**EKS cluster installation**

we need a two Roles iam ploicyes

AmazonEKSClusterPolicy

second policy (on ec2)

AmazonEC2ContainerRegistryReadOnly

AmazonEKS\_CNI\_Policy

AmazonEKSWorkerNodePolicy

we vpc and subnets

for that we are cloud formation template

https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml

Through this dedicated vpc and subnet create one instance

kubectl installation

1 apt update

2 sudo apt-get update && sudo apt-get install -y apt-transport-https gnupg2 curl

3 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

4 echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list

5 sudo apt-get update

6 sudo apt-get install -y kubectl

7 curl -LO https://dl.k8s.io/release/v1.29.2/bin/linux/amd64/kubectl

8 apt update

9 curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

10 apt update

11 curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

12 chmod +x kubectl

13 sudo mv kubectl /usr/local/bin

14 kubectl version

---------------------------------------------------------------------------------------------

aws cli installtion

15 curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

16 ll

17 apt install unzip

18 ll

19 unzip awscliv2.zip

20 sudo ./aws/install

21 cd ~/.aws/ (not required)

22 vi config

23 cd

24 aws --version

-----------------------------------------------------------------------------

aws configuration

25 aws configure

26 AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2CDR]: AKIA4HWJUI6MAJV6KYOH

27 AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ly84]: 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

28 Default region name [ap-southeast-2]: us-east-1

29 Default output format [None]:

---------------------------------------------------------------------------------------------

30 aws eks update-kubeconfig --name demo\_cluster --region ap-southeast-2

Added new context arn:aws:eks:ap-southeast-2:654654333490:cluster/demo\_cluster to /root/.kube/config

31 cd /root/.kube/

32 ll

33 vi config

34 cd

35 aws eks list-clusters

root@ip-172-31-3-103:~# aws eks list-clusters

{

"clusters": [

"demo\_cluster"

]

}

30 kubectl get pods

31 root@ip-172-31-14-229:~# kubectl get all

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 38m

32 kubectl get nodes

kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-192-168-62-80.ap-southeast-2.compute.internal Ready <none> 4m5s v1.29.3-eks-ae9a62a

ip-192-168-99-194.ap-southeast-2.compute.internal Ready <none> 4m v1.29.3-eks-ae9a62a

root@ip-172-31-14-229:~# kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 39m

kubectl get pods

not running any pods

vi myapp.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 3 # Number of pods to run

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest # You can specify a different version if needed

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

type: LoadBalancer # Exposes the service via a load balancer

ports:

- port: 80 # External port

targetPort: 80 # The port to forward to the container

selector:

app: nginx # Selects the pods with the label 'app: nginx'

:wq

save

take another terminal for watch purpose

sudo -i

watch -x kubectl get all

Every 2.0s: kubectl get all ip-172-31-14-229: Fri May 3 11:52:41 2024

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 95m

in main terminal

kubectl apply -f myapp.yaml

then it create a load balancer

watch in second terminal

Every 2.0s: kubectl get all ip-172-31-14-229: Fri May 3 12:29:27 2024

NAME READY STATUS RESTARTS AGE

pod/nginx-project-78b94b9cc8-2nd5d 1/1 Running 0 6m14s

pod/nginx-project-78b94b9cc8-cph25 1/1 Running 0 6m15s

pod/nginx-project-78b94b9cc8-krkql 1/1 Running 0 6m14s

pod/nginx-project-78b94b9cc8-n2l55 1/1 Running 0 6m14s

NAME TYPE CLUSTER-IP EXTERNAL-IP

PORT(S) AGE

service/kubernetes ClusterIP 10.100.0.1 <none>

443/TCP 131m

service/nginx-project LoadBalancer 10.100.159.124 afe2bc5538a0044289a623d3f1f1db43-1912667682.ap-southeast-2.elb.amazonaws.com

80:31550/TCP 6m15s

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/nginx-project 4/4 4 4 6m15s

NAME DESIRED CURRENT READY AGE

replicaset.apps/nginx-project-78b94b9cc8 4 4 4 6m15s

change the security group of load balanacer

sgr-0a81ea606f1b1f6dc ALLTCP 0-65535 CUSTOM 0.0.0.0/0

sgr-070ec3c2e80f020ff HTTP 80 CUSTOM 0.0.0.0/0

or

ssh 22

http 80 ports are allowed

nginx will display

then main server

kubectl get pods

kubectl get deployments

kubectl get services

kubectl delete service/nginx-project

kubectl delete deployment nginx-project