Container Instance Service

Operation Guide

Product Documentation

Contents

Operation Guide
virtual-kubelet Deployment Guide

Operation Guide
virtual-kubelet Deployment Guide

Last updated：2018-11-22 10:37:05

Preparation Before Use

The virtual-kubelet dependent file:

- qcloud-vkubelet.yaml
- virtual-kubelet.yaml
- config/config.toml
- config/server.crt
- config/server.key

```
[root@VM_168_12_centos virtual-kubelet]# ll
total 12
drwxr-xr-x 2 root root 4096 Jun  4 15:29 config
-rwxr-xr-x 1 root root  743 Jun  4 15:29 qcloud-vkubelet.yaml
-rwxr-xr-x 1 root root 1086 Jun  4 15:29 virtual-kubelet.yaml
[root@VM_168_12_centos virtual-kubelet]# ll config/
total 12
-rw-r--r-- 1 root root  209 Jun  4 15:29 config.toml
-rw-r--r-- 1 root root 1285 Jun  4 15:29 server.crt
-rw-r--r-- 1 root root 1675 Jun  4 15:29 server.key
```

For more information, see virtual kubelet deployment template . The template supports uploading TKE nodes, decompressing, and modifying specific parameters for direct use.

1. The virtual-kubelet startup configuration file config.toml

```
config.toml:
Region = "ap-guangzhou" #Indicates the created region where CIS resides
Zone = "ap-guangzhou-4" #Indicates the created availability zone where CIS resides
Version = "2018-04-08"
SecretId = "" #Indicates the CIS user's SecretId
SecretKey = ""#Indicates the CIS user's SecretKey
CPU = "100"
Memory = "100Gi"
Pods = "50"
```

Where, Region and Zone are in the formats of `Region = "Region"` and `Zone = "Availability zone"` respectively. Guangzhou Zone 4 is taken as an example in the above code.
For more information on availability zones, see Regions and Availability Zones.

2. The certfile and keyfile of the virtual-kubelet 10250 port: server.crt and server.key.
   This 10250 port is mainly used for kubectl logs. When we use kubectl logs to get Pod container logs, kube-apiserver will access the 10250 port of the node to get relevant information about the logs.
   In Tencent Cloud TKE service, 10250 port verification is not set, but kube-apiserver needs to access the node's 10250 port via HTTPS, otherwise kube-apiserver will report an error. Therefore, you need to set fake server.key and server.crt to implement kubectl logs feature.

3. The virtual-kubelet deployment file:
   Create virtual-kubelet's serviceaccount using qcloud-vkubelet.yaml to work with Pod and other resources:

```
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
name: vkubelet
subjects:
- kind: ServiceAccount
name: vkubelet
namespace: default
roleRef:
kind: ClusterRole
name: vkubelet
apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
name: vkubelet
labels:
k8s-app: vkubelet
rules:
- apiGroups: [""] # "" indicates the core API group
resources:
- namespaces
- pods
- pods/status
- nodes
- nodes/status
```

```
- secrets
- configmaps
verbs:
- create
- update
- get
- watch
- list
- delete
---
apiVersion: v1
kind: ServiceAccount
metadata:
name: vkubelet
namespace: default
labels:
k8s-app: vkubelet
---
```

Create a Pod using virtual-kubelet.yaml to run the virtual-kubelet project:

```
apiVersion: v1
kind: Pod
metadata:
name: virtual-kubelet
labels:
k8s-app: vkubelet
spec:
serviceAccountName: vkubelet
restartPolicy: "Never"
imagePullSecrets:
- name: qcloudregistrykey
containers:
- name: virtual-kubelet
image: ccr.ccs.tencentyun.com/tencentyun/virtual-kubelet:dev
imagePullPolicy: Always
env:
- name: KUBELET_PORT
value: "10250"
- name: APISERVER_CERT_LOCATION
value: /etc/virtual-kubelet/server.crt
- name: APISERVER_KEY_LOCATION
value: /etc/virtual-kubelet/server.key
- name: VKUBELET_POD_IP
```

```
valueFrom:
fieldRef:
fieldPath: status.podIP
volumeMounts:
- name: credentials
mountPath: "/etc/virtual-kubelet"
command: ["/usr/bin/virtual-kubelet"]
args: ["--provider", "qcloud", "--namespace", "default", "--provider-config", "/etc/virtual-kubelet/ config.toml"]
volumes:
- name: credentials
hostPath:
The path: /home/ubuntu/for-show/config (config folder contains config.toml, server.crt and server.key)
```

Procedure

1. Log in to the Kubernetes node server where kubectl is installed and initialized.
   For more information on how to install and initialize kubectl, see Manage Clusters with Kubectl.

2. Execute the following command:

   ```
   kubectl create -f qcloud-vkubelet.yaml
   ```

   Example of execution result:

   ```
   ubuntu@VM-66-110-ubuntu:~/for-show$ kubectl create -f qcloud-vkubelet.yaml
   clusterrolebinding "vkubelet" created
   clusterrole "vkubelet" created
   serviceaccount "vkubelet" created
   ```

3. Execute the following command:

   ```
   kubectl create -f virtual-kubelet.yaml
   ```

   Example of execution result:

```
ubuntu@VM-66-110-ubuntu:~/for-show$ kubectl create -f virtual-kubelet.yaml
pod "virtual-kubelet" created
ubuntu@VM-66-110-ubuntu:~/for-show$ kubectl get po
NAME READY STATUSRESTARTS AGE
virtual-kubelet 1/1 Running 0 3s
ubuntu@VM-66-110-ubuntu:~/for-show$ kubectl get no
NAME STATUS ROLES AGE VERSION
192.168.66.110 Ready none 3d v1.8.7-qcloud
192.168.66.16 Ready none 9d v1.8.7-qcloud
virtual-kubelet Ready agent 5s v1.8.3
```

Example and Notes

Example

```
ubuntu@VM-66-110-ubuntu:~/for-show$ cat busybox-pod-pass.yaml
apiVersion: v1
kind: Pod
metadata:
name: busybox
annotations:
kubernetes.io/cis.vpcId: vpc-lpaa5xe3
kubernetes.io/cis.subnetId: subnet-7z46i306
labels:
qcloud-app: busybox
spec:
containers:
- image: busybox
imagePullPolicy: Always
name: busybox
resources:
requests:
memory: 1Gi
cpu: "1"
limits:
memory: 1Gi
cpu: "1"
command: ["/bin/sh"]
args: ["-c", "while true; do echo hello world; sleep 2; done"]
dnsPolicy: ClusterFirst
nodeName: virtual-kubelet
```

Notes

1. Specify both IDs of VPC and subnet where CIS runs.

```
kubernetes.io/cis.vpcId: vpc-lpaa5xe3
kubernetes.io/cis.subnetId: subnet-7z46i306
```

2. Specify the specification supported to run CIS. Note that request and limit should be consistent.

```
resources:
requests:
memory: 1Gi
cpu: "1"
limits:
memory: 1Gi
cpu: "1"
```

3. Specify virtual-kubelet as the node where CIS runs.

```
nodeName: virtual-kubelet
```