CICD说明文档

EKS CICD

1 构建环境

创建master

Plain Text

```
# master
cat >eks-cluster-role-trust-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
aws iam create-role --role-name AmazonEKSClusterRole --assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy --role-name AmazonEKSClusterRole
aws ec2 describe-vpcs --query "Vpcs[?InstanceTenancy=='default'].VpcId" --output text
aws ec2 describe-subnets --query "Subnets[?VpcId=='vpc-0068f2f40191aace5'].[SubnetId,AvailabilityZone]"

# cli  install
curl https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip -o awscliv2.zip
unzip awscliv2.zip
./aws/install
mv /usr/local/bin/aws /bin/aws
aws --version

# kubectl install
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
chmod +x kubectl
mv kubectl /usr/bin/

aws eks create-cluster --region us-east-1 --name adp-k8s-prod --kubernetes-version 1.28 \
   --role-arn arn:aws:iam::547384405015:role/AmazonEKSClusterRole \
   --resources-vpc-config subnetIds=subnet-01e82b74dc8d09b8c,subnet-0e2bffaba035bfd5f

  aws eks create-addon --cluster-name adp-k8s-prod --addon-name coredns
```

```
aws eks create-addon --cluster-name adp-k8s-prod --addon-name kube-proxy
aws eks create-addon --cluster-name adp-k8s-prod --addon-name vpc-cni
```

创建node

Plain Text

```
# OIDC
aws eks update-kubeconfig --region us-east-2 --name adp-k8s-prod
OIDC: https://docs.aws.amazon.com/eks/latest/userguide/enable-iam-roles-for-service-accounts.html

#node
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
  --role-name AmazonEKSNodeRole \
  --assume-role-policy-document file://"node-role-trust-relationship.json"
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name AmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name AmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name AmazonEKSNodeRole

# 控制台手动点击创建node
```

创建ebs provisioner（可选）

Plain Text

```
# EBS-driver
aws eks describe-cluster --name adp-k8s-prod --query "cluster.identity.oidc.issuer" --output text
cat > aws-ebs-csi-driver-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::547384405015:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/1B287476D5B0BA8D494474280FD01B0F"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region-code.amazonaws.com/id/1B287476D5B0BA8D494474280FD01B0F:aud": "sts.amazonaws.com",
          "oidc.eks.region-code.amazonaws.com/id/1B287476D5B0BA8D494474280FD01B0F:sub": "system:serviceaccount:kube-system:ebs-csi-controller-sa"
        }
      }
    }
  ]
}
EOF
# 替换region-code  和账号id 和OIDC码

aws iam create-role \
  --role-name AmazonEKS_EBS_CSI_DriverRole \
  --assume-role-policy-document file://"aws-ebs-csi-driver-trust-policy.json"

aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
  --role-name AmazonEKS_EBS_CSI_DriverRole

aws eks create-addon --cluster-name adp-k8s-prod --addon-name aws-ebs-csi-driver \
  --service-account-role-arn arn:aws:iam::547384405015:role/AmazonEKS_EBS_CSI_DriverRole

git clone https://github.com/kubernetes-sigs/aws-ebs-csi-driver.git
cd aws-ebs-csi-driver/examples/kubernetes/dynamic-provisioning/
```

```
38  echo "parameters:
39    type: gp3" >> manifests/storageclass.yaml
40  kubectl apply -f manifests/
41  kubectl get pv
```

kubesphere install

```
kubectl apply -f https://github.com/kubesphere/ks-installer/releases/download/v3.4.1/kubesphere-installer.yaml



kubectl apply -f https://github.com/kubesphere/ks-installer/releases/download/v3.4.1/cluster-configuration.yaml




apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: kubesphere-ingress
  namespace: kubesphere-system
  annotations:
    nginx.ingress.kubernetes.io/proxy-body-size: 600m
    nginx.org/client-max-body-size: "10m"
    nginx.ingress.kubernetes.io/proxy-read-timeout: "1800"
    nginx.ingress.kubernetes.io/proxy-send-timeout: "1800"
    nginx.ingress.kubernetes.io/websocket-services: proxy-public
    nginx.org/websocket-services: proxy-public
spec:
  rules:
  - host: ks.lvtujingji.click
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: ks-console
            port:
              number: 80
  ingressClassName: nginx
```

ingress nginx install

```
# ingress nginx

wget https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.8.2/deploy/static/provider/aws/nlb-with-tls-termination/deploy.yaml
aws acm request-certificate --domain-name *.lvtujingji.click --validation-method DNS
aws acm describe-certificate --certificate-arn arn:aws:acm:us-east-1:547384405015:certificate/6843d443-8907-4dee-99b5-ad8f45c27105

aws route53 list-hosted-zones

aws route53 change-resource-record-sets --hosted-zone-id Z03012041JX1FRPDJQ2NX --change-batch file://config.json
cat >> config.json << EOF
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "UPSERT",
      "ResourceRecordSet": {
        "Name": "ks.lvtujingji.click.",
        "Type": "CNAME",
        "TTL": 60,
        "ResourceRecords": [
          {
            "Value": ""
          }
        ]
      }
    }
  ]
}
EOF

需要修改 proxy-real-ip-cidr
        service.beta.kubernetes.io/aws-load-balancer-ssl-cert
kubectl  apply -f deploy.yaml

```

jenkins install

Plain Text

```
git clone https://github.com/scriptcamp/kubernetes-jenkins
kubectl create namespace devops-tools
kubectl apply -f serviceAccount.yaml

cat > volume.yaml << EOF
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: jenkins-pv-claim
  namespace: devops-tools
spec:
  storageClassName: local-storage
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 30Gi
EOF

# 替换local-storage ebs-sc
kubectl create -f volume.yaml
kubectl apply -f deployment.yaml
kubectl get deployments -n devops-tools

cat >> service.yaml << EOF
apiVersion: v1
kind: Service
metadata:
  name: jenkins-service
  namespace: devops-tools
  annotations:
      prometheus.io/scrape: 'true'
      prometheus.io/path:   /
      prometheus.io/port:   '8080'
spec:
  selector:
    app: jenkins-server
  type: ClusterIP
  ports:
    - port: 8080
      targetPort: 8080
      name: httpport
    - port: 50000
      targetPort: 50000
```

```
      name: jnlpport
EOF

# 可选
cat >> secret.yaml << EOF
---
apiVersion: v1
kind: Secret
metadata:
  name: jenkins-admin
  namespace: devops-tools
  annotations:
    kubernetes.io/service-account.name: "jenkins-admin"
type: kubernetes.io/service-account-token
EOF

cat >> jenkins-ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: jenkins-ingress
  namespace: devops-tools
spec:
  rules:
  - host: js.lvtujingji.click
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: jenkins-service
            port:
              number: 8080
  ingressClassName: nginx

kubectl apply -f  jenkins-ingress.yaml
```

2 配置jenkins

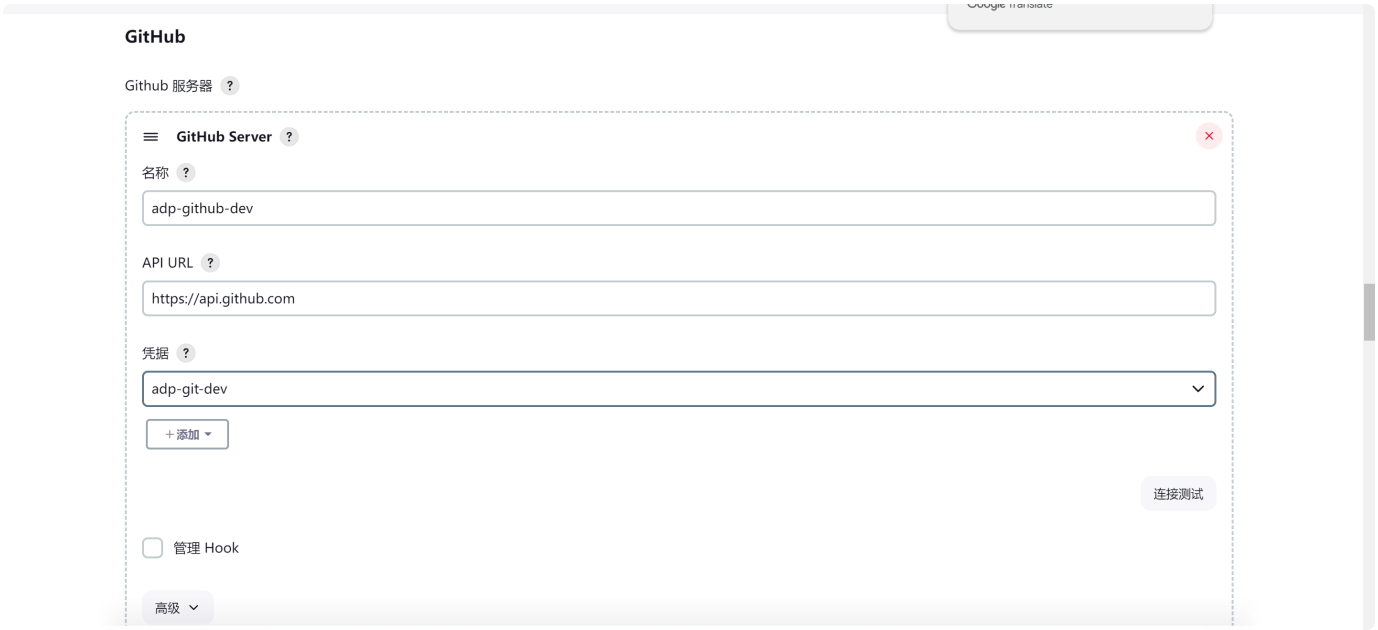
插件

```
Kubernetes plugin
Kubernetes CLI Plugin
Pipeline
GitHub plugin

```

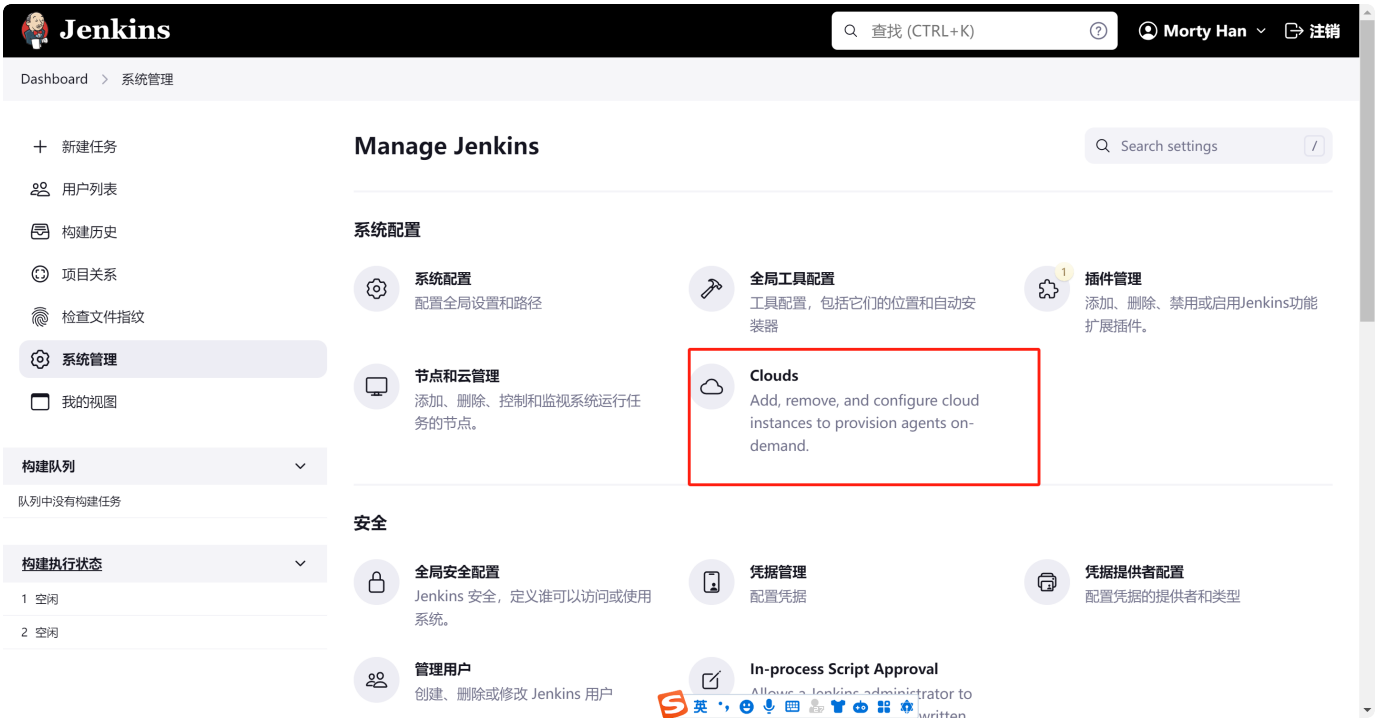
需要提前配置凭证

jenkins面板，系统管理，系统配置

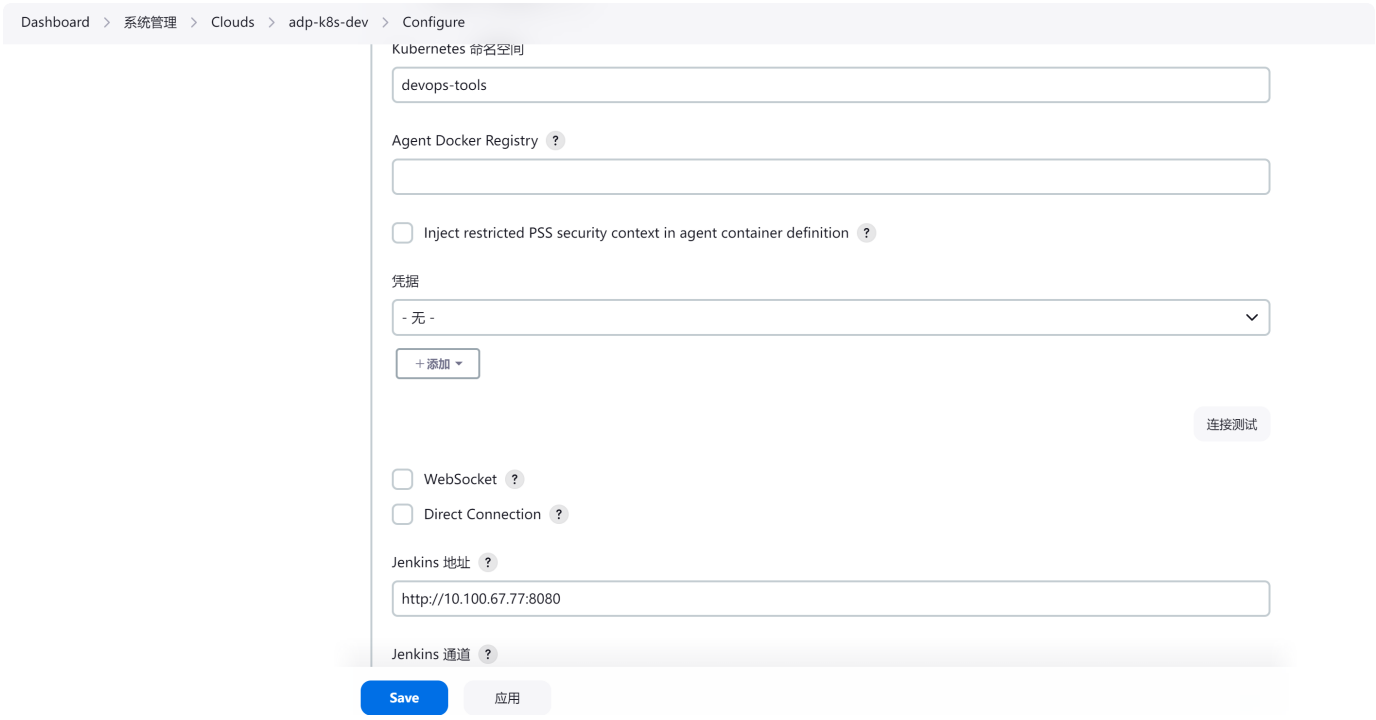


凭证管理里输入jenkins连接k8s凭证，连接github的凭证

添加cloud



需要调整的地方不多，配置下jenkins的地址就行，如果master jenkins是在eks中启动



需要提前在devops-tools名称空间创建ecrconfig configmap

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: ecrconfig
  namespace: devops-tools
data:
  config.json: |
    { "credsStore": "ecr-login"}

```

jenkins file

Plain Text

```
pipeline{
  agent{
    kubernetes{
      cloud 'adp-k8s-dev'
      yaml '''
apiVersion: v1
kind: Pod
metadata:
  name: kaniko
  namespace: devops-tools
spec:
  serviceAccountName: jenkins-admin
  containers:
    - name: kaniko
      image: gcr.io/kaniko-project/executor:debug
      env:
      - name: AWS_SDK_LOAD_CONFIG
        value: "true"
      command:
        - sleep
      args:
        - 99d
      volumeMounts:
      - name: ecrconfig
        mountPath: /kaniko/.docker/
  restartPolicy: Never
  volumes:
  - name: ecrconfig
    configMap:
      name: ecrconfig
'''
    }
  }
  parameters{
    string( name:'ENVIRONMENT',defaultValue:'dev',description:'Target environment (dev, test, prod)')
    string( name:'VERSION',defaultValue:'1.10',description:'Target Verison to deply')
  }
  stages('Begging Deply'){
    stage('Pull Code'){
      steps {
                // 使用 checkout 步骤拉取代码
                checkout([$class: 'GitSCM',
                          branches: [[name: 'main']],
```

```
                        doGenerateSubmoduleConfigurations: false,
                        extensions: [],
                        submoduleCfg: [],
                        userRemoteConfigs: [[credentialsId: 'adp-git-pass-dev', url: 'https://github.com/lvtujingji/lvtujingji.git']]])
                sh 'sed -i -E "/server_name/s/web/${ENVIRONMENT}/" lvtujingji.conf'
      }
    }
    stage('Build Image'){
      steps{
        container('kaniko'){
                sh "/kaniko/executor --context git://github.com/lvtujingji/lvtujingji.git#refs/heads/main --dockerfile dockerfile --destination 547384405015.dkr.ecr.us-east-1.amazonaws.com/adp-ecr-dev:nginx-v${params.VERSION}"
        }
      }
    }
    stage('Deploy !!!'){
      steps{
        script{
          sh 'curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl && chmod +x kubectl'
          if (params.ENVIRONMENT == 'prod') {
            sh 'sed -i -E "/image/s/nginx-v[0-9]?\\.[0-9]+/nginx-v${VERSION}/g" nginx-prod.yaml'
                        sh "./kubectl apply -f nginx-prod.yaml"
                    } else if (params.ENVIRONMENT  == 'test') {
            sh 'sed -i -E "/image/s/nginx-v[0-9]?\\.[0-9]+/nginx-v${VERSION}/g" nginx-test.yaml'
                        sh "./kubectl apply -f nginx-test.yaml"
                    } else {
            sh 'sed -i -E "/image/s/nginx-v[0-9]?\\.[0-9]+/nginx-v${VERSION}/g" nginx-dev.yaml'
                        sh "./kubectl apply -f nginx-dev.yaml"
                    }
        }
      }
    }
  }
}
```

3 结语

后续的deploy代码部署在https://github.com/lvtujingji/lvtujingji

```
[root@adp-jumpserver lvtujingji]# kubectl get pods -A
NAMESPACE                     NAME                                                READY   STATUS      RES
TARTS        AGE
adp-dev                       adp-nginx-dev-6f98584698-dgfkh                      1/1     Running     0
             17h
adp-dev                       adp-nginx-dev-6f98584698-s5vfn                      1/1     Running     0
             17h
adp-dev                       adp-nginx-dev-6f98584698-t7lrf                      1/1     Running     0
             17h
adp-prod                      adp-nginx-prod-77594c54cd-cv6zs                     1/1     Running     0
             16h
adp-prod                      adp-nginx-prod-77594c54cd-gjpd9                     1/1     Running     0
             16h
adp-prod                      adp-nginx-prod-77594c54cd-n95pm                     1/1     Running     0
             16h
adp-test                      adp-nginx-test-b97d5f4f6-fh2mn                      1/1     Running     0
             16h
adp-test                      adp-nginx-test-b97d5f4f6-gvq6j                      1/1     Running     0
             16h
adp-test                      adp-nginx-test-b97d5f4f6-nht7g                      1/1     Running     0
             16h
default                       app                                                 1/1     Running     0
             4d23h
devops-tools                  jenkins-bf6b8d5fb-s9jkr                             1/1     Running     1 (
4d21h ago)   4d22h
```