Tencent Push Notification Service

Users and Permissions

Product Documentation

Contents

Users and Permissions
- Quick Configuration
- Advanced custom configuration

Users and Permissions Quick Configuration

Last updated : 2020-12-14 11:53:22

This document describes how to create and authorize sub-users. If you have never used Tencent Cloud Access Management (CAM), please read this document for more information on the configuration.

TPNS uses CAM for permission management. You need to authorize applications, create sub-users, and grant application permissions to the sub-users. For detailed directions, please see the following sections.

Creating a Sub-User

1. Log in to the CAM console and click **Create User**.

2. The following describes the custom creation method. Click **Custom Create** to enter the **Create Sub-user** page.

3. Configure the login information of the sub-user as instructed and grant the sub-user application permissions on the **Setting User Permission** page.

Granting Application Permissions

Granting permissions of all applications in a centralized manner

1. Continue from the last step in the previous section as shown below:

2. Enter "TPNS" in the search box. In the search results, there are two default preset permissions as listed below:

Policy Name	Permission Scope
QcloudTPNSFullAccess	Grants all permissions of all applications under the root account
QcloudTPNSReadOnlyAccess	Grants read and push permissions of all applications under the root account

Granting permissions of selected applications

1. Click **Create Custom Policy**.

2. On the displayed page, select **Create by Policy Syntax** as shown below.

3. Select **Blank Template**.

4. Click **Next** to enter the syntax creation page as shown below.

Copy the following syntax code:

```
{
"version": "2.0",
"statement": [
{
"action": [
"tpns:Describe*",
"tpns:CancelPush",
"tpns:DownloadPushPackage",
"tpns:CreatePush",
"tpns:UploadPushPackage"
],
"resource": [
"qcs::tpns::uin/1000000000:app/1500000000"
],
"effect": "allow"
},
{
"action": [
"tpns:Describe*"
],
"resource": [
"qcs::tpns::uin/1000000000:/*"
],
"effect": "allow"
}

]
}
```

Replace parameters in the syntax code as follows:

- Replace the ID of the root account: enter the Account Info page under the current root account, copy the account ID, and replace `1000000000` in the syntax above with it.

> **Note :**
>
> If your current login account is a collaborator or sub-account, you need to get the account ID from the owner of the root account that grants you permissions.

- Replace the application `Access_ID` : log in to the TPNS console, copy the `Access_ID` of the application whose permissions you want to grant, and replace `1500000000` in the syntax above with it. If you want to grant permissions of multiple applications, you can change `resource` to: `"qcs::tpns::uin/1000000000:app/{application Access_ID1}"` , `"qcs::tpns::uin/1000000000:app/{application Access_ID2}"`

> **Note :**
>
> Please delete "{" and "}" in actual use. For detailed directions, please see Advanced Custom Configuration.

5. Return to the user creation page.

Search for the created policy by name, select it, click **Next**, and click **Complete**.
6. After the permission configuration, you can select **Sub-User Login** on the login page to verify the account permissions.

Advanced custom configuration

Last updated : 2020-09-01 18:07:36

Cloud Access Management Overview

If you use the TPNS service in Tencent Cloud, and the service is managed by different users sharing your Tencent Cloud account key, you may face the following problems:

- Your key is shared by multiple users, leading to high risk of compromise.
- You cannot limit the access permissions of other users, which poses a security risk due to potential faulty operations.

You can allow different users to manage different services through sub-accounts so as to avoid the above problems. By default, a sub-account does not have permission to use a TPNS service or related resources. Therefore, you need to create a policy to grant the required permission to the sub-account.
Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access permissions to your Tencent Cloud resources. Using CAM, you can create, manage, and terminate users (groups), and control the Tencent Cloud resources that can be used by the specified user through identity and policy management.
When using CAM, you can associate a policy with a user or user group to allow or forbid them to use specified resources to complete specified tasks. For more information on CAM policies, please see Syntax Logic. For more information on how to use CAM policies, please see Policy.
If you do not need to manage the access permissions to TPNS resources for sub-accounts, you can skip this part. This will not affect your understanding and usage of other parts in the documentation.

Policy Syntax Description

A CAM policy must authorize or deny the use of one or more TPNS operations. At the same time, it must specify the resources that can be used for the operations (which can be all resources or partial resources for certain operations). For TPNS operations that do not support resource-level authorization, you need to specify the authorized object as all resources.
CAM policy syntax description:

```
{
"version":"2.0",
"statement":
```

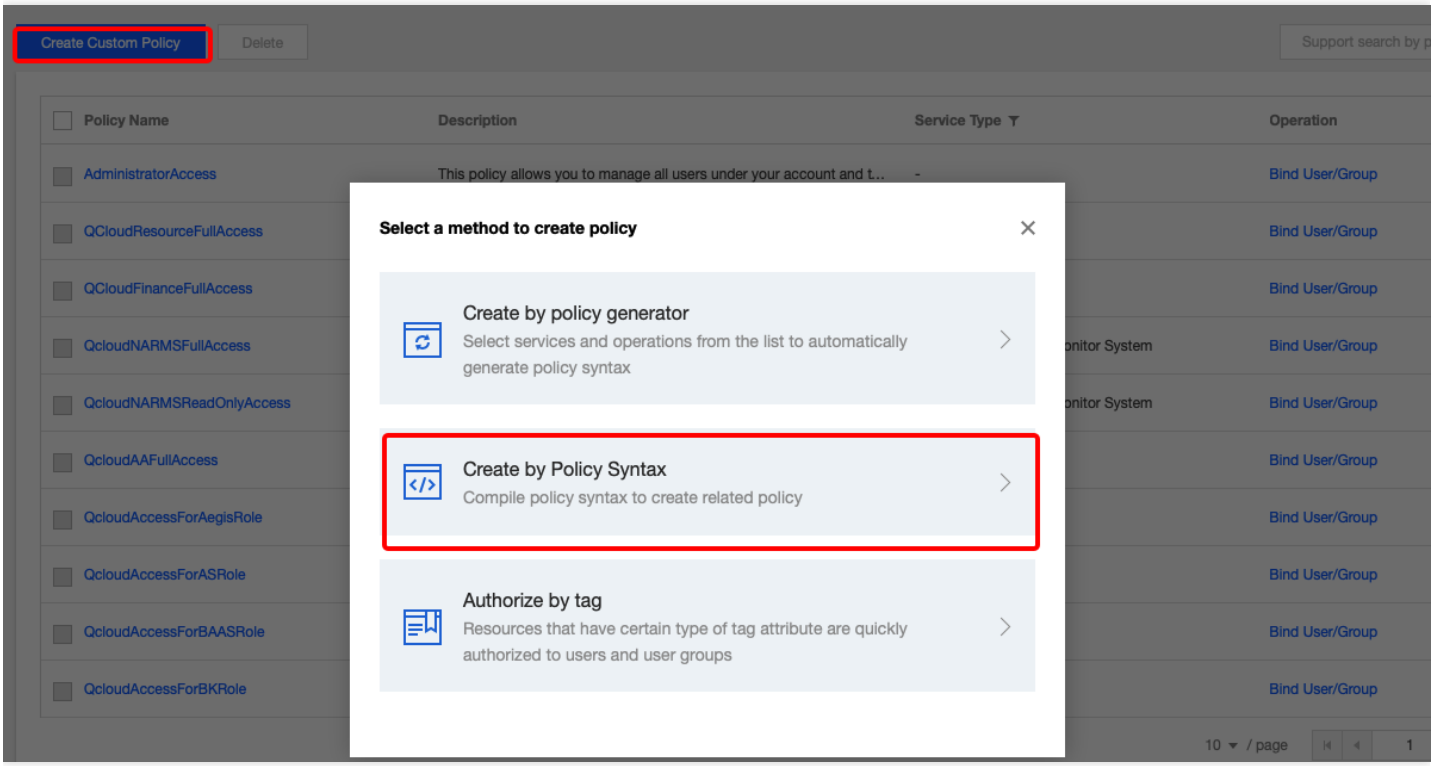
```
[
{
"effect":"effect",
"action":["action"],
"resource":["resource"],
"condition": {"key":{"value"}}
}
]
}
```

Parameter description:

Parameter Name	Required	Description
version	Yes	It is the version number. Currently, only "2.0" is allowed.
statement	Yes	This element describes the details of one or more permissions. It contains a permission or permission set of other elements such as `effect`, `action`, `resource`, and `condition`. One policy has only one statement. An `action` (operation) describes an allowed or denied operation, which can be an API or a feature set (a set of specific APIs prefixed with `permid`).
resource	Yes	This element describes the details of authorization. A resource is described in a six-segment format. Detailed resource definitions vary by product. For more information on how to specify a resource, please see the documentation for the product whose resources you are writing a statement for.
condition	No	This element describes the condition for the policy to take effect. A condition consists of operator, action key, and action value. A condition value may contain information such as time and IP address. Some services allow you to specify additional values in a condition.
effect	Yes	This element describes whether the statement result is an "allow" or "explicit deny".

Creating Policy and Granting Permission

Two types of system-level policies are preset for you to quickly and easily grant permissions. You can go to the console > Access Management > Policy Management, click **Create Custom Policy**, and select "Create by Policy Syntax" as shown below:



On the policy editing page, you can search and find two preset policy templates provided by TPNS, which grants the full access and read-only access, respectively (you can view the list of specific permissions during policy creation). You can select a template and edit it or create a blank template.

After creating a policy, you can find it on the Policy Management page in the CAM Console and associate it with sub-user to complete permission configuration.
This document describes how to grant TPNS permissions in CAM.

Authorizable TPNS Resources

Resource-level permission can be used to specify which resources a user can manipulate. The type of resources that can be authorized in TPNS is "application", that is, you can grant resource-level permissions in CAM at the application granularity. The resource description method is as follows:

```
qcs::tpns::uin/1000000000:app/*
```

Here, `*` indicates all resources at the application granularity, which can be replaced with the `Access ID`. You can find the application's `Access ID` in the Product Management module in the TPNS Console. For the `uin`, get the account ID on the Account Info page in the console and replace the `uin` with it (such as `1000000000`, which is a sample Tencent Cloud ID of a root account).

When authorizing multiple resources, separate them with commas.

Authorizable TPNS Operations

In a CAM policy statement, you can specify any API operation from any service that supports CAM. APIs prefixed with `name/tpns:` should be used for TPNS, such as `name/tpns:CreateProduct`.
To specify multiple operations in a single statement, separate them with commas as shown below:

```
"action":["tpns:action1","tpns:action2"]
```

You can also specify multiple operations by using a wildcard. For example, you can specify all operations beginning with "Describe" in name, as shown below:

```
"action":["tpns:Describe*"]
```

If you want to specify all operations in TPNS, use a wildcard "*" as shown below:

```
"action":["tpns:*"]
```

List of operations that can be authorized:

⚠ Note :

Only operations that support resource-level permissions can be authorized at the application level.

Name	Description	Resource-Level Permission Supported
AddChannelInfo	Adds vendor channel	Yes
CancelPush	Cancels scheduled push task	Yes
CreateApp	Creates application	No

CreateAppTrialRequest	Applies for product trial	Yes
CreateProduct	Creates product	No
DeleteAppInfo	Deletes application	Yes
DeleteProductInfo	Deletes product	No
DescribeApnsCertInfo	Queries APNS certificate information	Yes
DescribeAppAllTags	Queries all tag information	Yes
DescribeAppInfo	Queries application information	Yes
DescribeAppVipInfo	Queries VIP information	Yes
DescribeChannelInfo	Queries vendor channel information	Yes
DescribeProductInfo	Queries product information	No
DescribeTagTokenNums	Queries the quantity of devices under tag	Yes
DownloadPushPackage	Downloads push number package	Yes
DescribeAccountByToken	Queries account bound to device	Yes
DescribeAccountPushStatInfo	Queries the total number of push messages under account	No
DescribeAccountPushStatInfoAllZone	Queries the total number of messages supposed to be sent by all applications in cluster	No
DescribeAppSecretInfo	Queries `AppSecret` information	Yes
DescribeDeviceStatOverview	Queries the number of accumulated and daily active devices of application	Yes
DescribeProductDeviceStatWithRatioOverview	Queries application statistics	Yes
DescribePushPackaDescribeoken	Uploads number package to get temporary COS token	Yes

DescribePushTaskGroupStatAllChannel	Queries the aggregated data of pushes in all channels	Yes
DescribePushTaskStatAllChannel	Queries the data of each push channel	Yes
DescribeTagsByToken	Queries tags bound to device	Yes
DescribeTokenInfos	Queries `tokenInfo` information	No
DescribePushInfos	Queries push list	Yes
ModifyAppInfo	Updates application information	Yes
ModifyProductInfo	Updates product information	No
CreatePush	Creates push	Yes
UpdateAppStatus	Updates application status	Yes
UploadCert	Uploads iOS certificate	Yes
UploadPushPackage	Uploads push number package	Yes
DescribePlanPushInfos	Queries the task list under push plan	Yes
DescribePushPlans	Queries push plan list	Yes
UpdatePushPlan	Modifies push plan	Yes
DeletePushPlan	Deletes push plan	Yes
CreatePushPlan	Creates push plan	Yes

Sample Policy for Operator

Suppose the main responsibility of an operator is to view push records and create pushes, then the operation permissions can be queried according to the list of authorizable operations:

- All query operations.
- Canceling scheduled push task.
- Creating push.

- Uploading push number package.
- Downloading push number package.

Suppose the current root account ID is 1000000000 and the `Access_id` values of authorized applications are 1500000000 and 1500000001, respectively;
The corresponding policy syntax should be as follows:

```
//
{
"version": "2.0",
"statement": [
{
"action": [
"tpns:Describe*",
"tpns:CancelPush",
"tpns:DownloadPushPackage",
"tpns:CreatePush",
"tpns:UploadPushPackage"
],
"resource": [
"qcs::tpns::uin/1000000000:app/1500000000","qcs::tpns::uin/1000000000:app/1500000001"
],
"effect": "allow"
},
{
"action": [
"tpns:Describe*"
],
"resource": [
"qcs::tpns::uin/1000000000:other/*"
],
"effect": "allow"
}

]
}
```

After creating a policy, you can find it on the Policy Management page of CAM and associate it with the sub-user to complete permission configuration. It can also be associated with other sub-users.

Sample Policy for Developer

Suppose the main responsibility of a developer is to access and test and all operation permissions should be granted;
Suppose the current root account ID is 1000000000 and the `Access_id` values of authorized applications are 1500000000 and 1500000001, respectively;
The corresponding policy syntax should be as follows:

```
//
{
"version": "2.0",
"statement": [
{
"action": "*",
"resource": [
"qcs::tpns::uin/1000000000:app/1500000000","qcs::tpns::uin/1000000000:app/1500000001"
],
"effect": "allow"
},
{
"action": [
"tpns:Describe*"
],
"resource": [
"qcs::tpns::uin/1000000000:other/*"
],
"effect": "allow"
}

]
}
```

After creating a policy, you can find it on the Policy Management page of CAM and associate it with the sub-user to complete permission configuration. It can also be associated with other sub-users.