**Infra Optimization - Source Code**

* **Saaketh Gunturu**
* ***Course End Project #1***
* ***PGP DO - DevOps Capstone Project***
* ***PGP DO JUL 2022 Cohort 1***

***Create an Elastic IP address using Ansible***

| ---  - name: Create Elastic IP address  hosts: localhost  connection: local  become: false  vars:  region: us-east-1  tasks:  - name: Allocate Elastic IP address  ec2\_eip:  region: "{{ region }}"  register: eip  - name: Print Elastic IP address  debug:  var: eip.public\_ip |
| --- |

***Provision EC2 instances using Ansible:***

| ***Master:***  *- name: Provision EC2 instance*  *hosts: localhost*  *gather\_facts: no*  *vars:*  *keypair: cap-key1*  *region: us-east-1*  *zone: us-east-1d*  *instance\_type: t3.micro*  *ami\_id: ami-0557a15b87f6559cf*  *tasks:*  *- name: Create EC2 instance*  *ec2:*  *key\_name: "{{ keypair }}"*  *instance\_type: "{{ instance\_type }}"*  *image: "{{ ami\_id }}"*  *wait: yes*  *region: "{{ region }}"*  *count: 1*  *instance\_tags:*  *name: simplilearn*  *vpc\_subnet\_id: subnet-0a055bfd59c3adbee*  *assign\_public\_ip: yes*  *register: ec2*  *- name: Add new instance to host group*  *add\_host:*  *name: "{{ item.public\_ip }}"*  *groups: launched*  *with\_items: "{{ ec2.instances }}"*  *- name: Print the public IP addresses*  *debug:*  *var: item.public\_ip*  *with\_items: "{{ ec2.instances }}"* |
| --- |

| ***Worker nodes***  - name: Provision EC2 instance  hosts: localhost  gather\_facts: no  vars:  keypair: cap-key1  region: us-east-1  zone: us-east-1d  instance\_type: t2.micro  ami\_id: ami-0557a15b87f6559cf  tasks:  - name: Create EC2 instance  ec2:  key\_name: "{{ keypair }}"  instance\_type: "{{ instance\_type }}"  image: "{{ ami\_id }}"  wait: yes  region: "{{ region }}"  count: 2  instance\_tags:  name: simplilearn  vpc\_subnet\_id: subnet-0a055bfd59c3adbee  assign\_public\_ip: yes  register: ec2  - name: Add new instance to host group  add\_host:  name: "{{ item.public\_ip }}"  groups: launched  with\_items: "{{ ec2.instances }}"  - name: Print the public IP addresses  debug:  var: item.public\_ip  with\_items: "{{ ec2.instances }}" |
| --- |

***Add Elastic IP address to EC2 instances using Ansible***

| ---  - name: Add Elastic IP address to EC2 instance  hosts: localhost  become: true  vars:  region: us-east-1  tasks:  - name: Associate Elastic IP address with instance  ec2\_eip:  region: "{{ region }}"  device\_id: i-046820428d3c2ba87  ip: 44.208.48.229  - name: Associate Elastic IP address with instance  ec2\_eip:  region: "{{ region }}"  device\_id: i-0490c9efef3f996b9  ip: 44.209.45.77  - name: Associate Elastic IP address with instance  ec2\_eip:  region: "{{ region }}"  device\_id: i-084f421940e20f30e  ip: 44.195.169.96 |
| --- |

***Install Docker using Ansible***

| ---  - name: Install Docker on EC2 instance  hosts: aws  become: true  tasks:  - name: Install Docker dependencies  apt:  name: "{{ item }}"  state: present  update\_cache: yes  loop:  - apt-transport-https  - ca-certificates  - curl  - gnupg2  - software-properties-common  - name: Add Docker GPG key  apt\_key:  url: https://download.docker.com/linux/ubuntu/gpg  state: present  - name: Add Docker repository  apt\_repository:  repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable  state: present  - name: Install Docker  apt:  name: docker-ce  state: present |
| --- |

***Install Kubernetes***

| *---*  *- hosts: all*  *become: true*  *vars:*  *kube\_version: "1.23.6-00"*  *pod\_network\_cidr: "10.244.0.0/16"*  *tasks:*  *- name: Add Kubernetes repository*  *apt\_key:*  *url: https://packages.cloud.google.com/apt/doc/apt-key.gpg*  *- name:*  *apt\_repository:*  *repo: deb https://apt.kubernetes.io/ kubernetes-xenial main*  *state: present*  *- name: Install Kubernetes components*  *apt:*  *name: "{{ item }}"*  *state: present*  *update\_cache: yes*  *loop:*  *- kubelet={{ kube\_version }}*  *- kubeadm={{ kube\_version }}*  *- kubectl={{ kube\_version }}* |
| --- |

***Initialize Master Node***

| sudo kubeadm init --ignore-preflight-errors=all |
| --- |

***Install Celico driver***

| ---  - hosts: kubernetes-master  become: true  tasks:  - name: Install calico network plugin  shell: sudo kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/calico.yaml |
| --- |

***Creating new user***

| kubectl create clusterrole pod-user --verb=create,list,get,update,delete --resource=pods  kubectl create clusterrolebinding <binding-name> --clusterrole=<cluster-role-name> --serviceaccount=<namespace>:<service-account-name>  kubectl --kubeconfig=/etc/kubernetes/admin.conf create role pod-user --verb=create,list,get,update,delete --resource=pods  kubectl --kubeconfig=/etc/kubernetes/admin.conf create rolebinding user4-access --user=user4 --role=pod-user  kubectl --kubeconfig=/home/certs/user4.conf get pods |
| --- |

***Take snapshot of ETCD backup***

| export RELEASE="3.3.13"  wget [https://github.com/etcd-io/etcd/releases/download/v${RELEASE}/etcd-v${RELEASE}-linux-amd64.tar.gz](https://github.com/etcd-io/etcd/releases/download/v$%7BRELEASE%7D/etcd-v$%7BRELEASE%7D-linux-amd64.tar.gz)  tar xvf etcd-v${RELEASE}-linux-amd64.tar.gz  cd etcd-v${RELEASE}-linux-amd64  Mkdir etcd-backup  ETCDCTL\_API=3 etcdctl --endpoints=172.31.8.70:2379 --cacert /etc/kubernetes/pki/etcd/ca.crt  --cert /etc/kubernetes/pki/etcd/server.crt --key /etc/kubernetes/pki/etcd/server.key snapshot save /etcd-backup/etcd-snapshot-latest.db |
| --- |

***Creating deployment, service and HPA***

| ---  apiVersion: apps/v1  kind: Deployment  metadata:  name: sb-app1  labels:  app: sb-app1  spec:  replicas: 2  selector:  matchLabels:  app: sb-app1  template:  metadata:  name: sb-app1  labels:  app: sb-app1  spec:  containers:  - name: sb-cont1  image: springio/gs-spring-boot-docker:latest  resources:  limits:  cpu: 100m ## 10% of 1 core on your system  ---  apiVersion: v1  kind: Service  metadata:  name: sb-service  spec:  type: NodePort ## this is default if we do not type in service definition  selector:  app: sb-app1  ports:  - protocol: TCP  port: 80  targetPort: 8080  nodePort: 32708  ---  apiVersion: autoscaling/v2  kind: HorizontalPodAutoscaler  metadata:  name: sb-hpa  spec:  scaleTargetRef:  apiVersion: apps/v1  kind: Deployment  name: sb-app1  minReplicas: 2  maxReplicas: 20  metrics:  - type: Resource  resource:  name: cpu  target:  type: Utilization  averageUtilization: 50 |
| --- |

***Jmeter Command used to run Load test***

| jmeter -n -t "E:\simplilearn\cap-stone\apache-jmeter-5.5\apache-jmeter-5.5\bin\Simplilearn\Simplilearn example.jmx" -l E:\simplilearn\cap-stone\apache-jmeter-5.5\apache-jmeter-5.5\bin\Simplilearn\ |
| --- |