

容器服务

远程终端

产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

远程终端

远程终端概述

远程终端基本操作

其他容器登录方式

kubectl 命令行介绍

kubectl 工具安装

创建 Service

创建 Ingress

创建 PV&PVC

远程终端

远程终端概述

最近更新时间：2017-07-14 16:32:55

远程终端概述

远程终端帮助您快速调试容器，连接容器查看问题，支持复制粘贴、上传下载文件功能，解决用户登录容器路径长、调试难的问题。

使用帮助

- [远程终端的基本操作](#)
- [其他容器登录方式](#)

远程终端基本操作

最近更新时间：2018-05-30 14:40:45

远程终端连接到容器

1. 登录 [腾讯云容器服务控制台](#)。
2. 单击左侧导航栏中的【服务】，单击服务名称（如图中 test-cfs）。
3. 在【实例列表】下，单击实例名称(如图中 xxx-3001348546-xxx)。
4. 单击【远程终端】，登录到远程终端。

未安装 shell 的容器运行命令

1. 进入远程终端页面。
2. 在下方输入要执行的命令，单击确认。

文件上传下载

1. 进入远程终端页面。
2. 单击下方的文件助手，选择上传或下载文件。
 - 上传需指定上传的文件目录。
 - 下载需指定下载的文件的路径。

其他容器登录方式

最近更新时间：2018-07-05 16:00:04

通过web终端登录到容器(推荐)

- 1.单击进入[容器服务控制台页面](#)。
- 2.选择容器所属服务，单击进入详情，查看实例列表，选择容器并连接远程终端。
- 3.更多远程终端常见问题单击[查看详情](#)

实例名称	状态	监控	主机IP	实例IP	重启次数	运行时间	创建时间	操作
nginx-772167675-6ayzj	运行中		10.0.0.136	172.20.0.7	0	2d 0h	2017-03-14 15:42:02	删除
容器ID: 777573c1669cfa397d2cc49558ce9e51a69a66dabdaf9cc61d								远程终端
nginx-772167675-n928	运行中		10.0.0.49	172.20.1.5	0	2d 0h	2017-03-14 15:42:32	删除
容器ID: 49275b90cea5e1bfc702dc4fbb2ff492223fb6994e61964b504								远程终端
nginx-772167675-v2zuq	运行中		10.0.0.136	172.20.0.6	0	2d 0h	2017-03-14 15:41:22	删除

通过容器所在节点登录到容器

- 1.获取容器所在节点IP地址，容器ID。

容器ID	状态	主机IP	容器IP	运行时间	重启次数
4c468d61e539a12c63ea86ed9cb44f1a8e5bd0a8e3198...	运行中	10.0.0.60	100.64.0.7	2016-11-10 20:52:13	0
49d6bc869cd7a758616d51b06a3ffb5e626f11d83634a...	运行中	10.0.0.107	100.64.1.5	2016-11-10 20:50:16	0
fd9bd2a28340b3110822f4ecd263d917d70304ddd37e...	运行中	10.0.0.107	100.64.1.6	2016-11-10 20:52:13	0
4c23084ca3ea2278c5f798758be0907698fc5ebeca3d23...	运行中	10.0.0.60	100.64.0.8	2016-11-10 20:52:13	0
fde731efe1c1ca4c88209ee367e89a19e1b80fb3792f52...	运行中	10.0.0.60	100.64.0.5	2016-11-10 20:39:55	0
7585e4eea603e93500a8736b5e632a524dda89044bf...	运行中	10.0.0.107	100.64.1.8	2016-11-10 20:52:13	0

2.登录到节点，详情查看[登录到云主机](#)。

3.通过docker ps命令查看需登录的容器。

```
[root@VM_88_88_centos ~]# docker ps | grep 75b3b15af61a
75b3b15af61a      nginx:latest      "nginx -g 'daemon off'" About a minute ago Up Abc
```

4.通过docker exec命令登录到容器。

```
[root@VM_0_60_centos ~]# docker ps | grep 75b3b15af61a
75b3b15af61a      nginx:latest      "nginx -g 'daemon off'" 2 minutes ago Up 2 minu
[root@VM_0_60_centos ~]# docker exec -it 75b3b15af61a /bin/bash
root@worid-24bn2:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

容器已安装SSH服务端，通过SSH登录到容器

1.获取容器IP地址。



The screenshot shows the Tencent Cloud Container Service console. The 'Container List' tab is selected, displaying a table of containers. The 'Container IP' column for the container 'e2a846f36a1bef15b179510225558d13e03fea8a783cf82...' is highlighted with a red box and a red arrow, showing the IP address 100.64.0.4.

容器ID	状态	主机IP	容器IP	运行时间	重启次数
e2a846f36a1bef15b179510225558d13e03fea8a783cf82...	运行中	10.0.0.60	100.64.0.4	2016-11-14 10:46:13	0

2.登录集群内任意节点,详情查看[登录到云主机](#)。

3.通过SSH登录到容器。

通过管理节点登录到容器

敬请期待。

kubectl 命令行介绍

最近更新时间：2018-07-06 16:36:58

kubectl 命令行介绍

kubectl 是一个用于对 Kubernetes 集群操作命令行工具。本文涵盖 kubectl 语法，常见命令操作，并提供常见示例。有关每个命令（包括所有主命令和子命令）的详细信息，请参阅 [kubectl 参考文档](#) 或使用 kubectl help 命令查看详细帮助，有关安装说明，请参阅 [安装 kubectl](#)。

通过 kubectl 创建 Nginx

以下示例帮助您熟悉运行常用 kubectl 操作：

kubectl create

由于 Nginx 名称容易冲突，从文件或标准输入创建资源。

```
$ kubectl create -f nginx.yaml # nginx-test.yaml文件见下
```

nginx-test.yaml 文件如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-test
  labels:
    qcloud-app: nginx-test
spec:
  replicas: 1
  revisionHistoryLimit: 5
  selector:
    matchLabels:
      qcloud-app: nginx-test
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        qcloud-app: nginx-test
    spec:
```



```
containers:
- image: nginx:latest
  imagePullPolicy: Always
  name: nginx-test
  resources:
    limits:
      cpu: 200m
      memory: 128Mi
    requests:
      cpu: 200m
      memory: 128Mi
  securityContext:
    privileged: false
  serviceAccountName: ""
  volumes: null
  imagePullSecrets:
  - name: qcloudregistrykey
status: {}
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-test
  labels:
    qcloud-app: nginx-test
spec:
  ports:
  - name: tcp-80-80-ogxxh
    nodePort: 0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    qcloud-app: nginx-test
  type: LoadBalancer
status:
  loadBalancer: {}
```

```
root@kubectld-1180979449-j2qtb:/# kubectl create -f nginx-test.yaml
deployment "nginx-test" created
service "nginx-test" created
root@kubectld-1180979449-j2qtb:/#
```

通过以下命令获取外网访问地址：

```
kubectl get services
```

```
root@kubectld-1180979449-b8kkr:/# kubectl get service
NAME          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
nginx         172.16.255.15   111.230.         80:32218/TCP     3m
root@kubectld-1180979449-b8kkr:/#
```

在浏览器上输入该 IP，便可直接访问 nginx 服务。

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

kubectl logs

在容器中打印容器的日志。

```
// Return a snapshot of the logs from pod <pod-name>.
$ kubectl logs <pod-name>
```

```
root@kubectld-1180979449-j2qtb:/# kubectl get pod
NAME          READY    STATUS    RESTARTS   AGE
kubectld-1180979449-j2qtb  1/1      Running   0          1h
nginx-test-4016666277-q44kg 1/1      Running   0          1m
root@kubectld-1180979449-j2qtb:/# kubectl logs nginx-test-4016666277-q44kg
10.0.0.104 - - [05/Feb/2018:09:54:36 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5
10.0.0.104 - - [05/Feb/2018:09:54:36 +0000] "GET /favicon.ico HTTP/1.1" 404 571 "ht
537.36" "-"
2018/02/05 09:54:36 [error] 6#6: *1 open() "/usr/share/nginx/html/favicon.ico" fail
146.34", referer: "http://203.195.146.34/"
10.0.0.112 - - [05/Feb/2018:09:54:43 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5
10.0.0.105 - - [05/Feb/2018:09:54:43 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5
10.0.0.104 - - [05/Feb/2018:09:54:43 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5
root@kubectld-1180979449-j2qtb:/#
```

kubectl 工具安装

最近更新时间：2018-07-06 16:37:58

安装 kubectl 工具

如果您已经安装 kubectl 工具，请忽略本步骤。详细安装 kubectl 过程参考 [Installing and Setting up kubectl](#)。

下载 kubectl

腾讯云服务器内网下载

注：腾讯云容器服务创建的云服务器，默认已安装 kubectl 工具。

- Linux 系统

```
curl -LO http://mirrors.tencentyun.com/install/ccs/v1.4.6/linux/amd64/kubectl
```

- Windows 系统

```
curl -LO http://mirrors.tencentyun.com/install/ccs/v1.4.6/windows/amd64/kubectl.exe
```

公网下载

- MAC OS X 系统

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/${curl -s https://storage.googlea
```

- Linux 系统

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/${curl -s https://storage.googlea
```

- Windows 系统

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/${curl -s https://storage.googlea
```

添加执行权限

使用以下命令添加执行权限：

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

测试

测试安装结果，输入以下命令，输出版本信息即安装成功。

```
kubectl version
```

```
Client Version: version.Info{Major:"1", Minor:"5", GitVersion:"v1.5.2", GitCommit:"08e099554f3c31f6e6f07
```

获取集群账号密码以及证书信息

1. 登录 [容器服务控制台](#) > [集群](#)，单击需要连接的集群 ID/名称，查看集群详情。
2. 在集群信息页，单击【显示凭证】，查看用户名、密码和证书信息。
3. 复制或下载证书文件到本地。

通过证书信息使用 kubectl 操作集群

单次 kubectl 操作请求，附带证书信息

该方法适用于单次操作集群，不将容器集群的证书信息保存到机器上。

请求方法，kubectl 命令格式：

```
-s "域名信息" --username=用户名 --password=密码 --certificate-authority=证书路径
```

示例：

```
kubectl get node -s "https://cls-66668888.ccs.tencent-cloud.com" --username=admin --password=6666c
```

修改 kubectl 配置文件，长期有效

该方法适用于长期通过 kubectl 操作集群，一次配置，只要文件不修改就长期有效。

设置 kubectl 配置，修改以下命令中的密码、证书信息。

```
kubectl config set-credentials default-admin --username=admin --password=6666o9oIB2gHD88882qu  
kubectl config set-cluster default-cluster --server=https://cls-66668888.ccs.tencent-cloud.com --certifica  
kubectl config set-context default-system --cluster=default-cluster --user=default-admin  
kubectl config use-context default-system
```

配置完成，直接使用 kubectl 命令：

```
kubectl get nodes
```

NAME	STATUS	AGE
10.0.0.61	Ready	10h

设置 kubectl 命令自动补全

您可以通过配置 kubectl 自动补全，提高可使用性。

```
source <(kubectl completion bash)
```

创建 Service

最近更新时间：2018-07-06 16:42:25

创建公网服务

使用以下命令创建 mysvc.yaml 文件：

```
$ kubectl create -f mysvc.yaml
```

mysvc.yaml 文件内容如下：

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  type: LoadBalancer
```

创建内网服务

创建通过 VPC 内网访问的服务，只需在 metadata 下的 annotations 中增加 **service.kubernetes.io/qcloud-loadbalancer-internal-subnetid** 即可，即创建的内网 LB 所在子网的唯一 ID，此子网必须在集群 VPC 下。

命令如下：

```
metadata:
  annotations:
    service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: subnet-xxxxxxx
```

带宽上移用户

带宽上移用户在创建公网访问方式的服务时需要指定如下两个 annotations 项：

- **service.kubernetes.io/qcloud-loadbalancer-internet-charge-type**
公网带宽计费方式，可选值有：TRAFFIC_POSTPAID_BY_HOUR（按使用流量计费），
BANDWIDTH_POSTPAID_BY_HOUR（按带宽计费）。
- **service.kubernetes.io/qcloud-loadbalancer-internet-max-bandwidth-out**
带宽上限，范围：[1,2000] Mbps。

创建命令：

```
metadata:
  annotations:
    service.kubernetes.io/qcloud-loadbalancer-internet-charge-type: TRAFFIC_POSTPAID_BY_HOUR
    service.kubernetes.io/qcloud-loadbalancer-internet-max-bandwidth-out: "10"
```

创建 Ingress

最近更新时间：2018-07-06 16:42:41

创建公网 Ingress

使用以下命令创建 myingress.yaml 文件：

```
$ kubectl create -f myingress.yaml
```

myingress.yaml 文件内容如下：

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: qcloud
  name: my-ingress
  namespace: default
spec:
  rules:
  - host: localhost
    http:
      paths:
      - backend:
          serviceName: non-service
          servicePort: 65535
        path: /
```

创建内网 Ingress

内网 Ingress 只需在 metadata 下的 annotations 中增加 **kubernetes.io/ingress.subnetId** 即可，即创建的内网 LB 所在子网的唯一 ID，此子网必须在集群 VPC 下。

```
metadata:
  annotations:
    kubernetes.io/ingress.subnetId: subnet-xxxxxxx
```


带宽上移用户

带宽上移用户在创建公网 Ingress 时需要指定如下两个 annotations 项：

- **kubernetes.io/ingress.internetChargeType**

公网带宽计费方式，可选值有：TRAFFIC_POSTPAID_BY_HOUR（按使用流量计费），
BANDWIDTH_POSTPAID_BY_HOUR（按带宽计费）。

- **kubernetes.io/ingress.internetMaxBandwidthOut**

带宽上限，范围：[1,2000] Mbps。

创建命令如下：

```
metadata:
  annotations:
    kubernetes.io/ingress.internetChargeType: TRAFFIC_POSTPAID_BY_HOUR
    kubernetes.io/ingress.internetMaxBandwidthOut: "10"
```

创建 PV&PVC

最近更新时间：2018-07-06 16:45:26

当前仅支持 CBS 类型的 PV&PVC，敬请期待 CFS 和 COS 存储类型支持。

静态创建 CBS 类型 PV&PVC

1. 创建 PV (可选)

通过已有 CBS 创建 PV。若未通过本步骤创建 PV，直接执行步骤 2 时，创建 PVC 将自动创建对应的 PV。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nginx-pv
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  qcloudCbs:
    cbsDiskId: disk-xxxxxxx ## 指定已有的CBS id
    fsType: ext4
    storageClassName: cbs
```

2. 创建 PVC

创建命令如下：

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: nginx-pv-claim
spec:
  storageClassName: cbs
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

- 普通云盘大小必须是 10 的倍数，最小为 10，最大为 4000。

- 高效云盘最小为 50 GB。
- SSD 云硬盘最小为 200 GB，具体策略见 [云硬盘文档](#)。

3. 使用 PVC

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      qcloud-app: nginx-deployment
  template:
    metadata:
      labels:
        qcloud-app: nginx-deployment
    spec:
      containers:
        - image: nginx
          imagePullPolicy: Always
          name: nginx
          volumeMounts:
            - mountPath: "/opt/"
              name: pvc-test
      volumes:
        - name: pvc-test
          persistentVolumeClaim:
            claimName: nginx-pv-claim # 已经创建好的pvc
```

动态创建 CBS 类型 PV&PVC

1. 创建 StorageClass

如果不创建 StorageClass，集群内将默认存在 name 为 cbs 的 StorageClass。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  # annotations:
  # storageclass.beta.kubernetes.io/is-default-class: "true"
```

```
# 如果有这一条，则会成为default-class，创建pvc时不指定类型则自动使用此类型
name: cloud-premium
provisioner: cloud.tencent.com/qcloud-cbs
parameters:
  type: cloudPremium
# 支持 cloudBasic,cloudPremium,cloudSSD 如果不识别则当做cloudBasic
# zone:ap-shanghai-1
# zone 支持指定zone，如果指定，则讲云盘创建到此zone，如果不指定，则拉取所有node的zone信息，随机
# aspid:asp-123
# 支持指定快照策略，创建云盘后自动绑定此快照策略,绑定失败不影响创建
```

2. 创建多实例 StatefulSet

使用云硬盘 CBS 创建多实例 StatefulSet：

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      terminationGracePeriodSeconds: 10
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
              name: web
          volumeMounts:
            - name: www
              mountPath: /usr/share/nginx/html
      volumeClaimTemplates: # 自动创建pvc，进而自动创建pv
        - metadata:
            name: www
          spec:
            accessModes: [ "ReadWriteOnce" ]
            storageClassName: cloud-premium
            resources:
```

```
requests:  
  storage: 10Gi
```