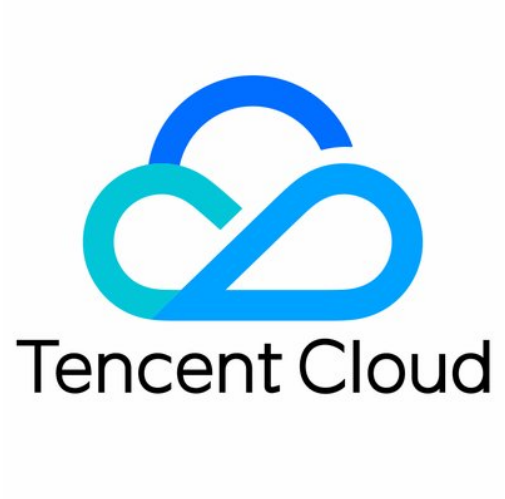
Cloud Message Queue

Operation Guide

Product Documentation



Contents

Operation Guide
Users and Permissions
Overview

Last updated：2021-09-22 18:01:20

In Tencent Cloud CMQ, you can use **users and permissions** to manage and control permissions and set permissions in **policy management** for specified **users** or **user groups** at the level of API or API set.

This document describes the dimensions of CMQ permission settings and how to set permissions.

1. Policy

Tencent Cloud uses **policies** to control relevant permissions of users or user groups. You can create an appropriate policy first and then associate it with specified users or user groups.

- Policy Definitions >>
- Policy Authorization Guide >>
- Policy Syntax Logic >>

2. User and User Group

When setting permissions, you need to specify the users or user groups that the permissions will be granted to; when setting a user or user group, you can associate it with relevant policies.

User

- User Types >>
- Sub-User Operations >>
- Collaborator Operations >>

User group

- Creating User Groups >>
- User Group Management >>
- User Group Permission Settings >>

3. CMQ Permission Dimensions

This section will be completed in the near future.

4. Sample CMQ Permission Settings

Setting Permissions Through API >>

Setting Permissions in Console >>

Setting Permissions in Console

Last updated：2021-06-08 11:37:06

Overview

This document describes how to grant a user with CMQ permissions by taking **write permissions for message consumption and batch message consumption** of the CMQ queue model as an example.

Permission Description

After CAM is connected, a sub-account can only view lists by default without any other operation permissions (the sub-account key is used for console login). The sub-account can get access permission only after it is authorized by the root account in CAM.

If the sub-account wants to view monitoring data in the console, it needs the permissions of Cloud Monitor APIs, which can be granted in CAM.

Directions

Creating a sub-user

1. Log in to the **CAM console**, select **Users** > **User List**, and click **Create User** on the upper left corner.
2. On the **Create User** page, you can choose **Create Now** or **Custom Create** to create a sub-user. For detailed directions, see Creating a Custom Sub-user.
3. After successful creation, you can view the newly created sub-user in **Users** > **User List**.

Creating a custom policy

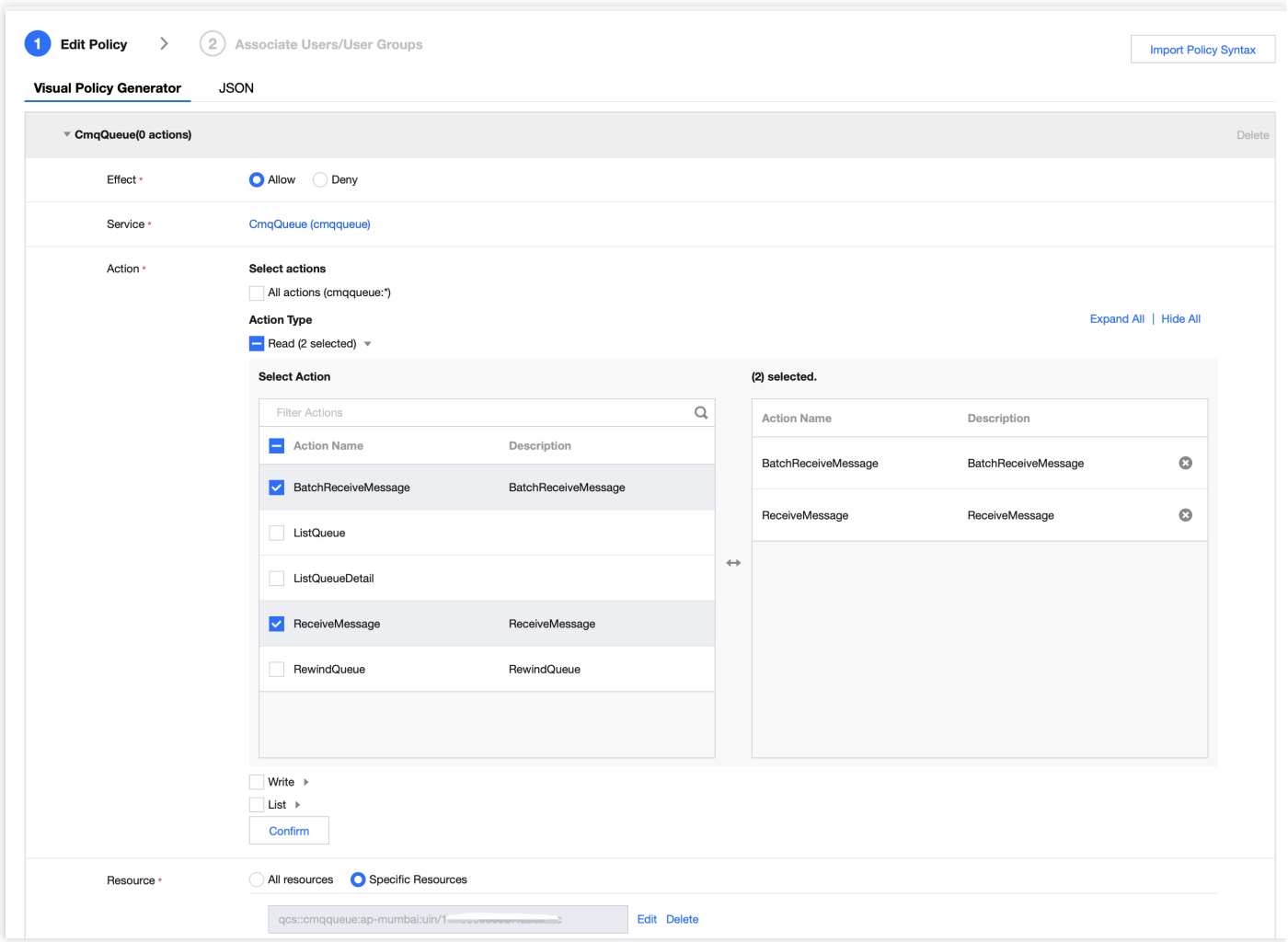
You can create a custom policy to grant the permissions of a specific API. The following takes the write permission (message consumption and batch message consumption) of CMQ queues as an example:

1. Log in to the CAM console, go to **Policies** from the left sidebar, and click **Create Custom Policy** on the upper left corner.

2. Select **Create by Policy Generator** in the pop-up dialog box.

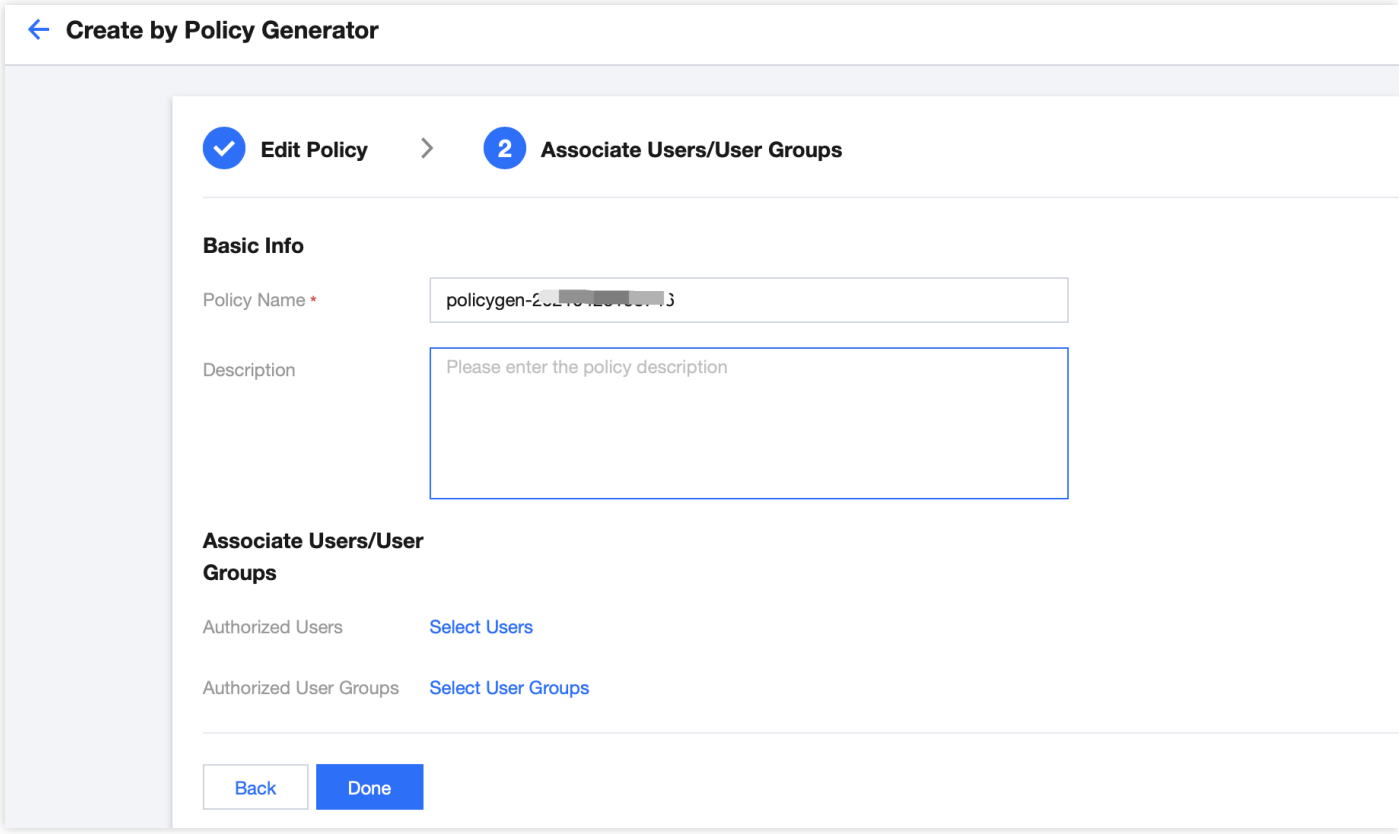
3. Provide the following information in the **Visual Policy Generator** tab.

- **Service** (required): select **CmqQueue (cmqqueue)** (if it is not found, please confirm whether you have activated the CMQ service).
- **Action** (required): select the actions you want to authorize.
- **Resource** (required): enter the six-segment description of the resource you want to authorize, for example, `qcs::cmqqueue:bj:uin/1238423:queueName/uin/3232/myqueue`. For more information, see [Authorization of CAM-Enabled APIs](#Authorization of CAM-Enabled APIs).
  - The first segment is always `qcs`.
  - The second segment is empty.
  - The third segment is the message queue type, which should be `cmqqueue` for queue model or `cmqtopic` for topic model.
  - The fourth segment is the region information, such as `gz`, `bj`, or `sh`. If you want to specify all regions, leave this segment empty.
  - The fifth segment is `uin/{root account uin}` of the root account.
  - The sixth segment is the resource description, which should be `queueName/uin/{creator Uin}/{queue name}` for queue model or `topicName/uin/{creator Uin}/{topic name}` for topic model. You can find the `creator Uin` on the details page in the console or in the returned value of `createUin` of the `GetQueueAttributes` or `GetTopicAttributes` API.
- **Condition** (optional): set the conditions that must be met for the authorization to take effect for the sub-account. For more information, see Condition.



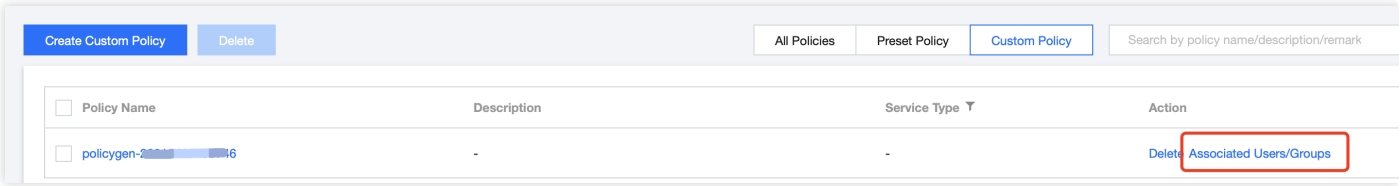
4. Click **Add Statement** > **Next** to go to the policy editing page.

5. On the policy editing page, set the policy name, add description, and confirm the policy content. The policy name and content are automatically generated by the console.
   - **Policy Name**: `policygen` by default. The suffix number is generated based on the creation date and can be customized.
   - **Policy Content**: corresponds to the service and actions selected in step 3. You can modify the content as needed.



6. Click **Done** to complete the custom policy creation .

7. In the policy list, select the target policy, click **Associated Users/Groups** in the **Action** column, select the users or user groups to associate, and click **Confirm** to complete the configuration.



For more information about CAM policies, see Policy.

Note：

The `list` API permissions of CMQ are all enabled by default (i.e., you can view the specific resource lists in the CMQ console after logging in). You can use the permissions to control what resource content can be displayed.

Authorizations of CAM-Enabled APIs

List of APIs supporting authorization at resource level

API Name	API Description	Resource Type	Example of Resource Six-Segment Description
ClearSubscriptionFilterTags	Clears the message tags of a subscriber.	Subscription API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`
CreateSubscribe	Creates a subscription API.	Subscription API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`
DeleteSubscribe	Deletes a subscription.	Subscription API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`
ModifySubscriptionAttribute	Modifies subscription attributes.	Subscription API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`
CreateTopic	Creates a topic.	Topic API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`
DeleteTopic	Deletes a topic.	Topic API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`
ModifyTopicAttribute	Modifies topic attributes.	Topic API	`qcs::cmqqueue:$region:uin/{root account uin}:topicName/uin/{creator Uin}/{topic name}`

API Name	API Description	Resource Type	Example of Resource Six-Segment Description
ClearQueue	Clears the messages in a queue.	Queue API	`qcs::cmqqueue:$region:uin/{root account uin}:queueName/uin/{creator Uin}/{queue name}`
CreateQueue	Creates a queue.	Queue API	`qcs::cmqqueue:$region:uin/{root account uin}:queueName/uin/{creator Uin}/{queue name}`
DeleteQueue	Deletes a queue.	Queue API	`qcs::cmqqueue:$region:uin/{root account uin}:queueName/uin/{creator Uin}/{queue name}`
ModifyQueueAttribute	Modifies queue attributes.	Queue API	`qcs::cmqqueue:$region:uin/{root account uin}:queueName/uin/{creator Uin}/{queue name}`

List of APIs not supporting authorization at resource level

API Name	API Description	Resource Type	Example of Resource Six-Segment Description
DescribeSubscriptionDetail	Queries subscription details.	Subscription API	`*`
DescribeTopicDetail	Queries topic details.	Topic API	`*`
DescribeDeadLetterSourceQueues	Enumerates the source queues of a dead letter queue.	Queue API	`*`
DescribeQueueDetail	Enumerates queues.	Queue API	`*`
RewindQueue	Rewinds a queue.	Queue API	`*`
UnbindDeadLetter	Unbinds a dead letter queue.	Queue API	`*`

Setting Permission Through API

Last updated：2020-07-14 15:08:49

Sub-user Key

Use a sub-account to log in to the CAM Console, enter **TencentCloud API Key**, and find the **sub-user key**, which is used to generate a signature for authentication. Successfully authenticated sub-accounts can access relevant Tencent Cloud resources.

Role of signature:

- Authenticate requesting user: the user key is used for authentication.
- Prevent content from being tampered with: the request content is signed with the hash algorithm, so that the system can check whether the content has been tampered with based on signature consistency.
- Prevent replay attacks: the signing information contains the request time and signature time and validity period, which can avoid replay of expired requests. In addition, Tencent Cloud services can reject expired requests based on request time.

Sample API Calls

API protocol

- Encoding type: UTF-8
- Encoding format: JSON
- Transfer method: POST
- Request protocol: HTTP

Call specification example:

```
{
"version": 1,
"componentName": "MC",
"eventId": 123456,
"interface": {
"interfaceName": "API name",
"para": {
Corresponding API parameters
}
```

```
}
}
```

Returned result:

```
{
"version": 1,
"eventId": 123456,
"componentName": "CONSOLE_LOGICAL_SERVER",
"returnValue": 0,
"returnCode": 0,
"returnMessage": "OK",
"data": {
"ownerUin": 123,
"uin": 124,
"ownerAppid": 323
}
}
```

If an error is returned, `returnCode` will not be 0, and `returnMessage` will be the error message.
For more information on `interfaceName` and `para` in the input parameters and `data` in the output parameters, please see [Call Description](#Call Description).

API description

For more information on CAM user and permission APIs, please see the CAM API documentation.

Sample call

Adding policy (CreateCamStrategy)

Policy example: this example shows you how to grant a sub-user (Uin: 3232) permission to list all queues under the account and permission to consume messages and delete messages in batches on `horacetest1` in the Beijing region.

- Field description

Parameter	Description	Example Value
strategyName	Policy name.	strategy1

Parameter	Description	Example Value
strategyInfo	Policy description (a **JSON string** needs to be passed in here).	For more information, please see [Sample code] (#Sample Code 1)
remark	Policy remarks.	hello test
resource	Six-segment description of CMQ resource, such as `qcs::cmqqueue:bj:uin/1238423:queueName/uin/3232/myqueue`. The first segment is fixed to `qcs`; The second segment is empty; The third segment indicates the message queue type, which is `cmqqueue` for the queue model or `cmqtopic` for the topic model; The fourth segment is the region information, such as gz, bj, and sh. If you want to specify all regions, leave this segment empty; The fifth segment is `uin/{root account uin}` of the root account; The sixth segment is the resource description, which is `queueName/uin/{creator Uin}/{queue name}` for the queue model or `topicName/uin/{creator Uin}/{topic name}` for the topic model. The creator `Uin` can be obtained on the details page in the console or through the returned value `createUin` of the `GetQueueAttributes` or `GetTopicAttributes` API.	*

- Sample code:

```
{
"strategyName":"strategy1",
"strategyInfo":{"version":"2.0","principal":{"qcs":["qcs::cam::uin/1238423:uin/3232/myqueue","qcs::cam::uin/1238423:groupid/13"]},"statement":[{"effect":"allow","action":"name/cmqqueue:ListQueue","resource":"*"},{"effect":"allow","action":["name/cmqqueue:ReceiveMessage","name/cmqqueue:BatchDeleteMessage"],"resource":["qcs::cmqqueue:bj:uin/1238423:queueName/uin/3232/myqueue","qcs::cmqqueue:bj:uin/1238423:queueName/uin/3232/*"]}]},
"remark":"horace test"
}
```

Note：

> The creator ID after `uin/` in the resource description in the sixth segment can be found during policy creation.

Associating/Unassociating policy with/from sub-account (OperateCamStrategy)

This API is used to associate/unassociate a policy with/from a user or user group.

- Policy example: this example shows you how to associate a policy (ID: 666) with a user (UIN: 123456).
- Field description:

Parameter	Description	Example Value
groupId	If the operation object is a user, set `groupId` to `-1`; If the operation object is a user group, set `groupId` to a specific group ID.	-1
relateUin	If the operation object is a user, set `relateUin` to a specific user `uin`; if the operation object is a user group, set `relateUin` to `-1`.	123456
strategyId	Target policy ID.	666
actionType	1: associates policy; 2: unassociates policy.	1

- Sample code:

```
{
"groupId":-1,
"relateUin":123456,
"strategyId":666,
"actionType":1
}
```

">

Call Description

The following description is applicable to user and permission management in various services. When configuring the CMQ service, please select values for CMQ parameters accordingly:

1. You can leave `principal` empty and associate the user by using the policy associating API.

2. If there is only one element in `principal`, `action`, or `resource`, you do not need to add `[]`.
3. `resource` is generally described in a six-segment format of `qcs:project:serviceType:region:account:resource`.
   - project: you can use `id/0`, `*`, or `id/*` to indicate all projects. If `project` is empty during authorization, the value will be `id/0` by default. If `project` is empty during authentication, it indicates that the resource can exist in all projects. This segment is empty by default.
   - serviceType: valid values include `cos`, `cdn`, `vpc`, etc. `*` indicates all services. You cannot leave this segment empty.
   - region: it specifies the region. If this segment is empty, it indicates all regions. It is empty by default. Valid values include `gz`, `st`, `tj`, `sh`, `hk`, `ca`, `shjr`, and `bj`.
   - account: it can be represented as `uin/${uin}` or `uid/${uid}`. If this segment is empty, it will be populated with `uin/${uin}` for resources of services such as CDN and VPC or with `uid/${uid}` for COS resources. `${uin}` and `${uid}` indicate the `uin` and `uid` of the requester, respectively. This segment is empty by default.
     There is a special case: `uin/-1` is generally used in preset policies. After the extension table is expanded, `-1` will be replaced with the developer `uin`. In addition, preset policies support authorization for sub-accounts and roles only; therefore, you can directly replace `-1` with the `uin` of the root account of the sub-account or role.
   - `resource` consists of `name` and `value`. `name` represents the resource definition in the service; for example, it is described as `queueName` or `topicName` for CMQ, `prefix` for COS, and `host` for CDN. `*` indicates all resources, which will be represented as `*/*` uniformly. This segment cannot be empty.
   - Users and policies are also resources. A CAM root account is described as `qcs::cam::uin/1238423: uin/1238423`, a CAM sub-account is described as `qcs::cam::uin/1238423: uin/3236671`, and an anonymous user is described as `qcs::cam::anonymous:anonymous`.
   - If `resource` is empty, it indicates that no objects need to be associated with the operation, which will be represented as `*` in the system uniformly.
   - The service needs to verify whether the `uin` or `uid` in the resource description is the real resource owner. It is required that the service perform verification after successful authentication. It is recommended to perform verification during authentication as well.

Resource Tag Use Case

Last updated：2020-04-28 15:31:24

The CMQ resource tagging feature was relaunched on September 9, 2019, which is automatically compatible with previously configured resource tags.

Overview

A resource tag is a key-value pair provided by Tencent Cloud to identify a resource in the cloud.
You can use resource tags to classify CMQ resources based on various factors such as service, usage, and person in charge. With resource tags, you can quickly sift through the resource pool and find the corresponding resources. The values of resource tag keys do not mean anything to Tencent Cloud semantically and will be parsed and matched strictly according to the string.
Below is a specific use case to show how a resource tag is used.

Use Case Background

A company owns 10 CMQ instances in Tencent Cloud. Distributed in three departments (ecommerce, gaming, and entertainment), these instances are used to serve internal business lines such as marketing, game A, game B, and post-production. The OPS owners of the three departments are John, Jane, and Harry, respectively.

Directions

Setting resource tag

To facilitate management, the company categorizes its CMQ resources with resource tags and defines the following resource tag key-value pairs:

Resource Tag Key	Tag Value
Department	Ecommerce, gaming, and entertainment
Business	Marketing, game A, game B, and post-production

Resource Tag Key	Tag Value
OPS owner	John, Jane, and Harry

These resource tags are bound to CMQ instances in the following way:

ID	Department	Business	OPS Owner
queue-pale1	Ecommerce	Marketing	Harry
queue-pale12	Ecommerce	Marketing	Harry
queue-pale13	Gaming	Game A	John
queue-pale13	Gaming	Game B	John
queue-pale14	Gaming	Game B	John
queue-pale15	Gaming	Game B	Jane
queue-pale16	Gaming	Game B	Jane
queue-pale17	Gaming	Game B	Jane
queue-pale18	Entertainment	Post-production	Harry
queue-pale19	Entertainment	Post-production	Harry
queue-pale110	Entertainment	Post-production	Harry

Using resource tag

- Filter out the CMQ instances in the charge of Harry
  Filter out the CMQ instances where the OPS owner is "Harry". For detailed directions, please see Using Resource Tags.

- Filter out the CMQ instances in the charge of Jane in the gaming department
  Filter out the CMQ instances where the department is "gaming" and OPS owner is "Jane". For detailed directions, please see Using Resource Tags.

Using Resource Tags

Last updated：2022-03-29 16:29:41

Operation Scenarios

This document describes how to filter resources of the target queue by resource tag in the queue service in the CMQ Console.

Directions

1. Log in to the CMQ Console and select **Queue Service** > **Queue**.
2. Select the region at the top of the queue page.
3. In the search box in the top-right corner of the queue list, click the blank space and select **Resource Tag**.
4. Enter the tag key after **Resource Tag:**.
   For example, to filter out instances bound to a specified tag key (such as `name`), enter `name` (pay attention to the letter case).
5. Press Enter or click the search icon.