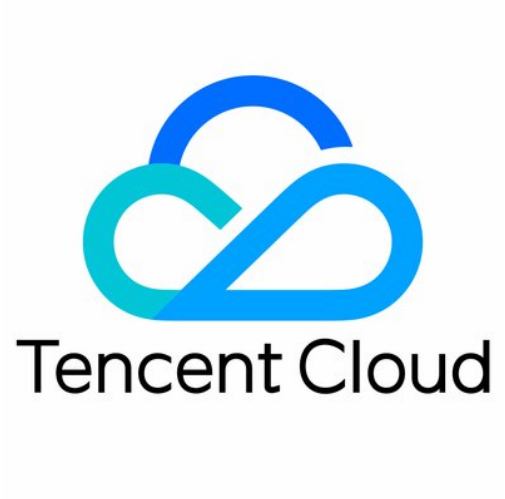
Tencent Container Registry

Access Management

Product Documentation



Contents

Access Management Overview

Last updated : 2020-07-28 15:55:33

Introduction to Cloud Access Management

Cloud access management (CAM) is a web service provided by Tencent Cloud. It helps users securely manage the permissions for accessing resources under their Tencent Cloud accounts. CAM allows you to create, manage, or terminate users (groups) and controls who can use Tencent Cloud resources through identity management and policy management.

When you use CAM, you can associate a policy with a user or a user group. The policy authorizes or refuses users to use the specified resource to complete the specified task. For more information on CAM policies, refer to Policy Syntax. For more information on how to use CAM policies, refer to Policies.

If you do not need to perform access management of TCR resources for sub-accounts, you can skip this section. This does not affect your understanding and use of other sections of this document.

CAM-based Resource-level Access Control of TCR

Resource-level permissions refer to the capabilities that can specify and allow users to perform specific operations on specific resources. TCR supports resource-level access control of CAM and controls the granularity to the repository level, that is, you can authorize sub-accounts to perform operations on resources in only the specified image repository or the Helm Chart repository by configuring the CAM policy.

Types of resources that can be authorized by TCR in CAM:

Resource Type	Resource Description Method in Authorization Policy
Enterprise edition instance	`qcs::tcr:$region:$account:instance/*`
Enterprise edition repository	`qcs::tcr:$region:$account:repository/*`
Personal edition repository	`qcs::tcr:$region:$account:repo/*`

- `$region` : the region information. For example, `ap-guangzhou` indicates the region of Guangzhou. If the value is null, the field indicates all regions. For the specific list of regions and abbreviations, refer to Regions and Availability Zones.
- `$account` : the root account of the resource owner. The value is expressed as `uin/${uin}` , for example, `uin/12345678` . If the value is null, the field indicates the root account of the CAM user who creates the policy.
  For details on resource description in the authorization policy, refer to Resource Description.

Container Image Service Enterprise Edition
CAM APIs for Enterprise Edition

Last updated : 2021-04-08 10:41:06

Instance Management APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateInstance Creating an instance	instance	`qcs::tcr:$region:$account:instance/$instanceid`
DescribeInstanceStatus Querying the instance status	instance	`qcs::tcr:$region:$account:instance/*` `qcs::tcr:$region:$account:instance/$instanceid`
DescribeInstances Querying the instance information	instance	`qcs::tcr:$region:$account:instance/*` `qcs::tcr:$region:$account:instance/$instanceid`
CreateInstanceToken Creating an instance access credential	instance	`qcs::tcr:$region:$account:instance/$instanceid`
DeleteInstanceToken Deleting a long-term access credential	instance	`qcs::tcr:$region:$account:instance/$instanceid`
ModifyInstanceToken Updating the instance’s long-term access credential	instance	`qcs::tcr:$region:$account:instance/$instanceid`
DescribeInstanceToken Querying the long-term access credential information	instance	`qcs::tcr:$region:$account:instance/$instanceid`

Namespace APIs

APIs and Description	Resource	Six-segment Example of Resource

	Type	
CreateNamespace Creating a namespace	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceI
DeleteNamespace Deleting a namespace	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceI
ModifyNamespace Updating the namespace information	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceI
DescribeNamespaces Querying the namespace information	repository	qcs::tcr:$region:$account:repository/$instanceId/* qcs::tcr:$region:$account:repository/$instanceId/$namespaceI

Image Repository APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateRepository Creating an image repository	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceName/
DeleteRepository Deleting an image repository	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceName/
ModifyRepository Updating the image repository information	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceName/
DescribeImages Querying the container image information	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceName/
DescribeImages Querying the image repository information	repository	qcs::tcr:$region:$account:repository/$instanceId/$namespaceName/ qcs::tcr:$region:$account:repository/$instanceId/$namespaceName/

Example of Authorization Solution of the Enterprise Edition

Last updated : 2021-11-24 17:34:28

This document describes how to enable sub-accounts to view and use the TCR related resources through the CAM policy, including specific operation steps and common policy configuration examples.

> Note :
> If you need the permissions of other Tencent Cloud services when using some features in TCR console such as VPC, CloudAudit, Tag, please see the corresponding CAM Guide in [CAM-Enabled Products](https://intl.cloud.tencent.com/ Document/product/598/10588).

Directions

This document takes the example of "granting the sub-account the read-only permission of an image repository" to introduce how to create a policy.

- **Instance ID**: tcr-xxxxxxxx
- **Namespace**: team-01
- **Image repository**: repo-demo

- Creating
- Creating

1. Log in to the CAM console.
2. Click **Policies** on the left sidebar to access the **Policies** page.
3. Click **Create Custom Policy** in the upper-left corner.
4. In the selection window that pops up, click **Create by Policy Generator** to go to the **Edit Policy** page.
5. Select the service in the Visual Policy Generator, enter the following information, and edit an authorization statement.
   - **Effect**: select **Allow** or **Deny**. Here we select **Allow**.
   - **Service**: select the service you want to authorize. Here we select **Tencent Container Registry (tcr)**.

- **Action**: select the operations you want to authorize. Here we select **Read**.
- **Resource**: select all resources or specific resources you want to authorize. Here we select **Specific resources**, and add the following six-segment resource to restrict the access.
  - **repository**: select the region where the repository resides, and enter the resource path of the repository, for example, `tcr-xxxxxxxx/team-01/repo-demo/*`. You can get the resource path in Image Repository.
  - **repo**: it is left empty.
  - **instance**: select the region where the repository resides, and enter the ID of the instance to which the repository belongs, for example, `tcr-xxxxxxxx`. You can get the instance ID in the Instance List.
- **Condition**: it is left empty.

6. Click **Next** to go to the **Associate Users/User Groups** page.
7. In the **Associate Users/User Groups** page, add the policy name and description, and you can associate users or user groups for quick authorization at the same time.
8. Click **Done** to complete the custom policy creation.

Common Policy Configuration

If you need to customize the policy JSON, please see CAM APIs for Enterprise Edition and Syntax Logic.

Preset policy configuration

- **QcloudTCRFullAccess**: full read/write permission of TCR.
  After the policy is bound to a sub-account, the sub-account has all operation permissions for all TCR resources, including the Enterprise Edition and the Personal Edition in TKE.

```
{
”version”: ”2.0”,
”statement”: [{
”action”: [
”tcr:*”
],
”resource”: ”*”,
”effect”: ”allow”
}]
}
```

- **QcloudTCRReadOnlyAccess**: read-only permission of TCR.
  After the policy is bound to a sub-account, the sub-account has the read-only permission for all TCR resources, including the Enterprise Edition and the Personal Edition in TKE.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:Describe*",
"tcr:PullRepository*"
],
"resource": "*",
"effect": "allow"
}]
}
```

Policy configuration in typical scenarios

Note :
The policies in the following use cases are used only for the Enterprise Edition. For the policies used for Personal Edition, please see Example of Authorization Solution of the Personal Edition.

- Grant a sub-account all read/write operation permissions for all resources in the TCR Enterprise Edition instance.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:*"
],
"resource": [
"qcs::tcr:::instance/*",
"qcs::tcr:::repository/*"
],
"effect": "allow"
}]
}
```

- Grant a sub-account the read-only operation permission for all resources in the TCR Enterprise Edition instance.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:Describe*",
"tcr:PullRepository*"
],
"resource": [
"qcs::tcr:::instance/*",
"qcs::tcr:::repository/*"
],
"effect": "allow"
}]
}
```

- Authorize a sub-account to manage the specified instance, for example, dev-guangzhou whose instance ID is tcr-xxxxxxxx.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:*"
],
"resource": [
"qcs::tcr:::instance/tcr-xxxxxxxx"
],
"effect": "allow"
}]
}
```

- Authorize a sub-account to manage the specified namespace in the specified instance, for example, team-01 under the instance tcr-xxxxxxxx.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:*"
],
```

```
"resource": [
"qcs::tcr:::repository/tcr-xxxxxxxx/team-01/*"
],
"effect": "allow"
},
{
"action": [
"tcr:DescribeInstance*"
],
"resource": [
"qcs::tcr:::repository/tcr-xxxxxxxx"
],
"effect": "allow"
}
]
}
```

- Authorize a sub-account the read-only permission of an image repository, which means that the sub-account can only pull the images in the image repository instead of deleting a repository, modifying repository attributes, or pushing images, for example, repo-demo in the namespace team-01 under the instance tcr-xxxxxxxx.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:DescribeRepositories",
"tcr:PullRepository",
"tcr:DescribeNamespaces"
],
"resource": [
"qcs::tcr:::repository/tcr-xxxxxxxx/team-01/repo-demo/*"
],
"effect": "allow"
},
{
"action": [
"tcr:DescribeInstance*"

],
"resource": [
"qcs::tcr:::instance/tcr-xxxxxxxx"
],
"effect": "allow"
}
```

```
]
}
```

Container Image Service personal Edition
CAM APIs for Personal Edition

Last updated : 2021-04-08 10:41:06

Namespace APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateNamespacePersonal Creating a namespace of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace`
DeleteNamespacePersonal Deleting a namespace of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace`

Image Repository APIs

APIs and Description	Resource Type	Six-segment Example of Resource
DescribeRepositoryOwnerPersonal Querying all repositories of Personal Edition	repo	`qcs::tcr:$region:$account:repo/*`
CreateRepositoryPersonal Creating an image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$repo`
DeleteRepositoryPersonal Deleting an image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$repo`
BatchDeleteRepositoryPersonal Deleting the image repositories of Personal Edition in batches	repo	`qcs::tcr:$region:$account:repo/$namespace/*`
DeleteImagePersonal Deleting the repository tag of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$repo`

APIs and Description	Resource Type	Six-segment Example of Resource
BatchDeleteImagePersonal Deleting the repository tags of Personal Edition in batches	repo	`qcs::tcr:$region:$account:repo/$namespace/$repo`
PullRepositoryPersonal Pulling the images in the image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$repo`
PushRepositoryPersonal Pushing the images in the image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$repo`

Example of Authorization Solution of the Personal Edition

Last updated : 2020-07-28 15:55:33

Configuring Policies in Typical Scenarios

⚠ Note :

The policies in the following scenarios are only used for Personal Edition.

- Grant a sub-account all read/write operation permissions for all resources in the TCR Personal Edition instance (image repository in TKE of the original TCR).

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

- Grant a sub-account the read-only operation permission for all resources in the TCR Personal Edition instance (image repository in TKE of the original TCR).

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:Describe*",
"tcr:PullRepository*"
],
"resource": [
"qcs::tcr:::repo/*"
],
"effect": "allow"
}]
}
```

- Authorize a sub-account to manage the specified instance in the specified region, for example, namespace team-01 in the default region.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:*"
],
"resource": [
"qcs::tcr:ap-guangzhou:*:repo/team-01/*"
],
"effect": "allow"
}
]
}
```

- Authorize a sub-account to read only an image repository and pull only images in the image repository instead of deleting a repository, modifying repository attributes, or pushing images, for example, the image repository repo-demo in the namespace team-01 under the default region.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:Describe*",
"tcr:PullRepositoryPersonal"
],
"resource": [
"qcs::tcr:ap-guangzhou:*:repo/team-01/repo-demo/*"
],
"effect": "allow"
},
{
"action": [
"tcr:Describe*"
],
"resource": [
"qcs::tcr:ap-guangzhou:*:repo/team-01/*"
],
"effect": "allow"
}
]
}
```

Update Guide of Resource Level APIs and Authorization Solution of Personal Edition

Last updated : 2021-04-08 10:41:06

Overview

TCR provides container image hosting and distribution services to enterprise users and personal users. The Personal Edition provides users with simple and free basic services, that is, the image repository in TKE.
To provide users with more standardized interface definitions and significantly reduced access delay API services, the APIs of the original Personal Edition image repository (CCR) has been upgraded from version 2.0 to the latest version 3.0, and the API name and authorization solutions have updated accordingly. This document describes the mappings between the new and legacy APIs after the upgrade of the APIs that support resource-level authentication and how to use the new authorization solution.

Mappings Between the v2.0 and v3.0 APIs

API Name of v2.0	API Name of v3.0	Description	Latest Resource De
CreateCCRNamespace	CreateNamespacePersonal	Creates a namespace of Personal Edition	`qcs::tcr:$region:$`
DeleteUserNamespace	DeleteNamespacePersonal	Deletes a namespace of Personal Edition	`qcs::tcr:$region:$`
GetUserRepositoryList	DescribeRepositoryOwnerPersonal	Queries all repositories of Personal Edition	`qcs::tcr:$region:$`
CreateRepository	CreateRepositoryPersonal	Creates an	`qcs::tcr:$region:$`

		image repository of Personal Edition	
DeleteRepository	DeleteRepositoryPersonal	Deletes an image repository of Personal Edition	`qcs::tcr:$region:$`
BatchDeleteRepository	BatchDeleteRepositoryPersonal	Deletes image repositories of Personal Edition in batches	`qcs::tcr:$region:$`
DeleteTag	DeleteImagePersonal	Deletes the repository tag of Personal Edition	`qcs::tcr:$region:$`
BatchDeleteTag	BatchDeleteImagePersonal	Deletes the repository tags of Personal Edition in batches	`qcs::tcr:$region:$`
pull	PullRepositoryPersonal	Pulls the images in the image repository of Personal Edition	`qcs::tcr:$region:$`
push	PushRepositoryPersonal	Pushes the images in the image repository of Personal Edition	`qcs::tcr:$region:$`

Mappings Between the legacy and new Authorization Solutions

Due to the upgrade and update of the product name and API version, the original resource description methods and actions of the TCR Personal Edition have been updated accordingly. Please use the latest resources and action authorization solutions while using the v3.0 APIs.
During the upgrade of APIs, CAM APIs will be compatible with both the legacy and new resource description methods and actions to ensure that the custom policies are still effective. To make it easier for you to manage the APIs and authorization solutions uniformly, we recommend that you upgrade the authorization solution to the latest version. For more information, please see Example of Authorization Solution of the Personal Edition.

The legacy resource-level authorization solution

- **Action**: use `ccr` as the product prefix, and the API name is version 2.0. For example, create a namespace as `ccr:CreateCCRNamespace` .
- **Resource description**: use `ccr` as the product name, and there is only a `repo` resource type. For example, to describe the image repository `repo-b` under the namespace `namespace-a` , it would be `qcs::ccr:::repo/namespace-a/repo-b` . If `$region` and `$account` are left empty, all regions will be used by default, and the account will be the root account of the CAM user who created the policy by default.
  For more information on authorization solution, see TKE Image Registry Resource-level Permission Settings.

The new resource-level authorization solution

- **Action**: use `tcr` as the product prefix, and the API name is version 3.0. For example, create a namespace of Personal Edition as `tcr:CreateNamespacePersonal` .
- **Resource description**: use `tcr` as the product name, and there are three resource types: `instance` , `repository` and `repo` . Among them, `repo` is a dedicated resource type of the Personal Edition. For example, to describe the image repository `repo-b` under the namespace of Personal Edition `namespace-a` , it would be `qcs::tcr:$region:$account:repo/namespace-a/repo-b` . If `$region` and `$account` are left empty, all regions will be used by default, and the account will be the root account of the CAM user who created the policy by default.
  For more information on authorization solution, see CAM APIs for Personal Edition and Example of Authorization Solution of the Personal Edition.

The compatible example of legacy and new authorization solutions

For example, the authorized sub-account can read the image repository of `repo-b` (an image repository of Personal Edition) under the namespace `namespace-a` in the default region. Then this account can only query the repository information and pull the image in the repository, but cannot modify the repository attributes, push images, and delete the repository.

- **Legacy authorization solution:**

```
{
"version": "2.0",
"statement": [{
"action": [
"ccr:pull"
],
"resource": "qcs::ccr:::repo/namespace-a/repo-b",
"effect": "allow"
}]
}
```

- **New authorization solution:**

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:PullRepositoryPersonal"
],
"resource": "qcs::tcr:::repo/namespace-a/repo-b",
"effect": "allow"
}]
}
```