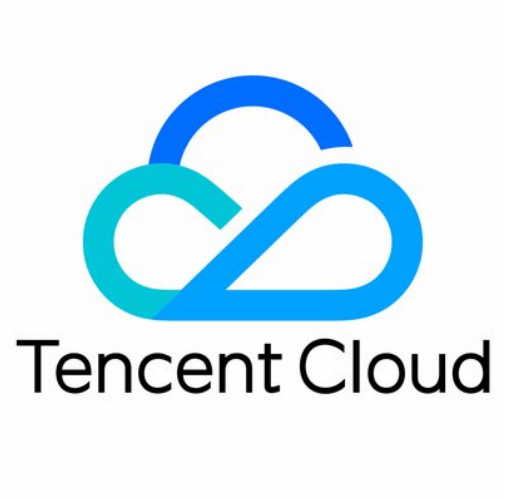
Cloud Access Management

Business Use Cases

Product Documentation



Contents

Business Use Cases
TencentDB for MySQL
Allowing Account to View TencentDB for MySQL Instances Under Specified Tag

Last updated：2022-09-19 16:02:04

The enterprise account “CompanyExample” (ownerUin: 12345678) has a sub-account “Developer” that requires permissions to view its two TencentDB for MySQL instances (instance IDs: “cdb-1” and “cdb-2”, with the tags being “game&webpage” and “game&app”, respectively).

Step 1: Create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"effect": "allow",
"action": [
"cdb:Describe*"
],
"resource": "*",
"condition": {
"for_any_value:string_equal": {
"qcs:resource_tag": [
"game&webpage",
"game&app"
]
}
}
}
]
}
```

Step 2. Authorize the policy to the sub-account. For more information on authorization, see Authorization Management.

Note：
The sub-account “Developer” can only view the resources of instances with the IDs being “cdb-1” and “cdb-2” in the TencentDB for MySQL query list.

Granting a Sub-account View Permission for Specified TencentDB for MySQL Instances

Last updated：2021-06-07 09:40:26

A sub-account, Developer, under the enterprise account, CompanyExample (ownerUin: 12345678), requires view permission for two TencentDB for MySQL instances (instance IDs cdb-1 and cdb-2 respectively) belonging to the CompanyExample enterprise account.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "cdb:*",
"resource": ["qcs::cdb::uin/12345678:instanceId/cdb-1", "qcs::cdb::uin/12345678:i
nstanceId/cdb-2"]
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Note: The Developer sub-account can only view the resources of instances with the IDs cdb-1 and cdb-2 in the TencentDB for MySQL query list.

CLB

Authorizing Sub-account Full Access to CLB （Includes payment permission）

Last updated：2022-07-14 10:48:07

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires full management permissions (creation, management, ordering, and payment for CLB) for the CLB service of enterprise account CompanyExample.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policies QcloudCLBFullAccess and QcloudCLBFinanceAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy according to policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "clb:*",
"resource": "*"
},
{
"effect": "allow",
"action": "finance:*",
"resource": "qcs::clb:::*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Read-only Access to CLB

Last updated：2019-12-25 15:58:10

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires read-only permission for the CLB service under the CompanyExample enterprise account. The sub-account is not permitted to create, update, or delete CLBs.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCLBReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

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

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Full CLB Access other than the Payment Permission

Last updated：2020-02-26 18:04:28

Authorizing a sub-account with all permissions of CLB except payment

A sub-account Developer under the enterprise account CompanyExample (ownerUin is 12345678) requires full management permissions (such as creation and management) of the enterprise account CompanyExample's CLB service, except the payment permission. The sub-account is allowed to place an order but cannot pay for it.

Solution A:

The enterprise account CompanyExample directly authorizes the preset policy QcloudCLBFullAccess to the sub-account Developer. For more information on authorization, please see Authorization Management.

Solution B:

Step 1: Create the following policy using policy syntax

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

Step 2: Authorize the policy to the sub-account. For more information on authorization, please see Authorization Management.

CMQ
Authorizing a Sub-account Full Permissions to Use Messaging Services

Last updated：2019-12-25 15:59:37

The enterprise account, CompanyExample, has a sub-account, Developer, that requires full permissions to read and write the message queue of CompanyExample. The message queue for read and write permissions can be either topic model or queue model.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policies QCloudCmqQueueFullAccess and QCloudCmqTopicFullAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": ["cmqtopic:*","cmqqueue:*"],
"resource": "*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing a Sub-account Full Permissions to Access the Message Queue It Created

Last updated：2019-12-25 16:03:29

A sub-account, Developer, under the enterprise account, CompanyExample, requires access to the message queue it created.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policies QCloudCmqQueueCreaterFullAccess and QCloudCmqTopicCreaterFullAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":
[
{
"effect": "allow",
"action": "cmqtopic:*",
"resource": "qcs::cmqtopic:::topicName/uin/${uin}/*"
},
{
"effect": "allow",
"action": "cmqqueue:*",
"resource": "qcs::cmqqueue:::queueName/uin/${uin}/*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing a Sub-account Permission to Read a Topic-based Message Queue

Last updated：2019-12-25 16:06:19

The enterprise account, CompanyExample (ownerUin: 12345678), wants to give its sub-account, Developer, access to its topic-based message queue.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": "cmqqueue:SendMessage",
"resource":"qcs::cmqqueue:::queueName/uin/12345678/test-caten",
"effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

COS

Authorizing Sub-account Full Access to COS Resources under the Account

Last updated：2019-12-25 16:07:42

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires full management permissions (creating, managing and accessing COS buckets or objects) for the COS service under the CompanyExample enterprise account.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCOSFullAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

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

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Full Access to Specific Directory

Last updated：2020-06-10 15:15:18

The enterprise account, CompanyExample (ownerUin: 12345678; appID: 1250000000), has a sub-account, Developer, that requires full access permissions to the dir1 directory of the Bucket1 bucket of the COS service in Shanghai region under the CompanyExample enterprise account.

Solution A:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "cos:*",
"resource": ["qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/*",
"qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1"]
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Configure the policy and ACL in the COS Console. For more information, see the COS documentation.

Authorizing Sub-account Read-only Access to Files in Specific Directory

Last updated：2020-06-10 15:17:33

The enterprise account, CompanyExample (ownerUin: 12345678; appID: 1250000000), has a sub-account, Developer, that requires read permissions for files under the dir1 directory of the Bucket1 bucket of the COS service in Shanghai region under the CompanyExample enterprise account.

Solution A:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": [
"cos:List*",
"cos:Get*",
"cos:Head*",
"cos:OptionsObject"
],
"resource": "qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Configure the policy and ACL in the COS Console. For more information, see the COS documentation.

Authorizing Sub-account Read/Write Access to Specific File

Last updated：2020-06-10 15:22:24

The enterprise account, CompanyExample (ownerUin: 12345678; appID: 1250000000), has a sub-account, Developer, that requires read/write permissions for the object, Object1, under the dir1 directory of the Bucket1 bucket of the COS service in Shanghai region under the CompanyExample enterprise account.

Solution A:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "cos:*",
"resource": "qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/object1"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Configure the policy and ACL in the COS Console. For more information, see the COS documentation.

Authorizing Sub-account Read-only Access to COS Resources

Last updated：2019-12-25 16:08:55

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires read-only access permissions to COS buckets, objects, and object lists belonging to the CompanyExample enterprise account.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCOSReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": [
"cos:List*",
"cos:Get*",
"cos:Head*",
"cos:OptionsObject"
],
"resource": "*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing a Sub-account Read/Write Access to All Files in Specified Directory Except Specified Files

Last updated：2020-05-15 10:55:56

The organizational account `CompanyExample` (ownerUin: 12345678; appId: 1250000000) has a sub-account `Developer` that requires read/write permissions for all objects except the `Object1` object in the `dir1` directory of the `Bucket1` bucket of the COS service in the Shanghai region under the `CompanyExample` account.

Solution A:

Step 1. Create the following policy according to the policy syntax:

```
{
"version": "2.0",
"statement":
[
{
"effect": "allow",
"action": "cos:*",
"resource": "qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/*"
},
{
"effect": "deny",
"action": "cos:*",
"resource": "qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/Object1"
}
]
}
```

Step 2. Associate the policy with the sub-account. For more information on authorization, please see Authorization Management.

Solution B:

Set the policy and ACL in the COS Console. For more information, please see ACL Practices.

Authorizing Sub-account Read/Write Access to Files with Specified Prefix

Last updated：2020-06-10 15:24:00

The enterprise account, CompanyExample (ownerUin: 12345678; appID: 1250000000), has a sub-account, Developer, that requires read/write permissions for objects with the prefix “test” under the dir1 directory of the Bucket1 bucket of the COS service in Shanghai region under the CompanyExample enterprise account.

Solution A:

Step 1: create the following policy according to policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "cos:*",
"resource": "qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/test*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Configure the policy and ACL in the COS Console. For more information, see the COS documentation.

Authorizing Another Account Read/Write Access to Specific Files

Last updated：2020-05-15 10:55:56

The organizational account `CompanyGranter` (ownerUin: 12345678; appId: 1250000000) has an object `Object1` located in the `dir1` directory of the `Bucket1` bucket in the Guangzhou region. Another organizational account `CompanyGrantee` (ownerUin: 87654321) requires read/write permissions for this object.

Set the policy and ACL in the COS Console. For more information, please see ACL Practices.

Authorizing Cross-Account ’s Sub-account Read/Write Access to Specified File

Last updated：2020-06-10 15:25:58

The enterprise account, CompanyGranter (ownerUin: 12345678; appID: 1250000000), has an object, Object1, that is located in the dir1 directory of the Bucket1 bucket in the Guangzhou region. The sub-account of another enterprise account, CompanyGrantee (ownerUin: 87654321), requires read/write permission for Object1.

This involves permission propagation.

Step 1: CompanyGrantee creates the following policy according to policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "cos:*",
"resource": "qcs::cos:ap-shanghai:uid/1250000000:Bucket1-1250000000/dir1/Object1"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Step 3: the CompanyGranter enterprise account grants CompanyGrantee enterprise account access to Object1 by configuring the policy and ACL in the COS Console. For more information, see COS documentation.

CVM

Authorizing Sub-account Full Access to CVMs

Last updated：2019-12-25 16:11:57

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires full management permissions (creation, management, ordering, and payment for CVMs) for the CVMs of the CompanyExample enterprise account.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policies QcloudCVMFullAccess and QcloudCVMFinanceAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "cvm:*",
"resource": "*"
},
{
"effect": "allow",
"action": "finance:*",
"resource": "qcs::cvm:::*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Read-only Access to CVMs

Last updated：2019-12-25 16:12:53

The enterprise account, CompanyExample (ownerUIN: 12345678), has a sub-account, Developer, that requires CVM instance query permission for the CVM service under the CompanyExample enterprise account. The sub-account is not allowed to create, delete, or start up/shut down CVM instances.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCVMInnerReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

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

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Read-only Access to CVM-related Resources

Last updated：2019-12-25 16:13:44

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires query permission for CVM instances and related resources (VPC, CLB) for the CVM service under the CompanyExample enterprise account. The sub-account is not allowed to create, delete, or start up/shut down CVM instances.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCVMReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"cvm:Describe*",
"cvm:Inquiry*"
],
"resource": "*",
"effect": "allow"
},
{
"action": [
"vpc:Describe*",
"vpc:Inquiry*",
"vpc:Get*"
],
"resource": "*",
"effect": "allow"
},
{
"action": [
"clb:Describe*"
],
"resource": "*",
"effect": "allow"
},
```

```
{
"effect": "allow",
"action": "monitor:*",
"resource": "*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on CBSs

Last updated：2019-12-25 16:14:42

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires permissions to view, create and use cloud disks in the CVM Console belonging to the CompanyExample enterprise account.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCBSFullAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"cvm:CreateCbsStorages",
"cvm:AttachCbsStorages",
"cvm:DetachCbsStorages",
"cvm:ModifyCbsStorageAttributes",
"cvm:DescribeCbsStorages",
"cvm:DescribeInstancesCbsNum",
"cvm:RenewCbsStorage",
"cvm:ResizeCbsStorage"
],
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Note: If the sub-account is not allowed to modify cloud disk properties, remove `cvm:ModifyCbsStorageAttributes` from the policy syntax.

Authorizing Sub-account Access to Perform Operations on Security Groups

Last updated：2019-12-25 16:15:34

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires permissions to view and use security groups belonging to the CompanyExample enterprise account in the CVM Console.

The following policy gives the sub-account permission to create and delete security groups in the CVM Console.

Step 1: Create the following policy using policy syntax

```
{
"version": "2.0",
"statement": [
{
"action": [
"cvm:DeleteSecurityGroup",
"cvm:CreateSecurityGroup"
],
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

The following policy grants the sub-account permission to create, delete, and modify security group policies in the CVM Console.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"cvm:ModifySecurityGroupPolicy",
"cvm:CreateSecurityGroupPolicy",
"cvm:DeleteSecurityGroupPolicy"
],
```

```
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on EIPs

Last updated：2019-12-25 16:17:09

The enterprise account, `CompanyExample` (ownerUin: 12345678), has a sub-account, `Developer`, that requires permissions for the CVM service under the `CompanyExample` enterprise account. The sub-account needs to view Elastic IPs (EIPs) in the CVM Console, and use the EIPs.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"cvm:AllocateAddresses",
"cvm:AssociateAddress",
"cvm:DescribeAddresses",
"cvm:DisassociateAddress",
"cvm:ModifyAddressAttribute",
"cvm:ReleaseAddresses"
],
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the policy with the sub-account. For more information on authorization, see Authorization Management.

The following policy allows a sub-account to view the EIPs and associate them with instances. The sub-account can modify the attributes of the EIPs, disassociate them from instances, and release the EIPs.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"cvm:DescribeAddresses",
"cvm:AllocateAddresses",
```

```
"cvm:AssociateAddress"
],
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the policy with the sub-account. For more information on authorization, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on Specific CVM

Last updated：2022-08-09 10:24:06

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires permissions to operate a specific CVM (ID: ins-1) in Guangzhou region belonging to the enterprise account CompanyExample.

Step 1: create the following policy according to policy syntax.

```
{
"version": "2.0",
"statement": [
{
"effect": "allow",
"action": [
"cvm:*",
"vpc:DescribeVpcEx",
"vpc:DescribeNetworkInterfaces"
],
"resource": "*",
"condition": {
"for_any_value:string_equal": {
"qcs:resource_tag": [
"game&webpage"
]
}
}
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on CVMs in Specific Region

Last updated：2020-02-26 18:14:21

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires operating permissions for all CVMs in Guangzhou region under the CompanyExample enterprise account.

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudCVMReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": "cvm:*",
“resource": "qcs::cvm:gz::*",
“effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Full Access to CVMs Except Payment

Last updated：2020-02-26 18:12:47

The enterprise account, CompanyExample, whose ownerUin is 12345678, has a sub-account, Developer, that requires full management permissions (including all operations such as creation and management) for the CVM service of the CompanyExample enterprise account. These permissions do not include payment permissions, but allow orders to be made.

Solution A:

The CompanyExample enterprise account directly authorizes the preset policy QcloudCVMFullAccess to the Developer sub-account. For more information about authorization, see Authorization Management.

Solution B:

Step 1. Create the following policy according to policy syntax.

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

Step 2. Authorize the policy to the sub-account. For more information about authorization, see Authorization Management.

VPC
Authorizing Sub-account Read-only Access to VPCs

Last updated：2019-12-25 16:21:35

The enterprise account, CompanyExample (ownerUIN: 12345678), has a sub-account, Developer, that requires read-only permission for VPC services under the CompanyExample enterprise account. This permission allows the sub-account to query VPCs and related resources, but will not permit the sub-account to create, update, or delete CLBs.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy QcloudVPCReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"vpc:Describe*",
"vpc:Inquiry*",
"vpc:Get*"
],
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on Specific VPC and Resources of This VPC

Last updated：2019-12-25 10:20:16

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires operating permissions for a specific VPC (ID: vpc-id1) and relevant resources (e.g., subnets, routing tables, but not CVMs and databases) of the VPC service belonging to the CompanyExample enterprise account.

Step 1: create the following policy according to policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": "vpc:*",
"resource": "*",
"effect": "allow",
"condition": {
"string_equal_if_exist": {
"vpc:vpc": [
"vpc-id1"
],
"vpc:accepter_vpc": [
"vpc-id1"
],
"vpc:requester_vpc": [
"vpc-id1"
]
}
}
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on VPC Except on Routing Table

Last updated：2019-12-25 16:22:45

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires read/write permissions for VPCs and relevant resources (except for routing tables) belonging to the CompanyExample enterprise account.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"vpc:*"
],
"resource": "*",
"effect": "allow"
},
{
"action": [
"vpc:AssociateRouteTable",
"vpc:CreateRoute",
"vpc:CreateRouteTable",
"vpc:DeleteRoute",
"vpc:DeleteRouteTable",
"vpc:ModifyRouteTableAttribute"
],
"resource": "*",
"effect": "deny"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Access to Perform Operations on VPN

Last updated：2019-12-25 16:23:56

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires view permission for all VPC resources in the VPC service under the CompanyExample enterprise account, but is only allowed to perform CRUD operations on VPNs.

Step 1: create the following policy by using policy syntax.

```
{
"version": "2.0",
"statement": [
{
"action": [
"vpc:Describe*",
"vpc:Inquiry*",
"vpc:Get*"
],
"resource": "*",
"effect": "allow"
},
{
"action": [
"vpc:*Vpn*",
"vpc:*UserGw*"
],
"resource": "*",
"effect": "allow"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Full Access to VPCs

Last updated：2019-12-25 16:20:08

The enterprise account, CompanyExample (ownerUin: 12345678), has a sub-account, Developer, that requires full management permissions (creation, management, ordering, and payment for VPCs) for the VPC service of the CompanyExample enterprise account.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policies QcloudVPCFullAccess and QcloudVPCFinanceAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy according to policy syntax.

```
{
"version": "2.0",
"statement":[
{
"effect": "allow",
"action": "vpc:*",
"resource": "*"
},
{
"effect": "allow",
"action": "finance:*",
"resource": "qcs::vpc:::*"
}
]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing a Sub-account Full Access to VPCs Except Payment

Last updated：2020-05-15 10:55:57

The organizational account `CompanyExample` (ownerUin: 12345678) has a sub-account `Developer` that requires full management permissions (for all operations such as creation and management) for the VPC service under `CompanyExample` except payment permissions. The sub-account should be able to place orders but cannot make payments.

Solution A:

The `CompanyExample` account directly authorizes the preset policy `QcloudVPCFullAccess` to the `Developer` sub-account. For more information on authorization, please see Authorization Management.

Solution B:

Step 1. Create the following policy according to the policy syntax:

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

Step 2. Associate the policy with the sub-account. For more information on authorization, please see Authorization Management.

VOD
Authorizing a Sub-account with Full Permissions to Manage VOD Services

Last updated：2020-02-26 18:03:19

Grant a sub-account full permissions to manage Tencent Cloud VOD services

A sub-account Developer under the enterprise account CompanyExample (ownerUin: 12345678) requires full permissions to manage Tencent Cloud VOD services in the enterprise account.

Solution A:

The enterprise account CompanyExample directly authorizes the preset policy QcloudVODFullAccess to the sub-account Developer.

Solution B:

Step 1: Create the following policy using policy syntax

```
{
"version": "2.0",
"statement": [
{
"action": [
"vod:*"
],
"resource": "*",
"effect": "allow"
},
{
"action": "cos:*",
"resource": "qcs::cos::uid/10022853:*",
"effect": "allow"
}
]
}
```

Step 2: Authorize the policy to the sub-account.

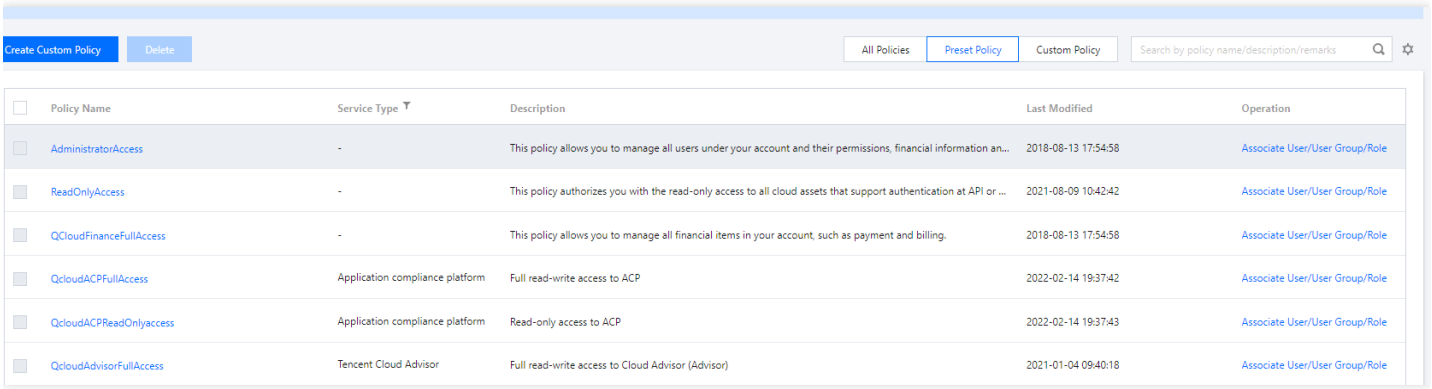
Others

Granting Management or Read-Only Permissions for Specified Product

Last updated：2022-09-19 16:02:04

CAM provides default (or preset) authorization policies to help you use it for your Tencent Cloud services, with at least preset management and read-only policies available for each service category. You can use these preset policies to associates CAM sub-accounts or user groups to control the access permissions for a service.

Step 1:
In the **CAM** console, select **Policies** on the left sidebar. Enter a Tencent Cloud service name (such as CVM) in the search box, and you can view a list of preset policies for this service.



`QcloudCVMFullAccess` is a management policy, and `QcloudCVMInnerReadOnlyAccess` is a read-only policy.

> Note：
> The management policies of some services do not include payment permissions. You can associate a default payment management policy (such as `QcloudCVMFullAccess`) with the CAM sub-user/user group you want to authorize.

Step 2:
Authorize policies in the above list to the CAM sub-account as needed. For more information on authorization, see Authorization Management.

- If you want to grant all the management permissions of a Tencent Cloud account to a sub-user, you can use the preset policy `AdministratorAccess`.

`AdministratorAccess` : This policy allows you to manage all users and their permissions, related financial information, and cloud service assets under this account.

- If you want to grant the read-only permissions of a Tencent Cloud account to a sub-user, you can use the preset policy `ReadOnlyAccess` .
  `ReadOnlyAccess` " This policy allows you to access all the cloud service assets that support API-level or resource-level authentication under a Tencent Cloud account in a read-only manner.

Authorizing Sub-account Access to Perform Operations on All Resources

Last updated：2019-12-25 16:25:07

A sub-account, Developer, under the enterprise account, CompanyExample, requires full access permission to all resources belonging to the enterprise account.

Solution A:

The CompanyExample enterprise account directly associates the Developer sub-account with the preset policy AdministratorAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

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

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Sub-account Read-only Access to All Resources

Last updated：2020-08-03 11:41:38

A sub-account, Developer, under the enterprise account, CompanyExample, requires read-only permissions for all resources belonging to the enterprise account.

Solution A:

The enterprise account, CompanyExample, associates the Developer sub-account with the preset policy ReadOnlyAccess. To learn how to associate a policy with a user account, see Authorization Management.

Solution B:

Step 1: create the following policy by using policy syntax.

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:Describe*",
                "cvm:Inquiry*",
                "vpc:Describe*",
                "vpc:Inquiry*",
                "vpc:Get*",
                "clb:Describe*",
                "monitor:Describe*",
                "monitor:Get*",
                "bm:Describe*",
                "bmeip:Describe*",
                "bmlb:Describe*",
                "bmvpc:Describe*",
                "bm:Get*",
                "bmlb:Get*",
                "cos:List*",
                "cos:Get*",
                "cos:Head*",
                "cos:OptionsObject",
                "cas:Describe*",
                "cas:List*",
                "cas:Get*",
                "kms:List*",
                "kms:Get*",
```

```
                "ccs:Describe*",
                "ccs:Check*",
                "cam:Get*",
                "cam:List*",
                "cam:Describe*",
                "cam:Query*",
                "cdb:Describe*",
                "batch:Describe*",
                "bgpip:BasicGet*",
                "bgpip:BasicCCGet*",
                "bgpip:BasicDDoSGet*",
                "bgpip:BgpGetFpcDeviceList",
                "bgpip:BGPGetInfo",
                "bgpip:BGPGetServiceStatistics",
                "bgpip:BGPGetServicePacks",
                "bgpip:BGPCCGet*",
                "bgpip:BGPWhitelistGet",
                "bgpip:Get*",
                "bgpip:BGPIPWhitelistGet",
                "bgpip:BGPIPGet*",
                "bgpip:BGPIPDDoSGet*",
                "bgpip:BGPIPCCGet*",
                "bgpip:BgpipGetIdByTran",
                "bgpip:BgpipModifyPrice",
                "bgpip:BgpipRenewPrice",
                "bgpip:BgpipCreatePrice",
                "bgpip:BgpipQueryResources",
                "bgpip:BgpipCheckModify",
                "bgpip:BgpipCheckRenew",
                "bgpip:BgpipCheckCreate",
                "bgpip:BGPDDoSGet*",
                "ccb:ListGitAuth",
                "ccr:pull",
                "ccs:Describe*",
                "ccs:Check*",
                "ckafka:Get*",
                "ckafka:List*",
                "organization:Get*",
                "organization:List*",
                "redis:Describe*",
                "scf:Get*",
                "scf:List*",
                "shield:*Get*",
                "tag:Get*",
                "waf:WafGet*",
                "waf:WAFGetUserInfo",
                "waf:WafDownloadAlerts",
```

```
                "waf:WafPackagePrice",
                "waf:WafAreaBanGetAreas",
                "waf:WafFreqGetRuleList",
                "waf:WafAntiFakeGetUrl",
                "waf:WafDNSdetectGet*",
                "waf:BotGet*",
                "wss:CertGetList",
                "cbm:previewProductDetail",
                "cbm:agentInfo",
                "cbm:viewDeals",
                "cbm:rebateInfo",
                "cbm:businessDetail",
                "cbm:inviteClient",
                "cbm:viewClients",
                "cbm:authorize",
                "cbm:viewMessage",
                "cbm:viewMenu",
                "snova:Describe*",
                "gme:Describe*",
                "gme:Download*"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

Step 2: associate the sub-account with the policy. To learn how to associate a policy with a user account, see Authorization Management.

Authorizing Different Sub-accounts Separate Permissions to Manage Tencent Cloud Resources

Last updated：2020-06-03 11:40:19

Introduction

If you have purchased different Tencent Cloud resources, you can use tags to group the resources for easy management. You can grant different sub-accounts management permissions by tags so that they can manage resources separately. This document takes a use case as an example to describe how to grant a sub-account the permission to manage separate Tencent Cloud resources by using tags.

Prerequisites

Suppose that:

- The enterprise account `CompanyExample` has two sub-accounts `DevA` and `DevB` .
- The ID of sub-account `DevA` is `12345` .
- The ID of sub-account `DevB` is `67890` .
- The enterprise account `CompanyExample` has two CVM instances whose IDs are `ins-1` and `ins-2` respectively.
- The enterprise account `CompanyExample` has two tag keys (`test1` and `test2`) and two tag values (`test1` and `test2`).

Directions

Tagging CVM instances

You can add tag keys and tag values to CVM instances `ins-1` and `ins-2` with the following steps to manage resources by tag.

Adding `test1` tag key and `test1` tag value to CVM instance `ins-1`

1. Log in to the Tag Console, set the following filters to filter out the target CVM instance, and click **Query Resource**.
   - Resource Type: type of the resource to be queried. Only products supporting tags can be queried. For more information, please see Products That Support Tags. In this example, select CVM instance.
   - Region: region of the resource to be queried. In this example, select Beijing.
2. Select the target CVM instance from the filtered results. In this example, we select CVM instance `ins-1`.
3. Click **Edit Tag Value**.
4. In the pop-up window, select the tag key and enter the tag value. In this example, the tag key and value are both `test1`.
5. Click **OK** to add `test1` tag key and `test1` tag value to CVM instance `ins-1`.

Adding `test2` tag key and `test2` tag value to CVM instance `ins-2`

1. Log in to the Tag Console, set the following filters to filter out the target CVM instance, and click **Query Resource**.
   - Resource Type: type of the resource to be queried. Only products supporting tags can be queried. For more information, please see Products That Support Tags. In this example, we select CVM instance.
   - Region: region of the resource to be queried. In this example, we select Beijing.
2. Select the target CVM instance from the filtered results. In this example, we select CVM instance `ins-2`.
3. Click **Edit Tag Value**.
4. In the pop-up window, select the tag key and enter the tag value. In this example, the tag key and value are both `test2`.
5. Click **OK** to add `test2` tag key and `test2` tag value to CVM instance `ins-2`.

Authorizing user by tag

You can grant sub-account `DevA` management permission for tag key `test1` and tag value `test1` and grant sub-account `DevB` tmanagement permission for tag key `test2` and tag value `test2`. They will then be able to manage tagged resources accordingly.

Granting sub-account `DevA` management permission for tag key `test1` and tag value `test1`

1. Log in to the CAM Console and click **Create Custom Policy** in the top-left corner.
2. In the creation method selection window that pops up, click **Authorize by Tag** to enter the authorization by tag page.
3. Select the following information and click **Next**.
   - Authorize User/User Group: check the user/user group to be authorized. In this example, `12345` is selected, which is the ID of sub-account `DevA`.
   - Tag Key: select the tag key to be authorized. In this example, we select tag key `test1`.
   - Tag Value: select the tag value to be authorized. In this example, we select tag value `test1`.
   - Resources: the management permission is granted by default.

4. On the verification page, enter the policy name, verify the policy content, and click **Done** to grant sub-account `DevA` management permission for tag key `test1` and tag value `test1`.

Granting sub-account `DevB` management permission for tag key `test2` and tag value `test2`

1. Log in to the CAM Console and click **Create Custom Policy** in the top-left corner.
2. In the creation method selection window that pops up, click **Authorize by Tag** to enter the authorization by tag page.
3. Select the following information and click **Next**.
   - Authorize User/User Group: check the user/user group to be authorized. In this example, `67890` is selected, which is the ID of sub-account `DevB`.
   - Tag Key: select the tag key to be authorized. In this example, we select tag key `test2`.
   - Tag Value: select the tag value to be authorized. In this example, we select tag value `test2`.
   - Resources: the management permission is granted by default.
4. On the verification page, enter the policy name, verify the policy content, and click **Done** to grant sub-account `DevB` management permission for tag key `test2` and tag value `test2`.

Managing new resources

Follow the instructions in Tagging CVM Instances to add tag keys and tag values to manage new resources.