

# 云资源自动化 for Crossplane

## 实践教学



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。

## 文档目录

### 实践教程

构建 CVM 资源

构建 MySQL 资源

构建 CKafka 资源

构建 SecurityGroup 资源

# 实践教程

## 构建 CVM 资源

最近更新时间：2024-05-20 18:23:32

利用 Crossplane 的 XRD 特性，我们可以通过 Composition 来定义自定义的组合资源，将需要的资源组合在一起部署和管理；通过在 XRD 中定义和配置对应的 API，提供该组合资源对外暴露的参数。

本文介绍如何创建一个包含 CVM、VPC 和对应子网的组合资源 XCvm。为了构建 XCvm 资源，需要创建并定义 xrd.yaml、composition.yaml、xcvm.yaml，然后执行这三个配置文件。

### 定义 XRD

首先我们需要创建自定义 API，即为后面创建的复合资源，定义该复合资源对应的参数和类型。

我们在 xrd.yaml 定义了 vpcName，subnetName，availabilityZone 作为必选参数，用户使用该复合资源时，必须提供这三个参数，用于创建 VPC 和对应子网。其余如 imageId，availabilityZone，instanceType，instanceChargeType 为可选参数。

```
# xrd.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xcvm.crd.tencentcloud.crossplane.io
spec:
  group: crd.tencentcloud.crossplane.io
  names:
    kind: XCvm
    plural: xcvm
  versions:
    - name: v1alpha1
      served: true
      referenceable: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties: # 参数定义
                vpcName:
                  type: string
                subnetName:
                  type: string
```

```

    imageId:
      type: string
      default: "img-9qrfy1xt" # 默认值
    availabilityZone:
      type: string
    instanceType:
      type: string
    instanceChargeType:
      type: string
    required: # 必选参数
      - vpcName
      - subnetName
      - availabilityZone
    status:
      description: A Status represents the observed state
      properties:
        share:
          description: Freeform field containing status information
          type: object
          x-kubernetes-preserve-unknown-fields: true
      type: object
    
```

## 定义 Composition

接着，在 `composition.yaml` 中定义复合资源，包括一个 VPC，一个子网，和在该 VPC 下的 CVM 实例。用户可在资源创建后，通过 `status.share` 获取 CVM 的实例 Id。

```

# composition.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: composition-cvm
spec:
  compositeTypeRef:
    apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
    kind: XCvm
  resources:
    - name: vpc
      base:
        apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
        kind: VPC
        spec:
          forProvider:
            cidrBlock: "10.1.0.0/16"
    
```

```
patches:
  - fromFieldPath: spec.vpcName
    toFieldPath: spec.forProvider.name
  - type: ToCompositeFieldPath
    fromFieldPath: status.atProvider.id
    toFieldPath: status.share.vpcId

- name: subnet
  base:
    apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
    kind: Subnet
    spec:
      forProvider:
        cidrBlock: "10.1.2.0/24"
  patches:
    - fromFieldPath: status.share.vpcId # 使用上面创建的vpc
      toFieldPath: spec.forProvider.vpcId
    - fromFieldPath: spec.subnetName
      toFieldPath: spec.forProvider.name
    - fromFieldPath: spec.availabilityZone
      toFieldPath: spec.forProvider.availabilityZone
    - type: ToCompositeFieldPath
      fromFieldPath: status.atProvider.id
      toFieldPath: status.share.subnetId

- name: cvm
  base:
    apiVersion: cvm.tencentcloud.crossplane.io/v1alpha1
    kind: Instance
    spec:
      forProvider:
        instanceName: test-crossplane-cvm
        # imageId: "img-9qrfy1xt"
  patches:
    - fromFieldPath: status.share.vpcId # 使用上面创建的vpc
      toFieldPath: spec.forProvider.vpcId
    - fromFieldPath: status.share.subnetId # 使用上面创建的subnet
      toFieldPath: spec.forProvider.subnetId
    - fromFieldPath: spec.availabilityZone
      toFieldPath: spec.forProvider.availabilityZone
    - fromFieldPath: spec.imageId
      toFieldPath: spec.forProvider.imageId
    - type: ToCompositeFieldPath
      fromFieldPath: status.atProvider.id
      toFieldPath: status.share.cvmId # cvm 实例Id
```

## 创建 XCvm 资源

在 `xcvm.yaml` 中为复合资源设置子网名称和可用区。

```
# xcvm.yaml
apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
kind: XCvm
metadata:
  name: xcvm-example # 复合资源前缀
spec:
  vpcName: xvpcName # 指定VPC和子网名称
  subnetName: xsubnetName
  availabilityZone: ap-guangzhou-3 # 指定可用区
  # imageUrl: img-xxxx # 使用指定镜像 Id
```

使用 `kubectl apply` 命令依次执行上述配置文件，完成资源的创建。

```
kubectl apply -f xrd.yaml
kubectl apply -f composition.yaml
kubectl apply -f xcvm.yaml
```

执行结果如下：

创建了前缀为 `xcvm-example-` 的 VPC、子网和 CVM 实例。

```
> kubectl get managed
NAME                                READY SYNCED EXTERNAL-NAME AGE
instance.cvm.tencentcloud.crossplane.io/xcvm-example-75vc2 True True ins-
7d5rb9us 3m4s

NAME                                READY SYNCED EXTERNAL-NAME AGE
subnet.vpc.tencentcloud.crossplane.io/xcvm-example-zlshp True True subnet-
2hmf8go 5m

NAME                                READY SYNCED EXTERNAL-NAME AGE
vpc.vpc.tencentcloud.crossplane.io/xcvm-example-rd5gq True True vpc-3fo55dx
5m
```

此时，您可在控制台查看刚才创建的 [CVM 资源](#)。

**HOT** 加入会员, 享续费折上9折起! 月月领专属优惠券!

实例 广州 1 其它地域 产品体验, 您说了算 场景教学 限时福利 实例使用指南

**安全提醒公告:** 若您使用XZ压缩工具的用户, 或者您的云服务器安装了XZ压缩工具, 建议您按照文档 [关于XZ-Utils漏洞风险\(CVE-2024-3094\)的检测与修复](#) 指引进行漏洞排查或修复。

新建 开机 关机 重启 续费 重置密码 销毁/退还 更多操作
切换到页签视图

多个关键字只支持精准查询, 用竖线“|”分隔, 多个过滤标签用回车键分隔

ID/名称	监控	状态	可用区	实例类型	实例配置	主IPv4地址	主IPv6地址	实例计费模式	网络计费模式	所属项目	操作
ins-7454b9us test-crossplane-cvm		运行中	广州三区	标准型S4	2核 2GB 0Mbps 系统盘: 高性能云硬盘 网络: xvpcName	- 12.3 (内)	-	按量计费 2024-04-23 00:08:21创建	-	默认项目	<a href="#">登录</a> <a href="#">更多</a>

## 查看 VPC 资源。

私有网络 广州 6 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

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-3ofta55dx xvpcName	10.0.0.0/16	-	1	1	0	0	1	0	否	2024-04-23 19:50:43		<a href="#">删除</a> <a href="#">更多</a>

## 查看 子网资源。

子网 广州 14 全部私有网络 子网帮助文档

新建

ID/名称	所属网络	CIDR	IPv6 CIDR	可用区	关联路由表	云服务器	可用IP	默认子网	创建时间	标签	操作
subnet-01m0d8go xsubnetName	vpc-31t55dx xvpcName	10.10.0.0/24	-	广州三区	rtb-790x4f3c default	1	252	否	2024-04-23 18:07:16		<a href="#">删除</a> <a href="#">更多</a>



# 构建 MySQL 资源

最近更新时间：2024-05-20 18:23:32

本文介绍如何创建一个包含 MySQL、VPC 和对应子网的组合资源 XMySQL。为了构建 XMySQL 资源，需要创建并定义 xrd.yaml、composition.yaml、xmysql.yaml 和 secret.yaml，然后执行这三个配置文件。

## 定义 XRD

首先我们需要创建自定义 API，即为后面创建的复合资源，定义该复合资源对应的参数和类型。

我们在 xrd.yaml 定义了 MySQL 相关的配置（例如规格，内存，计费类型等）和 VPC、子网资源配置。

其中 vpcName，subnetName，availabilityZone 为必选参数，用于创建 VPC 和对应子网。

```
# xrd.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xmysqls.crd.tencentcloud.crossplane.io
spec:
  group: crd.tencentcloud.crossplane.io
  names:
    kind: XMySQL
    plural: xmysqls
  versions:
    - name: v1alpha1
      served: true
      referenceable: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                vpcName:
                  type: string
                subnetName:
                  type: string
                availabilityZone:
                  type: string
                instanceName:
                  type: string
                  default: test-crossplane-mysql # 实例默认值
                chargeType:
```

```
type: string
default: POSTPAID # 默认按量计费
intranetPort:
  type: integer
  default: 3306 # 默认端口
parameters:
  type: object
properties:
  maxConnections:
    type: integer
    default: 1000 # 默认最大链接数
  character_set_server:
    type: string
    default: utf8 # 默认字符集
  volumeSize:
    type: integer
    default: 200 # 默认容量
  memSize:
    type: integer
    default: 4000 # 默认内存值
  sgId:
    type: string
  securityGroups:
    type: array
    items:
      type: string
  required:
    - vpcName
    - subnetName
    - availabilityZone
  status:
    description: A Status represents the observed state
    properties:
      share:
        description: Freeform field containing status information
        type: object
        x-kubernetes-preserve-unknown-fields: true
    type: object
```

## 定义 Composition

接着，在 `composition.yaml` 中定义复合资源，包括一个 VPC，一个子网，和在该 VPC 下的 MySQL 实例。用户可在资源创建后，通过 `status.share` 获取 MySQL 的实例 Id。

```
# composition.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: composition-mysql
spec:
  compositeTypeRef:
    apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
    kind: XMySQL
  resources:
    # vpc 配置
    - name: vpc
      base:
        apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
        kind: VPC
        spec:
          forProvider:
            cidrBlock: "10.1.0.0/16"
      patches:
        - fromFieldPath: spec.vpcName
          toFieldPath: spec.forProvider.name
        - type: ToCompositeFieldPath
          fromFieldPath: status.atProvider.id
          toFieldPath: status.share.vpcId

    # 子网配置
    - name: subnet
      base:
        apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
        kind: Subnet
        spec:
          forProvider:
            cidrBlock: "10.1.2.0/24"
      patches:
        - fromFieldPath: status.share.vpcId
          toFieldPath: spec.forProvider.vpcId
        - fromFieldPath: spec.subnetName
          toFieldPath: spec.forProvider.name
        - fromFieldPath: spec.availabilityZone
          toFieldPath: spec.forProvider.availabilityZone
        - type: ToCompositeFieldPath
          fromFieldPath: status.atProvider.id
          toFieldPath: status.share.subnetId

    - name: mysql
```

```
base:
  apiVersion: mysql.tencentcloud.crossplane.io/v1alpha1
  kind: Instance
  spec:
    forProvider:
      rootPasswordSecretRef: # password 配置
        key: password
        namespace: crossplane-system
        name: mysql-root-password
  patches:
    - fromFieldPath: status.share.vpcId # 使用上面创建的 VPC
      toFieldPath: spec.forProvider.vpcId
    - fromFieldPath: status.share.subnetId # 使用上面创建的子网
      toFieldPath: spec.forProvider.subnetId
    - fromFieldPath: spec.securityGroups
      toFieldPath: spec.forProvider.securityGroups
    - fromFieldPath: spec.chargeType
      toFieldPath: spec.forProvider.chargeType
    - fromFieldPath: spec.availabilityZone
      toFieldPath: spec.forProvider.availabilityZone
    - fromFieldPath: spec.instanceName
      toFieldPath: spec.forProvider.instanceName
    - fromFieldPath: spec.volumeSize
      toFieldPath: spec.forProvider.volumeSize
    - fromFieldPath: spec.memSize
      toFieldPath: spec.forProvider.memSize
    - fromFieldPath: spec.intranetPort
      toFieldPath: spec.forProvider.intranetPort
    - fromFieldPath: spec.parameters
      toFieldPath: spec.forProvider.parameters
    - type: ToCompositeFieldPath
      fromFieldPath: status.atProvider.id
      toFieldPath: status.share.mysqlId # 实例 Id
```

然后，在 `secret.yaml` 中定义 MySQL 实例的密码：

```
# secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-root-password
  namespace: crossplane-system
stringData:
  password: my-secure-password
```

## 创建 XMySQL 资源

现在，在 `xmysql.yaml` 中为复合资源设置子网和实例配置。

```
# xmysql.yaml
apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
kind: XMySQL
metadata:
  name: xmysql-example # 复合资源前缀
  namespace: crossplane-system
spec:
  # 指定子网和可用区
  vpcName: test-crossplane-vpc
  subnetName: test-crossplane-subnet
  availabilityZone: ap-guangzhou-3
  # 实例配置
  volumeSize: 30
  memSize: 2000
  instanceName: test-crossplane-mysql-instance
  # 安全组配置
  securityGroups:
    - sg-xxxxxx
```

使用 `kubectl apply` 命令依次执行上述配置文件，完成资源的创建。

```
kubectl apply -f xrd.yaml
kubectl apply -f secret.yaml
kubectl apply -f composition.yaml
kubectl apply -f xmysql.yaml
```

执行结果如下：

创建了前缀为 `xmysql-example-` 的 VPC，子网和 MySQL 实例。

```
> kubectl get managed
NAME                                     READY SYNCED EXTERNAL-NAME AGE
instance.mysql.tencentcloud.crossplane.io/xmysql-example-bqg56 True True cdb-071eyvvt 3m6s

NAME                                     READY SYNCED EXTERNAL-NAME AGE
subnet.vpc.tencentcloud.crossplane.io/xmysql-example-b5jmh True True subnet-qmac9t04 16m
```

NAME	READY	SYNCED	EXTERNAL-NAME	AGE
vpc.vpc.tencentcloud.crossplane.io/xmysql-example-pc7dt	True	True	vpc-	vpc-124cxb6p 16m

此时，您可在控制台查看刚才创建的 [MySQL 资源](#)。

MySQL - 实例列表 广州 3 其他地域 0 新功能速速 65 异常告警 用户指南

MySQL 全新节点版已支持上海、北京、广州、成都、中国香港、新加坡、法兰克福地域，使用全新云原生架构，满足测试、开发、个人网站等场景的需求，**极致性价比**。[了解详情](#)

[新建](#) [一键诊断](#) [对比监控](#) [重启](#) [续费](#) [更多操作](#)

多个关键字用竖线“|”分隔，多个过滤标签用回车键分隔

实例 ID / 名称	监控 / 状态 / 任务	可用区	配置	数据库版本	引擎	内网地址	计费模式	所属项目	操作
cdb-cxxyvvt test-crossplane-mysql-instance	运行中	广州三区	双节点(本地盘) 通用型-1核2000MB/30GB 网络: test-crossplane-vpc - test-crossplane-subnet	MySQL5.7	InnoDB	1.1.5-3306	按量计费	默认项目	<a href="#">登录</a> <a href="#">管理</a> <a href="#">更多</a>

查看 [VPC 资源](#)。

私有网络 广州 6 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版本接口访问时延迟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-cx6p test-crossplane-vpc	10.0.0/16	-	1	1	0	0	0	0	否	2023-04-23 14:23		<a href="#">删除</a> <a href="#">更多</a>

查看 [子网资源](#)。

子网 广州 16 test-crossplane-vpc(10.1.0.0/16) 子网帮助文档

[新建](#)

请输入子网 ID/名称

ID/名称	所属网络	CIDR	IPv6 CIDR	可用区	关联路由表	云服务器	可用IP	默认子网	创建时间	标签	操作
subnet-9t04 test-crossplane-subnet	vpc-cx6p test-crossplane-vpc	10.1.2.0/24	-	广州三区	rtb-3ujm	0	251	否	2023-04-23 14:48		<a href="#">删除</a> <a href="#">更多</a>

# 构建 CKafka 资源

最近更新时间：2024-05-20 18:23:32

本文介绍如何创建一个包含 CKafka、VPC和对应子网的复合资源 XCKafka。为了构建 XCKafka 资源，需要创建并定义 xrd.yaml、composition.yaml、xckafka.yaml，然后执行这三个配置文件。

## 定义 XRD

首先我们需要创建自定义 API，即为后面创建的复合资源，定义该复合资源对应的参数和类型。

我们在 xrd.yaml 定义了 MySQL 相关的配置（如：规格，内存，计费类型等）和 VPC、子网资源配置。

其中 vpcName，subnetName，zoneId，chargeType 为必选参数，用于创建 VPC、子网和 Kafka 实例。

```
# xrd.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xckafka.crd.tencentcloud.crossplane.io
spec:
  group: crd.tencentcloud.crossplane.io
  names:
    kind: XCKafka
    plural: xckafka
  versions:
    - name: v1alpha1
      served: true
      referenceable: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                vpcName: # VPC名称
                  type: string
                subnetName: # 子网名称
                  type: string
                vpcAvailabilityZone: # VPC可用区
                  type: string
                  default: ap-guangzhou-3
                chargeType: # 计费类型
                  type: string
```

```
zoneld: # Ckafka可用区，需要与VPC可用区匹配
  type: integer
diskSize: # 磁盘容量
  type: integer
diskType: # 磁盘类型
  type: string
bandWidth: # 带宽
  type: integer
partition: # 分区
  type: integer
instanceName: # 实例名称
  type: string
  default: test-crossplane-ckafka-instance
period: # 计费时长，用于包年包月类型
  type: string
required:
  - vpcName
  - subnetName
  - zoneld
  - chargeType
status:
  description: A Status represents the observed state
  properties:
    share:
      description: Freeform field containing status information
      type: object
      x-kubernetes-preserve-unknown-fields: true
  type: object
```

## 定义 Composition

接着，在 `composition.yaml` 中定义复合资源，包括一个 VPC，一个子网，和在该 VPC 下的 MySQL 实例。用户可在资源创建后，通过 `status.share` 获取 MySQL 的实例 Id。

```
# composition.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: composition-ckafka
spec:
  compositeTypeRef:
    apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
    kind: XCkafka
  resources:
```



```
- name: vpc
  base:
    apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
    kind: VPC
    spec:
      forProvider:
        cidrBlock: "10.1.0.0/16"
  patches:
    - fromFieldPath: spec.vpcName
      toFieldPath: spec.forProvider.name
    - type: ToCompositeFieldPath
      fromFieldPath: status.atProvider.id
      toFieldPath: status.share.vpcId

- name: subnet
  base:
    apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
    kind: Subnet
    spec:
      forProvider:
        cidrBlock: "10.1.2.0/24"
  patches:
    - fromFieldPath: status.share.vpcId # 使用上面定义的 VPC
      toFieldPath: spec.forProvider.vpcId
    - fromFieldPath: spec.subnetName
      toFieldPath: spec.forProvider.name
    - fromFieldPath: spec.vpcAvailabilityZone # VPC 的可用区
      toFieldPath: spec.forProvider.availabilityZone
    - type: ToCompositeFieldPath
      fromFieldPath: status.atProvider.id
      toFieldPath: status.share.subnetId

- name: ckafka
  base:
    apiVersion: kafka.tencentcloud.crossplane.io/v1alpha1
    kind: Instance
    spec:
      forProvider:
        kafkaVersion: 2.4.1
  patches:
    - fromFieldPath: status.share.vpcId
      toFieldPath: spec.forProvider.vpcId
    - fromFieldPath: status.share.subnetId
      toFieldPath: spec.forProvider.subnetId
    - fromFieldPath: spec.zoneId
```

```
toFieldPath: spec.forProvider.zoneld
- fromFieldPath: spec.chargeType
  toFieldPath: spec.forProvider.chargeType
- fromFieldPath: spec.instanceName
  toFieldPath: spec.forProvider.instanceName
- fromFieldPath: spec.diskSize
  toFieldPath: spec.forProvider.diskSize
- fromFieldPath: spec.bandWidth
  toFieldPath: spec.forProvider.bandWidth
- fromFieldPath: spec.partition
  toFieldPath: spec.forProvider.partition
- fromFieldPath: spec.diskType
  toFieldPath: spec.forProvider.diskType
- type: ToCompositeFieldPath
  fromFieldPath: status.atProvider.id
  toFieldPath: status.share.ckafkaId # 实例Id
```

## 创建 XKafka 资源

现在，在 `xckafka.yaml` 中为复合资源设置子网和实例配置。

```
# xckafka.yaml
apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
kind: XKafka
metadata:
  name: xckafka-example # 复合资源前缀
spec:
  # 指定子网和可用区
  vpcName: test-crossplane-vpc
  subnetName: test-crossplane-subnet
  vpcAvailabilityZone: ap-guangzhou-7
  # 实例配置
  zoneld: 100007
  chargeType: POSTPAID_BY_HOUR
  instanceName: test-crossplane-ckafka-instance
  diskSize: 200
  bandWidth: 20
  partition: 400
  diskType: CLOUD_BASIC
```

使用 `kubectl apply` 命令依次执行上述配置文件，完成资源的创建。

```
kubectl apply -f xrd.yaml
```

```
kubectl apply -f composition.yaml
kubectl apply -f xckafka.yaml
```

执行结果如下：

创建了前缀为 `xckafka-example-` 的 VPC，子网和 CKafka 实例。

```
> kubectl get managed
NAME                                READY SYNCED EXTERNAL-NAME
AGE
instance.kafka.tencentcloud.crossplane.io/xckafka-example-2fpsm False False
ckafka-xjvq5d5p 6m31s

NAME                                READY SYNCED EXTERNAL-NAME AGE
subnet.vpc.tencentcloud.crossplane.io/xckafka-example-gdjb4 True True subnet-
8ylo7i5s 6m31s

NAME                                READY SYNCED EXTERNAL-NAME AGE
vpc.vpc.tencentcloud.crossplane.io/xckafka-example-z8x5z True True vpc-
kkmy0ikp 6m31s
```

此时，您可在控制台查看刚才创建的 [CKafka 资源](#)。

ID/名称	监控	状态	可用区	实例类型	配置	网络类型	计费模式	公网计费模式	标签	操作
ckafka-xjvq5d5p test-crossplane-ckafka-instance		健康	广州七区	专业版 版本: 2.4.1 磁盘类型: 高性能云硬盘	Topic数量上限: 200个 Partition数量上限: 400个 峰值带宽: 20 MB/s 磁盘容量: 200GB	私有网络 test-vvwsplane-vpc test-vvwsplane-subnet	带宽包: 按小时后付费 存储类型: 固定存储	按小时计费		<a href="#">配置告警</a> <a href="#">升配</a> <a href="#">更多</a>

共 1 条

查看 [VPC 资源](#)。

私有网络 广州 6 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 Q 🔍 🔗 ⌵

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-4479y0ikp test-crossplane-vpc	10.1.0.0/16	-	1	1	0	0	0	0	否	2024-04-23 17:44		删除 更多

### 查看 子网资源。

子网 广州 16 全部私有网络 子网帮助文档

新建 请输入子网 ID/名称 Q 🔍 🔗 ⌵

ID/名称	所属网络	CIDR	IPv6 CIDR	可用区	关联路由表	云服务器	可用IP	默认子网	创建时间	标签	操作
subnet-14k75s test-crossplane-subnet	vpc-4479y0ikp test-crossplane-vpc	10.1.2.0/24	-	广州七区	rtb-2jyz19ie default	0	253	否	2024-04-23 17:09		删除 更多

# 构建 SecurityGroup 资源

最近更新时间：2024-05-20 18:23:32

本文介绍如何创建一个包含安全组和对应规则的组合资源 XSecurityGroup。为了构建 XSecurityGroup 资源，需要创建并定义 xrd.yaml、composition.yaml、xsecuritygroup.yaml，然后执行这三个配置文件。

## 定义 XRD

首先需要创建自定义 API，即为后面创建的复合资源，定义该复合资源对应的参数和类型。

在 xrd.yaml 定义安全组的入站和出站配置。其中 sgIngress，sgEgress 为必选参数。

```
# xrd.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xsecuritygroups.crd.tencentcloud.crossplane.io
spec:
  group: crd.tencentcloud.crossplane.io
  names:
    kind: XSecurityGroup
    plural: xsecuritygroups
  versions:
    - name: v1alpha1
      served: true
      referenceable: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                sgIngress: # 定义入站规则
                  type: array
                  items:
                    type: string
                sgEgress: # 定义出站规则
                  type: array
                  items:
                    type: string
          required:
            - sgIngress
            - sgEgress
```

```
status:
  description: A Status represents the observed state
  properties:
    share:
      description: Freeform field containing status information
      type: object
      x-kubernetes-preserve-unknown-fields: true
  type: object
```

## 定义 Composition

接着，在 `composition.yaml` 中定义复合资源，包括一个安全组和安全组规则。

用户可在资源创建后，通过 `status.share.sgId` 获取安全组 Id。

```
# composition.yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: composition-sg
spec:
  compositeTypeRef:
    apiVersion: crd.tencentcloud.crossplane.io/v1alpha1
    kind: XSecurityGroup
  resources:
    # 定义安全组
    - name: sg
      base:
        apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
        kind: SecurityGroup
        spec:
          forProvider:
            name: test-crossplane-sg # 安全组名称
            description: Security Group for the XSecurityGroup.
      patches:
        - type: ToCompositeFieldPath
          fromFieldPath: status.atProvider.id
          toFieldPath: status.share.sgId

    # 定义安全组规则
    - name: sgrules
      base:
        apiVersion: vpc.tencentcloud.crossplane.io/v1alpha1
        kind: SecurityGroupLiteRule
        spec:
```

```
patches:  
  - fromFieldPath: status.share.sgId  
    toFieldPath: spec.forProvider.securityGroupId  
  - fromFieldPath: spec.sgIngress  
    toFieldPath: spec.forProvider.ingress  
  - fromFieldPath: spec.sgEgress  
    toFieldPath: spec.forProvider.egress  
  - type: ToCompositeFieldPath  
    fromFieldPath: status.atProvider.id  
    toFieldPath: status.share.sgruleId
```

## 创建 XSecurityGroup 资源

现在，在 `xsecuritygroup.yaml` 中为复合资源设置具体的安全组规则。

```
# xsecuritygroup.yaml  
apiVersion: crd.tencentcloud.crossplane.io/v1alpha1  
kind: XSecurityGroup  
metadata:  
  name: xsg-example  
spec:  
  # 进站规则  
  sgIngress:  
    - "DROP#0.0.0.0/0#ALL#ALL" # 根据需求设置规则  
  # 出站规则  
  sgEgress:  
    - "DROP#0.0.0.0/0#ALL#ALL" # 根据需求设置规则
```

使用 `kubectl apply` 命令依次执行上述配置文件，完成资源的创建。

```
kubectl apply -f xrd.yaml  
kubectl apply -f composition.yaml  
kubectl apply -f xsecuritygroup.yaml
```

执行结果如下：

创建了前缀为 `xsg-example-` 的安全组实例。

```
> kubectl get managed  
NAME                                READY SYNCED EXTERNAL-NAME  
AGE  
securitygroupliterule.vpc.tencentcloud.crossplane.io/xsg-example-bks7q  True  True  
sg-3o46faav  5h48m
```

```

NAME                                READY SYNCED EXTERNAL-NAME AGE
securitygroup.vpc.tencentcloud.crossplane.io/xsg-example-j6fxz True True sg-
3o46faav 5h48m
    
```

此时，您可以在控制台查看创建的 [安全组资源](#)。



安全组 广州 8 所有项目 安全组帮助文档

新建 删除 编辑标签 分配至项目 关键字: sg-3o46faav

ID/名称	关联实例数	备注	类型	更新时间 ↓	创建时间	项目	标签	操作
找到 1 条结果 <a href="#">返回原列表</a>								
sg-3o46faav crossplane-sg	0	自定义	自定义	2024-11-21 02:15:43	2024-11-21 17:39:37	默认项目		<a href="#">修改规则</a> <a href="#">管理实例</a> <a href="#">更多</a>