ html\_content.content }}"

- name: test HAproxy 3

uri:

url: "http://localhost/"

return\_content: yes

register: html\_content

- debug:

msg: "{{ html\_content.content }}"

ggggggggggggggggggggggggggggggggggggggggg

---

- name: Deploy HAproxy and Flask app

hosts: HAproxy

become: true

tasks:

- name: Install HAproxy

apt:

name: haproxy

state: present

- name: Configure HAproxy

template:

src: templates/haproxy.cfg.j2

dest: /etc/haproxy/haproxy.cfg

notify: restart haproxy

- name: Install Flask

apt:

name: python3-flask

state: present

- name: Copy Flask app

copy:

src: files/application2.py

dest: /var/www/application2.py

notify: restart flask

handlers:

- name: restart haproxy

service:

name: haproxy

state: restarted

- name: restart flask

systemd:

name: flask

state: restarted

- name: Set up SSH access via Bastion host

hosts: webservers

become: true

tasks:

- name: Install sshpass

apt:

name: sshpass

state: present

- name: Add Bastion host to known\_hosts

shell: "ssh-keyscan bastionNSO >> /home/ubuntu/.ssh/known\_hosts"

- name: Copy SSH key to Bastion host

authorized\_key:

user: ubuntu

state: present

key: "{{ lookup('file', '/home/ubuntu/.ssh/id\_rsa.pub') }}"

delegate\_to: bastionNSO

- name: Copy SSH key to dev servers

authorized\_key:

user: ubuntu

state: present

key: "{{ lookup('file', '/home/ubuntu/.ssh/id\_rsa.pub') }}"

with\_items: "{{ groups['webservers'] }}"

delegate\_to: bastionNSO

- name: Test the deployed application

hosts: HAproxy

become: true

tasks:

- name: Send requests to HAproxy and check responses