

云资源自动化 for Crossplane

快速入门



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

快速入门

最近更新时间：2024-11-11 11:35:11

预置条件

在开始使用本快速入门指南之前，请确保满足以下预置条件：

- 腾讯云凭证：在使用 Crossplane 之前，您需要拥有有效的腾讯云 SecretId 和 SecretKey。如果您还没有这些凭证，请前往 [访问管理控制台](#) 的 API 密钥管理页面申请安全凭证。
- 腾讯云账号：您需要拥有创建 [VPC（私有网络）](#) 的权限。
- kubectl：您需要能够正常访问您的 Kubernetes 集群，详情请参见 [安装 kubectl](#)。
- Helm：您需要安装 Helm，并确保版本为 v3.2.0 或更高版本，详情请参见 [安装 Helm](#)。

创建集群

集群创建有多种方式，这里介绍通过 [kind](#) 快速创建一个本地 Kubernetes 集群。

更多方式，请参见 [结合 minikube 使用 CrossPlane](#)。

安装 kind

请参见 [kind 官方文档](#) 的 Installation 步骤完成安装。

kind 创建集群需要依赖 [docker](#)，请确保已经提前安装。

创建集群

在终端中执行以下命令，创建集群。

```
kind create cluster
```

稍等片刻，将创建默认集群 `kind`。

```
> kind create cluster
Creating cluster "kind" ...
✓ Ensuring node image (kindest/node:v1.29.2) 📦
✓ Preparing nodes 📦
✓ Writing configuration 📄
✓ Starting control-plane 🚦
✓ Installing CNI 🛠️
✓ Installing StorageClass 🗄️
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a nice day! 🙌
```

(可选) 查看集群状态

执行命令 `kubectl cluster-info --context kind-kind`，查看当前集群的状态：
如果您的环境存在多个 Kubernetes 集群，请参见 [在多个集群上下文中切换](#) 进行管理。

```
> kubectl cluster-info --context kind-kind
Kubernetes control plane is running at https://127.0.0.1:60425
CoreDNS is running at https://127.0.0.1:60425/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-
info dump'.
```

安装 Crossplane

执行下列命令，将 Crossplane 安装到当前集群中。

```
helm repo add crossplane-stable https://charts.crossplane.io/stable && \
helm repo update && \
helm install crossplane \
--namespace crossplane-system \
--create-namespace crossplane-stable/crossplane
```

验证 Crossplane 安装

执行以下命令，验证 Crossplane 是否成功安装。

```
kubectl get pods -n crossplane-system
```

安装成功后，您可以看到 `crossplane` 和 `crossplane-rbac-manager` 两个正在运行的 pod。

NAME	READY	STATUS	RESTARTS	AGE
crossplane-6494656b8b-qr62x	1/1	Running	0	30s
crossplane-rbac-manager-8458557cdd-1fn8q	1/1	Running	0	30s

安装 TencentCloud Provider

执行以下命令，将腾讯云 Provider（如 v0.8.1，点击获取 [更多版本](#)）的配置信息部署到集群中：

```
cat <<EOF | kubectl apply -f -
apiVersion: pkg.crossplane.io/v1
kind: Provider
metadata:
  name: provider-tencentcloud
spec:
  package: xpkg.upbound.io/crossplane-contrib/provider-
tencentcloud:v0.8.1
EOF
```

(可选) 此外, 也可以将上述配置信息写入 `provider.yml` 文件中, 然后执行以下命令:

```
kubectl apply -f provider.yml
```

验证 Provider 安装

执行以下命令, 验证 Provider 是否成功安装。

```
kubectl get providers
```

返回信息如下所示 (版本号可能存在差异), 则表示 Provider 安装成功。

NAME	INSTALLED	HEALTHY	PACKAGE
AGE			
provider-tencentcloud	True	True	
xpkg.upbound.io/crossplane-contrib/provider-tencentcloud:v0.8.1			22h

配置 Provider 密钥

创建密钥

执行以下命令, 通过 Kubernetes 配置文件, 创建密钥:

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: tencentcloud-aksk
  namespace: crossplane-system
type: Opaque
stringData:
```

```
credentials: |
  {
    "secret_id": "*****",
    "secret_key": "*****",
    "region": "ap-xxxxxx"
  }
EOF
```

输出信息如下：

```
secret/tencentcloud-aksk created
```

为 Provider 配置密钥

执行以下命令，通过 Kubernetes 配置文件，配置 Provider 密钥：

```
cat <<EOF | kubectl apply -f -
apiVersion: tencentcloud.crossplane.io/v1alpha1
kind: ProviderConfig
metadata:
  name: default
spec:
  credentials:
    source: Secret
    secretRef:
      name: tencentcloud-aksk
      namespace: crossplane-system
      key: credentials
EOF
```

输出信息如下：

```
providerconfig.tencentcloud.crossplane.io/default created
```

使用 Crossplane 创建腾讯云 VPC 资源

执行以下命令，通过 Kubernetes 配置文件，创建 VPC：

```
cat <<EOF | kubectl apply -f -
apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
kind: VPC
```

```
metadata:
  name: resource-vpc
  namespace: crossplane-system
spec:
  forProvider:
    cidrBlock: 10.1.0.0/16
    name: crossplane-test-vpc
EOF
```

输出信息如下:

```
vpc.vpc.tencentcloud.crossplane.io/resource-vpc created
```

验证创建的资源

执行命令 `kubectl get vpc`，输出如下信息代表 VPC 创建成功:

```
> kubectl get vpc
NAME                READY    SYNCED    EXTERNAL-NAME    AGE
resource-vpc       True     True      vpc-dhckvylj     13s
```

此时，您可以在 [腾讯云控制台](#) 查看创建的 VPC 资源。

私有网络 广州5 私有网络帮助文档

私有网络API 已全面升级至3.0版本，基于2.0版本接口访问时延较高和使用复杂的考虑，原私有网络API 2.0 接口服务将不再提供技术支持，并将于2023年3月31日起下线。如果您的业务还在使用私有网络 API 2.0 相关接口，建议尽快将服务升级至API 3.0 接口，以免对您的业务造成影响。具体迁移指引详见《私有网络API2.0切换API3.0指引》。

新建 请输入私有网络 ID/名称

ID/名称	IPv4 CIDR	IPv6 CIDR	子网	路由表	NAT 网关	VPN 网关	云服务器	专线网关	默认私有网络	创建时间	标签	操作
vpc-dhckvylj crossplane-system/vpc	10.1.0.0/16	-	0	1	0	0	0	0	否	2024-01-17 11:29:11		删除 更多