容器服务

访问管理

产品文档

腾讯云

文档目录

访问管理

概述

最近更新时间：2017-10-17 16:43:26

如果您在腾讯云中使用到了容器服务（ CCS，Cloud Container Service ）该服务由不同的人管理，但都共享您的云账号密钥，将存在以下问题：

- 您的密钥由多人共享，泄密风险高。
- 您无法限制其他人的访问权限，易被其他人误操作造成安全风险。

这个时候，您就可以通过子帐号实现不同的人管理不同的服务，以避免以上的问题。默认情况下，子帐号没有使用 CCS 的权限。因此，我们就需要创建策略来允许子帐号拥用他们所需要的权限。

概述

访问管理（ CAM，Cloud Access Management ）是腾讯云提供的一套 Web 服务，它主要用于帮助客户安全管理腾讯云账户下的资源的访问权限。通过 CAM，您可以创建、管理和销毁用户(组)，并通过身份管理和策略管理控制哪些人可以使用哪些腾讯云资源。

当您使用 CAM 的时候，可以将策略与一个用户或者一组用户关联起来，策略能够授权或者拒绝用户使用指定资源完成指定任务。有关 CAM 策略的更多相关基本信息，请参照 策略语法。有关 CAM 策略的更多相关使用信息，请参照 策略。
如果您不需要对子账户进行 CAM 相关资源的访问管理，您可以跳过此章节。跳过这些部分并不影响您对文档中其余部分的理解和使用。

入门

CAM 策略必须授权使用一个或多个 CCS 操作或者必须拒绝使用一个或多个 CCS 操作。同时还必须指定可以用于操作的资源（可以是全部资源，某些操作也可以是部分资源），策略还可以包含操作资源所设置的条件。

CCS 部分 API 操作支持资源级权限，意味着，对于该类 API 操作，您不能在使用该类操作的时候指定某个具体的资源来使用，而必须要指定全部资源来使用。

支持的资源级权限

最近更新时间：2018-09-20 14:19:58

资源级权限指的是能够指定允许用户对哪些资源具有执行操作的能力。TKE(原CCS) 支持部分资源级权限，这意味着对于某些 TKE 操作，您可以控制何时允许用户执行操作 (基于必须满足的条件)或是允许用户使用的特定资源。
TKE 中可授权的资源类型：

资源类型	授权策略中的资源描述方法
集群相关	qcs::ccs:$region::cluster/*

下表将介绍当前支持资源级权限的 TKE（ TKE，Tencnet Kubernetes Engines，容器服务） API 操作。指定资源路径的时候，您可以在路径中使用 * 通配符。

注意：
如果某一个 TKE API 操作在下表中没有列出，则它不支持资源级权限。如果TKE API 操作不支持资源级权限，那么您还是可以向用户授予使用该操作的权限，但是必须为策略语句的资源元素指定 * 。

API 操作	资源路径
DescribeClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DescribeClusterServiceInfo	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
CreateClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId 负载均衡资源 qcs::clb:$region:$account:clb/* 云硬盘资源 qcs::cvm:$region:$account:volume/* qcs::cvm:$region:$account:volume/$diskId

API 操作	资源路径
ModifyClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId 负载均衡资源 qcs::clb:$region:$account:clb/* 云硬盘资源 qcs::cvm:$region:$account:volume/* qcs::cvm:$region:$account:volume/$diskId
DeleteClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
ModifyServiceDescription	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DescribeServiceEvent	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
ResumeClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
PauseClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
RollBackClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
ModifyClusterServiceImage	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
RedeployClusterService	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DescribeServiceInstance	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId

API 操作	资源路径
ModifyServiceReplicas	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DeleteInstances	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DescribeClusterNameSpaces	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
CreateClusterNamespace	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DeleteClusterNamespace	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DescribeCluster	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
CreateCluster	云服务器资源 qcs::cvm:$region:$account:instance/*
DeleteCluster	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
DescribeClusterInstances	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId
AddClusterInstances	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId 云服务器资源 qcs::cvm:$region:$account:instance/*

API 操作	资源路径
DeleteClusterInstances	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId 云服务器资源 qcs::cvm:$region:$account:instance/* qcs::cvm:$region:$account:instance/$instanceId
AddClusterInstancesFromExistedCvm	集群资源 qcs::ccs:region:account:cluster/* qcs::ccs:region:account:cluster/$clusterId 云服务器资源 qcs::cvm:$region:$account:instance/* qcs::cvm:$region:$account:instance/$instanceId

配置 registry 镜像权限

最近更新时间：2018-05-30 15:14:38

容器镜像服务权限介绍

腾讯云容器镜像的描述格式是: `ccr.ccs.tencentyun.com/${namespace}/${name}:${tag}` 。
镜像仓库的权限围绕以下两个字段进行设置：

- `${namespace}` : 镜像所属命名空间；
- `${name}` : 镜像名字；

> **注意：**
> 命名空间 `${namespace}` 及镜像名字 `${name}` 中不能包含斜杠“ / ”；
> `${tag}` 字段目前只实现了删除操作鉴权，请参考 镜像Tag权限；

通过 `${namespace}` ， `${name}` 两个字段，开发商可以为协作者制定详细的权限方案，实现灵活的权限管理。
例如：

- 允许协作者A拉取镜像
- 禁止协作者A删除镜像
- 禁止协作者B拉取命名空间ns1中的镜像
- ...

如果您不需要详细管理镜像仓库权限，可以使用预设策略授权。

如果您需要细致地管理协作者权限，请使用自定义策略授权。

容器镜像服务权限基于腾讯云CAM进行管理，您可以详细了解CAM的使用方法：
用户管理，
策略管理，
授权管理

预设策略授权

为了简化容器镜像服务权限管理，容器镜像服务内置了两个预设策略:

- 镜像仓库（CCR）全读写访问权限

  该预设策略配置了容器镜像服务所有权限，如果协作者关联该预设策略后，将与开发商拥有相同的镜像仓库权限。 详情请查看权限列表。

- 镜像仓库（CCR）只读访问权限

  该预设策略包含了容器镜像服务只读操作的权限，如果协作者在容器镜像服务中 **只** 关联了该预设策略，则以下操作将被禁止：

  - `docker push` 推送镜像
  - 新建镜像仓库命名空间
  - 删除镜像仓库命名空间
  - 创建镜像仓库
  - 删除镜像仓库
  - 删除镜像Tag

如果您不了解如何为协作者关联预设策略，请参考CAM文档： 预设策略介绍、预设策略关联用户

自定义策略授权

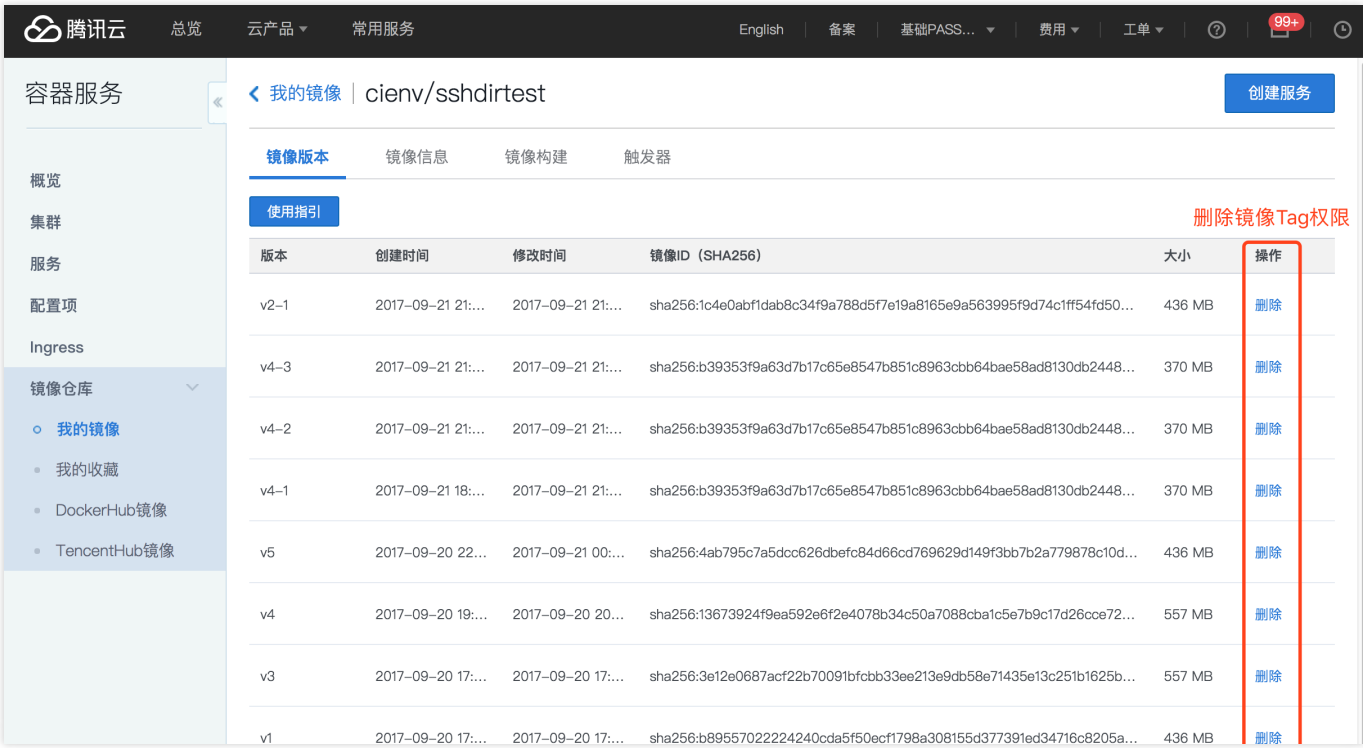
通过自定义策略，开发商可以为不同的协作者关联不同的权限。

当您分配权限时，考虑这些要素：

- **资源(resource)**： 该权限策略关联哪些镜像，例如所有镜像仓库描述为 `qcs::ccr:::repo/*` ，详见CAM资源描述方式；
- **动作(action)**： 该权限策略对 **资源(resource)** 有哪些操作，如删除、新建等，通常使用接口进行描述；
- **效力(effect)** ： 该权限策略对协作者表现出的效果(允许/拒绝)；

一旦您规划好权限设置，就可以开始进行权限分配。下面我们以“允许协作者创建镜像仓库”为例进行说明：

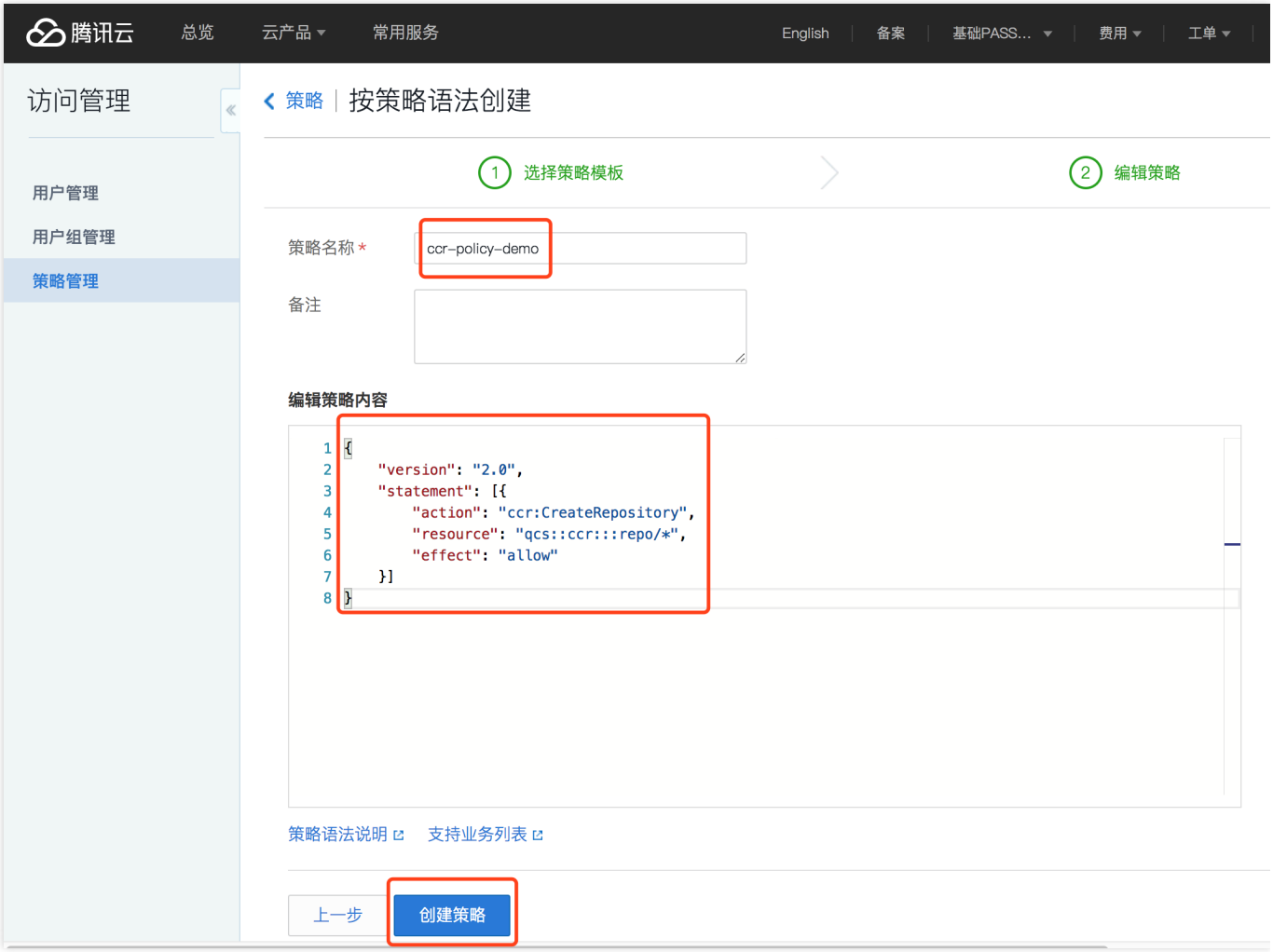
1. 创建自定义策略（CAM文档），
   - 使用开发商账号登录腾云讯-控制台
   - 进入CAM自定义策略管理页面，单击“新建自定义策略”按钮打开“选择策略创建方式”对话框



- 选择“按策略语法创建”选项 >> 选择“空白模板”

- 单击页面下方“下一步”按钮，进入“按策略语法创建” - “编辑策略” 页面
- 在“编辑策略内容”编辑框中填入以下内容，在“策略名称”中填入一个有意义的名字，如 ccr-policy-demo

```
{
"version": "2.0",
"statement": [{
"action": "ccr:CreateRepository",
"resource": "qcs::ccr:::repo/*",
"effect": "allow"
}]
}
```



*注: resource **未尾** 使用 * 表示可以在任意命名空间下创建镜像仓库*

- 单击页面底部“创建策略”按钮，结束策略创建过程。

2. 关联自定义策略。步骤1中的策略(ccr-policy-demo)创建完成以后，您可以将其关联到任意协作者，详见CAM文档。策略关联完成后协作者即拥有 **在任意命名空间下创建镜像仓库权限。**

resource `qcs::ccr:::repo/*` *格式说明:*

- `qcs::ccr:::` 为固定格式，表示开发商的腾讯云容器镜像仓库服务；
- `repo` 为固定前缀，代表资源类型，这里是镜像仓库；
- 斜杠(`/`)后面的 `*` 表示匹配所有镜像仓库。

关于resource更详细的描述，请参考CAM资源描述方式

按资源进行授权

您可以同时为多个资源进行授权。例如：“允许删除命名空间foo, bar中的镜像仓库”，可以创建下面的自定义策略：

```
{
"version": "2.0",
"statement": [{
"action": [
"ccr:BatchDeleteRepository",
"ccr:DeleteRepository"
],
"resource": [
"qcs::ccr:::repo/foo/*",
"qcs::ccr:::repo/bar/*"
]
"effect": "allow"
}]
}
```

注:

- qcs::ccr:::repo/foo/* 中 foo/* 表示镜像仓库命名空间 foo 下的所有镜像
- qcs::ccr:::repo/bar/* 中 bar/* 表示镜像仓库命名空间 bar 下的所有镜像

按动作(接口)进行授权

您可以对一个资源配置多个 action ，实现资源权限的统一管理。例如："允许创建、删除、push命名空间foo中的镜像仓库"，可以创建下面的自定义策略：

```
{
"version": "2.0",
"statement": [{
"action": [
"ccr:CreateRepository",
"ccr:BatchDeleteRepository",
"ccr:DeleteRepository",
"ccr:push"
],
"resource": "qcs::ccr:::repo/foo/*",
"effect": "allow"
}]
}
```

权限列表

docker client 权限

resource： qcs::ccr:::repo/${namespace}/${name}
action：

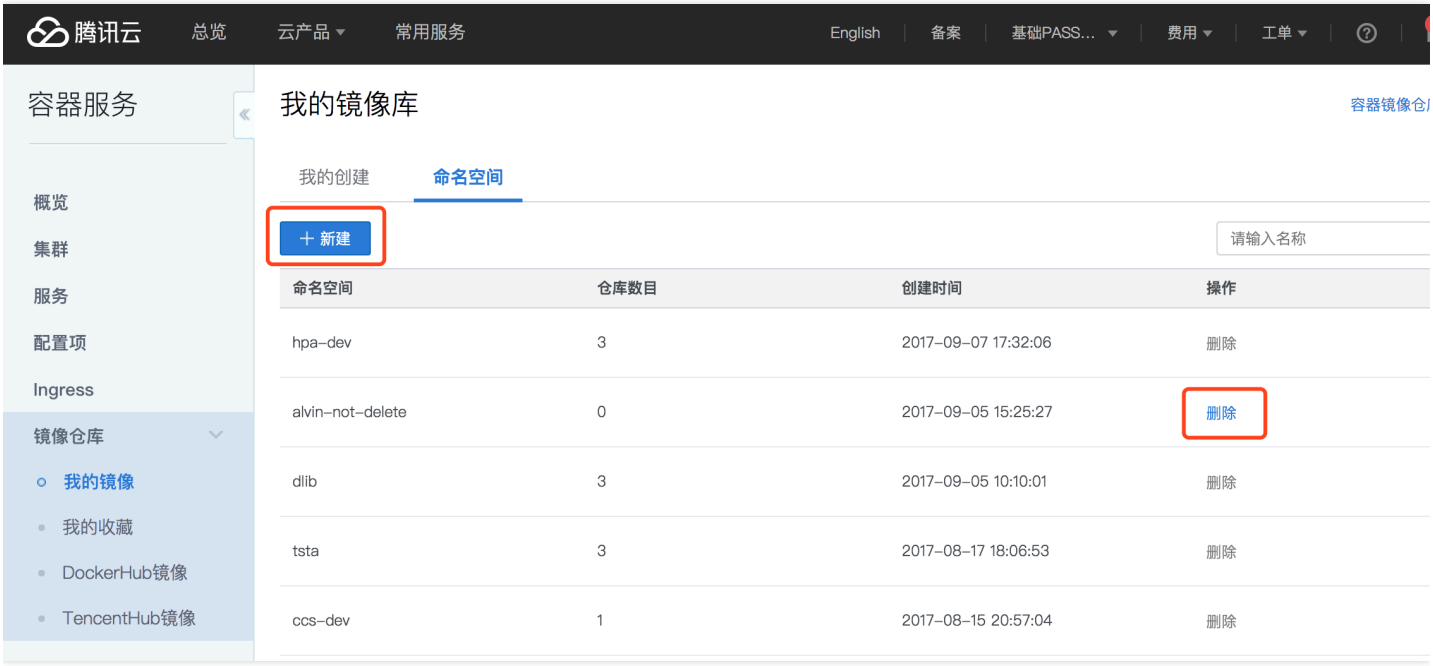
- ccr:pull 使用docker命令行pull镜像
- ccr:push 使用docker命令行push镜像

命名空间权限

resource： qcs::ccr:::repo/${namespace}
action:

- ccr:CreateCCRNamespace 新建镜像仓库命名空间
- ccr:DeleteUserNamespace 删除镜像仓库命名空间

功能指引: **容器服务** >> 左侧导航栏 **镜像仓库** >> **我的镜像** >> **命名空间**

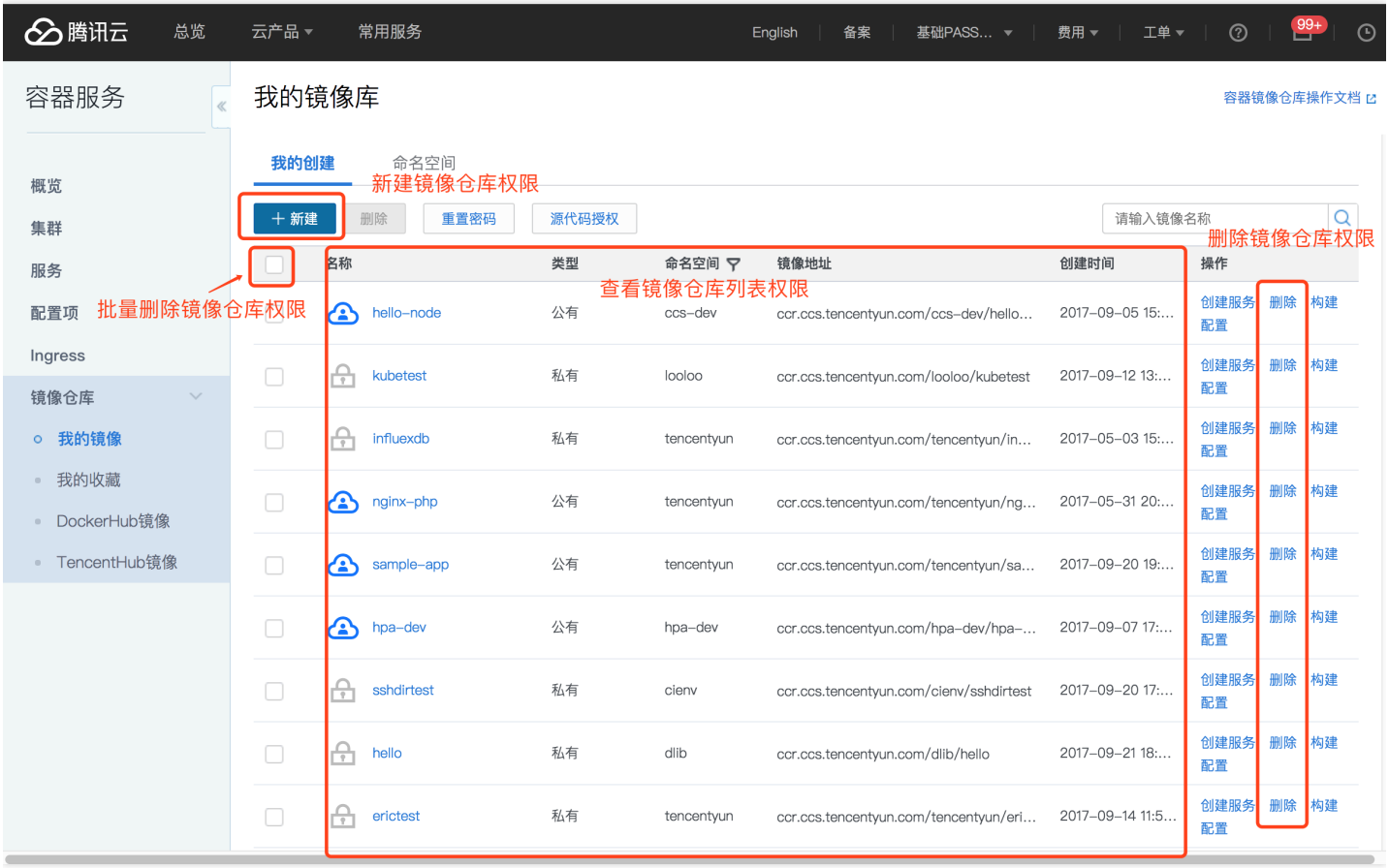


镜像仓库权限

resource： `qcs::ccr:::repo/${namespace}/${name}`
action：

- `ccr:CreateRepository` 创建镜像仓库
- `ccr:DeleteRepository` 删除镜像仓库
- `ccr:BatchDeleteRepository` 批量删除镜像仓库
- `ccr:GetUserRepositoryList` 查看镜像仓库列表

功能指引: **容器服务** >> 左侧导航栏 **镜像仓库** >> **我的镜像** >> **我的创建**



注意：

要阻止协作者删除某些镜像，请配置多个 action 来实现。

例如：禁止删除任何镜像仓库：

```
{
"version": "2.0",
"statement": [{
"action": [
"ccr:BatchDeleteRepository",
"ccr:DeleteRepository"
],
"resource": "qcs::ccr:::repo/*",
"effect": "deny"
}]
}
```

镜像Tag权限

resource： qcs::ccr:::repo/${namespace}/${name}:${tag}

action:

- ccr:DeleteTag 删除镜像Tag权限

功能指引: **容器服务** >> 左侧导航栏 **镜像仓库** >> **我的镜像** >> **我的创建** >> 单击某个镜像名称 >> **镜像版本** 页面