Kubernetes 节点弹性伸缩开源组件 Karpenter 实验

1. 配置环境变量

```
export CLUSTER_NAME="karpenter-clu"

export AWS_REGION="us-west-2"

export AWS_ACCOUNT_ID="$(aws sts get-caller-identity --query Account --output text)"
```

2. 创建 Cluster的配置文件

```
cat << EOF > cluster.yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
 name: ${CLUSTER_NAME}
 region: ${AWS_REGION}
 version: "1.21"
 tags:
   karpenter.sh/discovery: ${CLUSTER_NAME}

iam:
 withOIDC: true

managedNodeGroups:
 - name: ng-1
   privateNetworking: true
   instanceType: m5.large
   desiredCapacity: 3
EOF
```

3. 安装 eksctl 和 kubelet

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

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


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


curl -LO https://dl.k8s.io/release/v1.21.0/bin/linux/amd64/kubectl

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl


sudo curl --silent --location -o /usr/local/bin/kubectl \
   https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/
kubectl

sudo chmod +x /usr/local/bin/kubectl
```

4. 部署 cluster

```
eksctl create cluster -f cluster.yaml

export CLUSTER_ENDPOINT="$(aws eks describe-cluster --name ${CLUSTER_NAME} --query
"cluster.endpoint" --output text)"

echo $CLUSTER_ENDPOINT

[https://4F71450F00F2D92463DDE6BDF0A64B4C.gr7.us-west-2.eks.amazonaws.com](https://
4F71450F00F2D92463DDE6BDF0A64B4C.gr7.us-west-2.eks.amazonaws.com)
```

5. 配置karpenter用户角色

```
TEMPOUT=$(mktemp)

curl -fsSL https://karpenter.sh/v0.6.3/getting-started/cloudformation.yaml > $TEMPOUT \
&& aws cloudformation deploy \
 --stack-name "Karpenter-${CLUSTER_NAME}" \
 --template-file "${TEMPOUT}" \
 --capabilities CAPABILITY_NAMED_IAM \
 --parameter-overrides "ClusterName=${CLUSTER_NAME}"
```

eksctl-karpenter-addon-iamserviceaccount-kub-Role1-1G9PX8G86AXUL

身份提供商: arn:aws:iam::229474920661:oidc-provider/oidc.eks.us-west-2.amazonaws.com/id/7D0BD76F276A50872F4C306BEA8FDFB3

eksctl-karpenter-cluster-ServiceRole-1K4XCJW9OGGI8

AWS 服务: eks

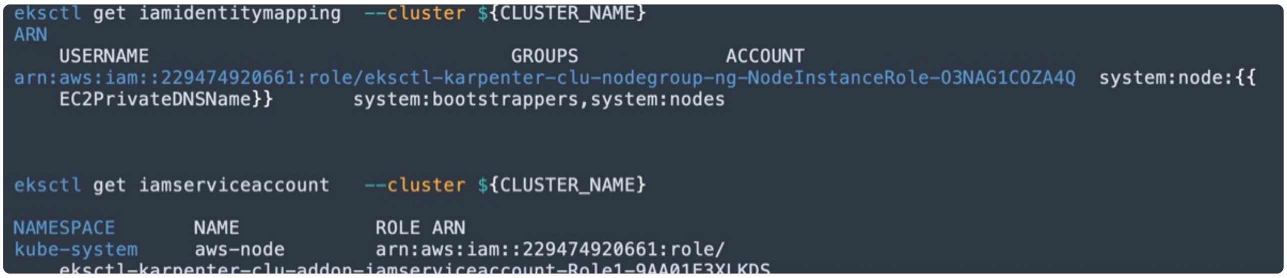
eksctl-karpenter-nodegroup-ng-1-NodeInstanceRole-1O8B30SYRDXT3

AWS 服务: ec2

KarpenterNodeRole-karpenter

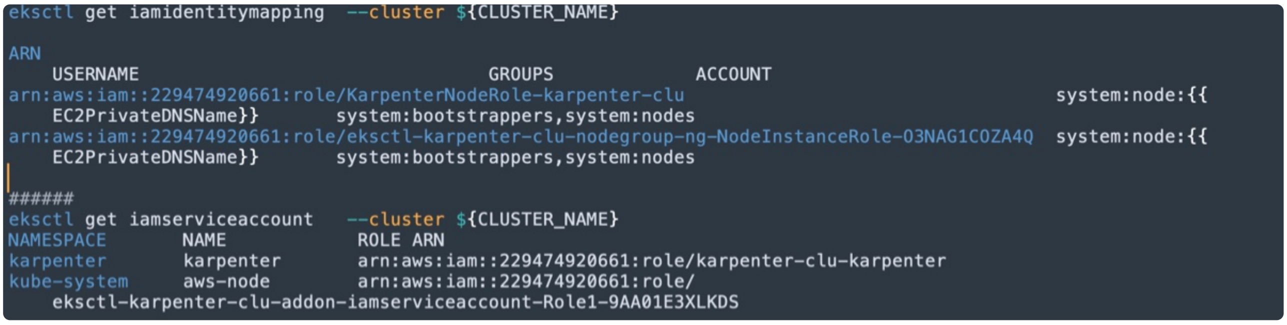
AWS 服务: ec2

6. 配置 iamidentitymapping 和 iamserviceaccount



```
eksctl get iamidentitymapping  --cluster ${CLUSTER_NAME}
ARN
    USERNAME                                        GROUPS                ACCOUNT
arn:aws:iam::229474920661:role/eksctl-karpenter-clu-nodegroup-ng-NodeInstanceRole-03NAG1C0ZA4Q  system:node:{{
    EC2PrivateDNSName}}        system:bootstrappers,system:nodes



eksctl get iamserviceaccount   --cluster ${CLUSTER_NAME}

NAMESPACE       NAME            ROLE ARN
kube-system     aws-node        arn:aws:iam::229474920661:role/
    eksctl-karpenter-clu-addon-iamserviceaccount-Role1-9AA01E3XLKDS
```

```
# 关联role
eksctl create iamidentitymapping \
 --username system:node:{{EC2PrivateDNSName}} \
 --cluster "${CLUSTER_NAME}" \
 --arn "arn:aws:iam::${AWS_ACCOUNT_ID}:role/KarpenterNodeRole-${CLUSTER_NAME}" \
 --group system:bootstrappers \
 --group system:nodes

#
eksctl utils associate-iam-oidc-provider --cluster ${CLUSTER_NAME} --approve

#
eksctl create iamserviceaccount \
 --cluster "${CLUSTER_NAME}" --name karpenter --namespace karpenter \
 --role-name "${CLUSTER_NAME}-karpenter" \
 --attach-policy-arn "arn:aws:iam::${AWS_ACCOUNT_ID}:policy/KarpenterControllerPolicy-$
{CLUSTER_NAME}" \
 --role-only \
 --approve
```



```
eksctl get iamidentitymapping  --cluster ${CLUSTER_NAME}

ARN
    USERNAME                                    GROUPS              ACCOUNT
arn:aws:iam::229474920661:role/KarpenterNodeRole-karpenter-clu                                  system:node:{{
    EC2PrivateDNSName}}        system:bootstrappers,system:nodes
arn:aws:iam::229474920661:role/eksctl-karpenter-clu-nodegroup-ng-NodeInstanceRole-03NAG1COZA4Q  system:node:{{
    EC2PrivateDNSName}}        system:bootstrappers,system:nodes

######
eksctl get iamserviceaccount   --cluster ${CLUSTER_NAME}
NAMESPACE       NAME            ROLE ARN
karpenter       karpenter       arn:aws:iam::229474920661:role/karpenter-clu-karpenter
kube-system     aws-node        arn:aws:iam::229474920661:role/
    eksctl-karpenter-clu-addon-iamserviceaccount-Role1-9AA01E3XLKDS
```

```
export KARPENTER_IAM_ROLE_ARN="arn:aws:iam::${AWS_ACCOUNT_ID}:role/${CLUSTER_NAME}-
karpenter"

echo $KARPENTER_IAM_ROLE_ARN
arn:aws:iam::229474920661:role/karpenter-clu-karpenter
```

7. 安装 helm 和 karpenter

```
# 安装helm
curl -sSL https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash


# 安装
helm repo add karpenter https://charts.karpenter.sh/
helm repo update

helm upgrade --install --namespace karpenter --create-namespace \
 karpenter karpenter/karpenter \
 --version v0.10.1 \
 --set serviceAccount.annotations."eks\.amazonaws\.com/role-arn"=$
{KARPENTER_IAM_ROLE_ARN} \
 --set clusterName=${CLUSTER_NAME} \
 --set clusterEndpoint=${CLUSTER_ENDPOINT} \
 --set aws.defaultInstanceProfile=KarpenterNodeInstanceProfile-${CLUSTER_NAME} \
 --set logLevel=debug \
 --wait


kubectl get deployment -n karpenter

kubectl get pods -n karpenter
```

8. 配置 Provisioner

```
cat << EOF > provisioner.yaml
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: cpu
spec:
  requirements:
    - key: karpenter.sh/capacity-type
      operator: In
      values: ["spot"]
    - key: node.kubernetes.io/instance-type
      operator: In
      values: ["m5.xlarge", "m5.2xlarge"]
  limits:
     resources:
      cpu: 1000
  provider:
    subnetSelector:
      karpenter.sh/discovery: karpenter-clu
   securityGroupSelector:
      kubernetes.io/cluster/karpenter-clu: owned
  ttlSecondsAfterEmpty: 30
EOF



kubectl get provisioners

kubectl describe provisioners
```

9. 使用karpenter扩容

```
# 扩容deployment
kubectl scale deployment eks-sample-linux-deployment --replicas 200 -n eks-sample-app

# 缩容
kubectl scale deployment eks-sample-linux-deployment --replicas 0 -n eks-sample-app

# 查看karpenter日志
kubectl logs -f -n karpenter -l app.kubernetes.io/name=karpenter -c controller
```