(1)AWS configure

* AKIAURLST6BDC7MJDJ5J
* ienSl2GYMDsxLAye5d7CwruuhGqBx/S9EhAr6AP/
* ap-northeast-2

(2) eksctl install

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

(3) kubectl install

curl -LO <https://dl.k8s.io/release/v1.23.6/bin/linux/amd64/kubectl>

chmod +x kubectl

sudo mv kubectl /usr/local/bin

(4) aws connect with ec2 bastion server

**aws eks update-kubeconfig --region ap-northeast-2 --name test-eks-cluster**

**(5) OIDC provider setting**

eksctl utils associate-iam-oidc-provider --cluster **test-eks-cluster** --approve

(6) install aws-load-balancer-controller

eksctl delete iamserviceaccount aws-load-balancer-controller --namespace=kube-system --cluster=test-eks-cluster

eksctl create iamserviceaccount \

--cluster=test-eks-cluster \

--namespace=kube-system \

--name=aws-load-balancer-controller \

--role-name AmazonEKSLoadBalancerControllerRole \

--attach-policy-arn=**arn:aws:iam::**058264082089**:policy/**test-alb-iam-policy \

--approve

(7) crd

kubectl apply -k github.com/aws/eks-charts/stable/aws-load-balancer-controller/crds?ref=master  
kubectl get crd

(8) install helm

curl -fsSL -o get\_helm.sh <https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3>

chmod 700 get\_helm.sh

./get\_helm.sh

(9)basic setting

sudo yum install git vim -y

(10) use helm

helm repo add eks <https://aws.github.io/eks-charts>

helm install aws-load-balancer-controller eks/aws-load-balancer-controller \

-n kube-system \

--set clusterName=test-eks-cluster \

--set serviceAccount.create=false \

--set serviceAccount.name=aws-load-balancer-controller

Check- >

kubectl get deployment -n kube-system aws-load-balancer-controller  
kubectl -n kube-system rollout status deployment aws-load-balancer-controller

Nlb deploy case

---

apiVersion: v1

kind: Namespace

metadata:

name: game-2048

---

apiVersion: apps/v1

kind: Deployment

metadata:

namespace: game-2048

name: deployment-2048

spec:

selector:

matchLabels:

app.kubernetes.io/name: app-2048

replicas: 5

template:

metadata:

labels:

app.kubernetes.io/name: app-2048

spec:

containers:

- image: public.ecr.aws/l6m2t8p7/docker-2048:latest

imagePullPolicy: Always

name: app-2048

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: nlb-sample-service

namespace: game-2048

annotations:

service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: ip

service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing

service.beta.kubernetes.io/aws-load-balancer-healthcheck-port: "8080"

service.beta.kubernetes.io/aws-load-balancer-cross-zone-load-balancing-enabled: "true"

spec:

ports:

- port: 80

targetPort: 80

protocol: TCP

type: LoadBalancer

loadBalancerClass: service.k8s.aws/nlb

selector:

app.kubernetes.io/name: app-2048

3. PVC

서비스 어카운트 생성

eksctl create iamserviceaccount --region ap-northeast-2 --namespace kube-system --cluster test-eks-cluster --name ebs-csi-controller-sa --role-name AmazonEKS\_EBS\_CSI\_DriverRole --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy --approve --role-only

Kubernetes 연동

eksctl create addon --name aws-ebs-csi-driver --cluster test-eks-cluster --service-account-role-arn arn:aws:iam::058264082089:role/[AmazonEKS\_EBS\_CSI\_DriverRole](https://us-east-1.console.aws.amazon.com/iam/home?region=ap-northeast-2#/roles/details/AmazonEKS_EBS_CSI_DriverRole) --force

test yaml

---

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: gp3

allowVolumeExpansion: true

provisioner: ebs.csi.aws.com

volumeBindingMode: WaitForFirstConsumer

parameters:

type: gp3

allowAutoIOPSPerGBIncrease: 'true'

encrypted: 'true'

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: ebs-claim

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 4Gi

storageClassName: gp3

---

apiVersion: v1

kind: Pod

metadata:

name: app

spec:

terminationGracePeriodSeconds: 3

containers:

- name: app

image: centos

command: ["/bin/sh"]

args: ["-c", "while true; do echo $(date -u) >> /data/out.txt; sleep 5; done"]

volumeMounts:

- name: persistent-storage

mountPath: /data

volumes:

- name: persistent-storage

persistentVolumeClaim:

claimName: ebs-claim

3. EFS

curl -o iam-policy-example.json <https://raw.githubusercontent.com/kubernetes-sigs/aws-efs-csi-driver/master/docs/iam-policy-example.json>

iam-policy-example -> terraform -> EFS\_CSI\_Driver\_Policy

서비스 어카운트 생성

eksctl create iamserviceaccount --name efs-csi-controller-sa --namespace kube-system --cluster test-eks-cluster --attach-policy-arn arn:aws:iam::058264082089:policy/EFS\_CSI\_Driver\_Policy --approve

helm repo add aws-efs-csi-driver https://kubernetes-sigs.github.io/aws-efs-csi-driver/

helm repo update

helm upgrade -i aws-efs-csi-driver aws-efs-csi-driver/aws-efs-csi-driver \

--namespace kube-system \

--set image.repository=602401143452.dkr.ecr.ap-northeast-2.amazonaws.com/eks/aws-efs-csi-driver \

--set controller.serviceAccount.create=false \

--set controller.serviceAccount.name=efs-csi-controller-sa

Pod.yaml

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: efs-claim

spec:

accessModes:

- ReadWriteMany

storageClassName: efs-sc

resources:

requests:

storage: 5Gi

---

apiVersion: v1

kind: Pod

metadata:

name: efs-app

spec:

containers:

- name: app

image: centos

command: ["/bin/sh"]

args: ["-c", "while true; do echo $(date -u) >> /data/out; sleep 5; done"]

volumeMounts:

- name: persistent-storage

mountPath: /data

volumes:

- name: persistent-storage

persistentVolumeClaim:

claimName: efs-claim

storage.yaml

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: efs-sc

provisioner: efs.csi.aws.com

parameters:

provisioningMode: efs-ap

fileSystemId: fs-001ae8ed2d19914b0

directoryPerms: "700"

gidRangeStart: "1000" # optional

gidRangeEnd: "2000" # optional

basePath: "/dynamic\_provisioning" # optional

kubectl exec -it <pod-name> -- /bin/bash