TDSQL for MySQL

Operation Guide

Product Documentation

Contents

Operation Guide
Private IP Conversion

Last updated : 2021-03-02 17:27:09

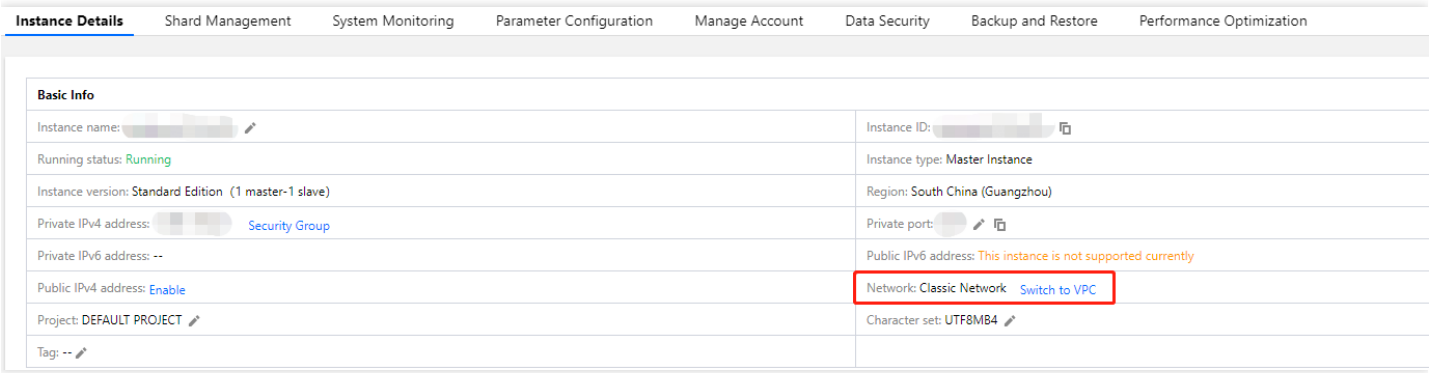
When the access address of a database instance needs to be modified, you can adjust the network using the private network translation feature.

⚠ Note :

- Modifying the private address of an instance is highly risky. Please only do so with caution during off-peak hours. After modification, unless occupied by another service, the original address will remain valid for another 24 hours. Please switch your business configuration as soon as possible.
- Once selected, a VPC cannot be changed.

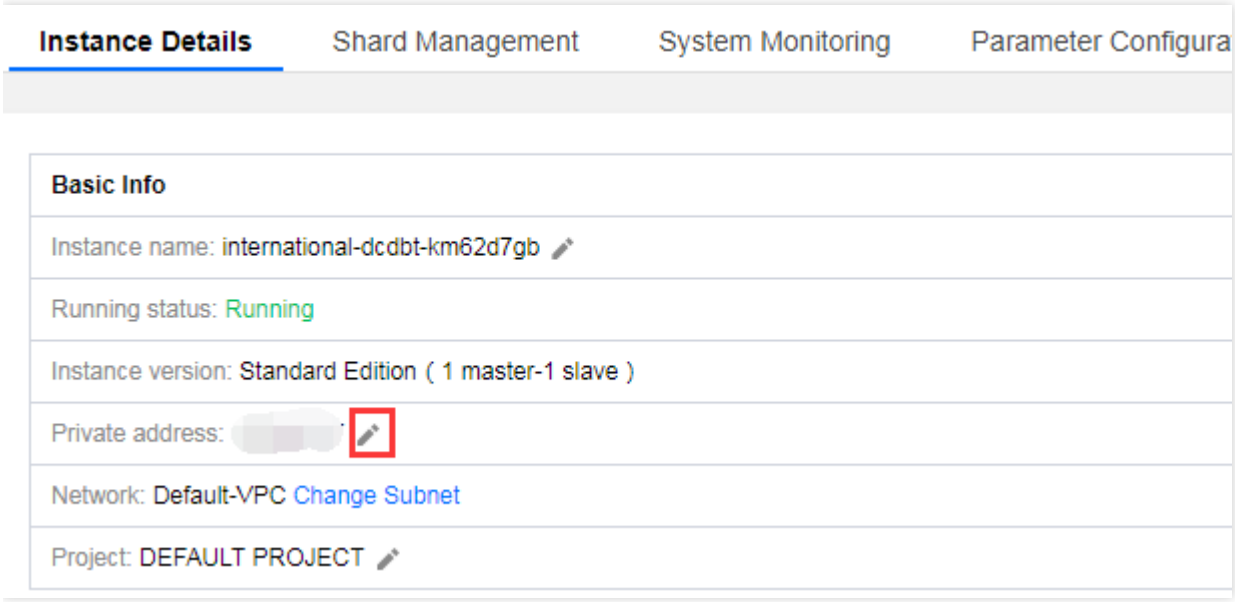
Switching from Basic Network to VPC

An instance can be switched from classic network to VPC. To do so, log in to the TDSQL for MySQL Console, click an instance name in the instance list, and click **Switch to VPC** in the "Network" section on the instance details page (provided that there is an available IP in the target VPC subnet).



Changing Private Address

The private address of a TencentDB instance can be changed. You can do so in the **Private IP** section on the instance details page, provided that there is an available IP in the current subnet.

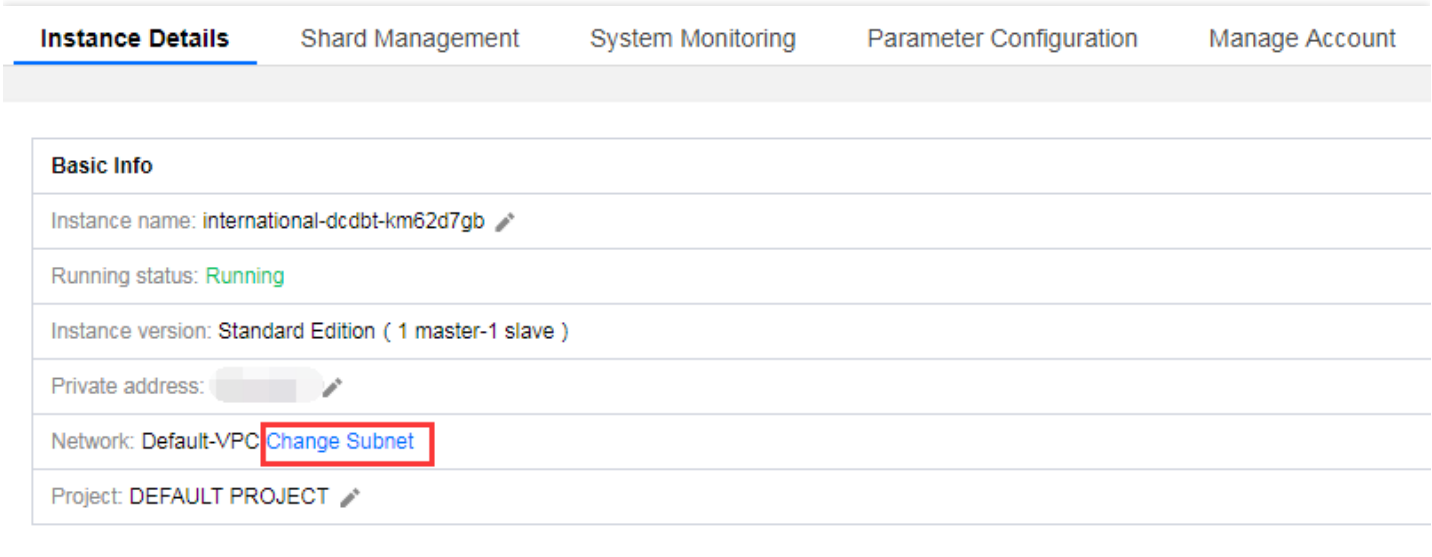


Switching Between VPC Subnets

The VPC subnet of an instance can be changed. To do so, click **Change Subnet** in the **Network** section on the instance details page, provided that there is an available IP in the target VPC subnet.

Note :

Because the product supports an intra-city active-active architecture, you are recommended to choose a VPC subnet in the same region as your business server or master node.



Security Management Access Management Overview

Last updated : 2020-08-26 09:57:52

If you use multiple Tencent Cloud services such as TencentDB, CVM, and VPC that are managed by different users sharing your Tencent Cloud account key, you may face the following problems:

- Your password is shared by multiple users, leading to high risk of compromise.
- You cannot limit the access permission of other users, which is easy to pose a security risk due to faulty operations.

This is exactly why CAM has been developed.
For a detailed description of CAM, see CAM Overview.

After connecting to CAM, you can allow different users to manage different services through sub-accounts so as to avoid the above problems. By default, a sub-account doesn't have permission to use a TencentDB instance or related resources. Therefore, you need to create a policy to grant the required permission to the sub-account.

A policy is a syntax rule used to define and describe one or more permissions. It can authorize or deny the use of the designated resources by a user or user group. For more information on CAM policy, see Policy Syntax. For more information on how to use a CAM policy, see Policy.

If you do not need to manage the access permission to TencentDB resources for sub-accounts, you can skip this chapter. This will not affect your understanding and usage of other parts in the documentation.

Getting Started

A CAM policy must authorize or deny the use of one or more TencentDB 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). A policy can also include the conditions set for the manipulated resources.

⚠ Note :

- You are recommended to manage TencentDB resources and authorize TencentDB operations through CAM policies. Although the experience stays the same for existing users who are

granted permission by project, it is not recommended to continue managing resources and authorizing operations in a project-based manner.
- Effectiveness conditions cannot be set for TencentDB for the time being.

Policy Structure

Last updated : 2020-11-17 10:43:33

Policy Syntax

CAM policy configuration example:

```
{
"version":"2.0",
"statement":
[
{
"effect":"effect",
"action":["action"],
"resource":["resource"],
"condition": {"key":{"value"}}
}
]
}
```

- **version** is required. Currently, only "2.0" is allowed. (This value actually represents the version of TencentCloud APIs acceptable to CAM.)
- **statement** describes the details of one or more permissions. This element contains a permission or permission set of other elements such as effect, action, resource, and condition. One policy has only one statement.
  - **action** describes the allowed or denied action. An action entered here is a string prefixed with "dcdb:" and suffixed with an TDSQL API. This element is required.
  ii. **resource** describes the details of authorization. A resource is described in a six-piece format. Detailed resource definitions vary by product. For more information on how to specify a resource, see the documentation for the product whose resources you are writing a statement for. This element is required.
  iii. **condition** 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. This element is optional.
  iv. **effect** describes whether the result produced by the statement is "allowed" (allow) or "denied" (deny). This element is required.

Actions in TencentDB

In a TencentDB policy statement, you can specify any API action from any service that supports TencentDB. APIs prefixed with "dcdb:" should be used for TencentDB, such as dcdb:CreateDBInstance (creating an instance - monthly subscription) or dcdb:CloseDBExtranetAccess (disabling public network access).

- To specify multiple actions in a single statement, separate them with commas, as shown below:

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

2. You can also specify multiple actions using a wildcard. For example, you can specify all actions whose names begin with "Describe", as shown below:

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

3. If you want to specify all operations in TencentDB, use a wildcard as shown below:

```
"action":["dcdb:*"]
```

TencentDB Resources

Each CAM policy statement has its own resources.
Resources are generally in the following format:

```
qcs:project_id:service_type:region:account:resource
```

- **project_id** describes the project information, which is only used to enable compatibility with legacy CAM logic and can be left empty.
- **service_type** describes the product abbreviation such as DCDB.
- **region** describes the region information, such as ap-guangzhou. For more information, see Regions.
- **account** is the root account of the resource owner, such as "uin/65xxx763".
- **resource** describes detailed resource information of each product, such as instance/instance_id1 or instance/*.

For example:

- You can specify a resource for a specific instance (dcdb-k05xdcta) in a statement as shown below:

```
"resource":[ "qcs::dcdb:ap-guangzhou:uin/65xxx763:instance/dcdb-k05xdcta"]
```

2. You can also use the wildcard "*" to specify it for all instances that belong to a specific account as shown below:

```
"resource":[ "qcs::dcdb:ap-guangzhou:uin/65xxx763:instance/*"]
```

3. If you want to specify all resources or a specific API action does not support resource-level permission control, you can use the wildcard "*" in the "resource" element as shown below:

```
"resource": ["*"]
```

4. To specify multiple resources in a single command, separate them with commas. Below is an example where two resources are specified:

```
"resource":["resource1","resource2"]
```

The table below describes the resources that can be used by TencentDB and the corresponding resource description methods.
In the table, words prefixed with $ are placeholders.

- "region" is region.
- "account" is account ID.

Resource	Resource Description Method in Authorization Policy
Instance	`qcs::dcdb:$region:$account:instance/$instanceId`

Resource-level Permissions Supported

Last updated : 2019-12-06 17:31:14

Resource-level permission can be used to specify which resources a user can manipulate. TencentDB supports certain resource-level permission. This means that for some TencentDB operations, you can control the time when a user is allowed to perform operations (based on mandatory conditions) or to use specified resources. The following table describes the types of resources that can be authorized in TencentDB.

Types of resources that can be authorized in CAM:

Resource Type	Resource Description Method in Authorization Policy

TencentDB instance-related
`qcs::dcdb:$region:$account:instance/*`
`qcs::dcdb:$region:$account:instance/$instanceId`	

The table below lists the TencentDB API operations which currently support resource-level permission control as well as the resources and condition keys supported by each operation. When specifying a resource path, you can use the "*" wildcard in the path.

> Any TencentDB API operation not listed here does not support resource-level permission. If a TencentDB API operation does not support resource-level permission, you can still authorize a user to perform this operation, but you must specify * for the resource element of the policy statement.

The following operations support resource-level permission control

Operation Name	API Name	Effective in Console After Configuration
Recovering a dedicated instance	ActiveDedicatedDBInstance	Yes
Binding security groups	AssociateSecurityGroups	Yes
Checking IP status	CheckIpStatus	Yes
Cloning an account	CloneAccount	Yes

Operation Name	API Name	Effective in Console After Configuration
Disabling public network access for an instance	CloseDBExtranetAccess	Yes
Copying account permission	CopyAccountPrivileges	Yes
Creating an account	CreateAccount	Yes
Creating an instance	CreateDCDBInstance	Yes
Deleting an account	DeleteAccount	Yes
Querying account permission	DescribeAccountPrivileges	Yes
Querying the account list	DescribeAccounts	Yes
Querying audit logs	DescribeAuditLogs	Yes
Querying audit rule details	DescribeAuditRuleDetail	Yes
Querying the audit rule list	DescribeAuditRules	Yes
Querying audit policies	DescribeAuditStrategies	Yes
Querying the price for batch instance renewal	DescribeBatchDCDBRenewalPrice	Yes
Querying instance objects	DescribeDatabaseObjects	Yes
Querying instance database names	DescribeDatabases	Yes
Querying column information of an instance table	DescribeDatabaseTable	Yes
Getting the log list	DescribeDBLogFiles	Yes
Querying monitoring information	DescribeDBMetrics	Yes
Viewing database parameters	DescribeDBParameters	Yes
Querying security group information of an instance	DescribeDBSecurityGroups	Yes
Getting slow log recording details	DescribeDBSlowLogAnalysis	Yes
Getting the slow log list	DescribeDBSlowLogs	Yes

Operation Name	API Name	Effective in Console After Configuration
Querying instance sync mode	DescribeDBSyncMode	Yes
Getting instance details	DescribeDCDBInstanceDetail	Yes
Viewing the instance list	DescribeDCDBInstances	Yes
Querying price	DescribeDCDBPrice	Yes
Querying the renewal price of an instance	DescribeDCDBRenewalPrice	Yes
Querying purchasable AZs	DescribeDCDBSaleInfo	Yes
Querying instance shards	DescribeDCDBShards	Yes
Querying the upgrade price of an instance	DescribeDCDBUpgradePrice	Yes
Querying dedicated cluster specification	DescribeFenceShardSpec	Yes
Querying flow status	DescribeFlow	Yes
Querying the latest DBA check result	DescribeLatestCloudDBAReport	Yes
Viewing backup log settings	DescribeLogFileRetentionPeriod	Yes
Querying order information	DescribeOrders	Yes
Querying projects	DescribeProjects	Yes
Querying security group information of a project	DescribeProjectSecurityGroups	Yes
Querying instance specification	DescribeShardSpec	Yes
Getting SQL logs	DescribeSqlLogs	Yes
Unbinding security groups from Tencent Cloud resources in batches	DisassociateSecurityGroups	Yes
Setting account permission	GrantAccountPrivileges	Yes
Initializing instances	InitDCDBInstances	Yes
Isolating a dedicated instance	IsolateDedicatedDBInstance	Yes
Modifying database account remarks	ModifyAccountDescription	Yes

Operation Name	API Name	Effective in Console After Configuration
Setting auto-renewal in batches	ModifyAutoRenewFlag	Yes
Renaming an instance	ModifyDBInstanceName	Yes
Modifying security groups bound to a TencentDB instance	ModifyDBInstanceSecurityGroups	Yes
Modifying instance project	ModifyDBInstancesProject	Yes
Modifying database parameters	ModifyDBParameters	Yes
Modifying instance sync mode	ModifyDBSyncMode	Yes
Modifying instance network	ModifyInstanceNetwork	Yes
Modifying instance VIP	ModifyInstanceVip	Yes
Modifying backup log settings	ModifyLogFileRetentionPeriod	Yes
Enabling public network access	OpenDBExtranetAccess	Yes
Renewing an instance	RenewDCDBInstance	Yes
Resetting account password	ResetAccountPassword	Yes
Enabling smart DBA	StartSmartDBA	Yes
Scaling an instance	UpgradeDCDBInstance	Yes
Upgrading a dedicated instance	UpgradeDedicatedDCDBInstance	Yes

Console Examples

Last updated : 2021-03-02 15:52:14

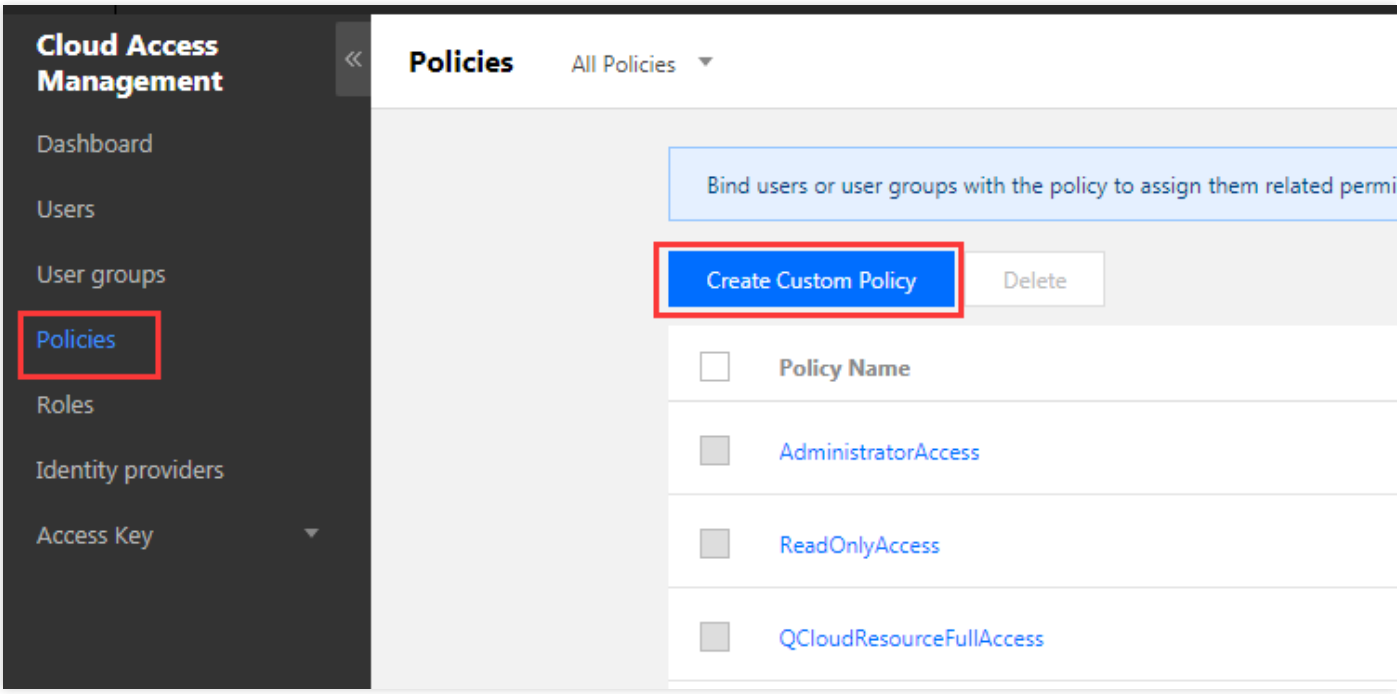
Sample CAM policies for TencentDB

You can grant a user the permission to view and use specific resources in the TencentDB Console by using a CAM policy. The sample below shows how to allow a user to use certain policies in the console.

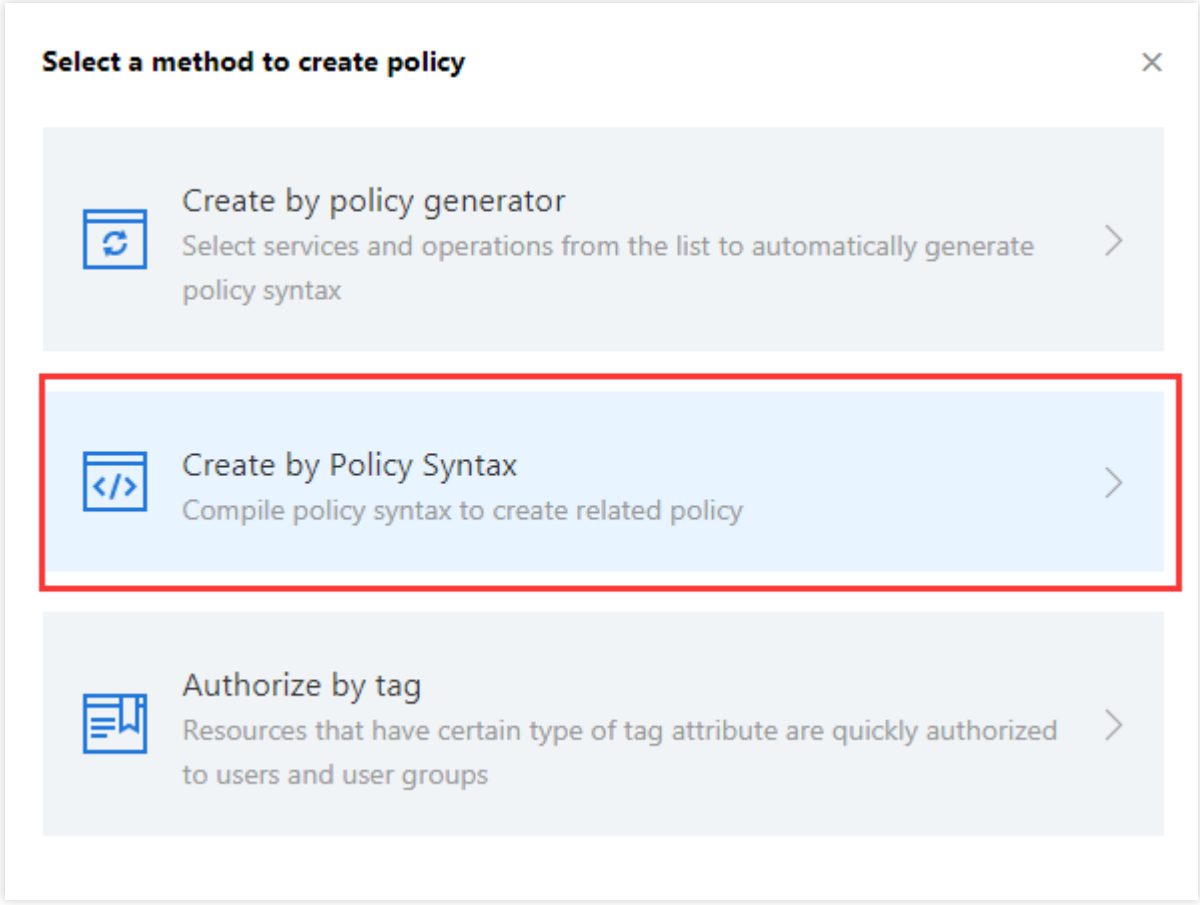
> **Note :**
>
> As TDSQL was formerly known as, its API keyword in CAM is "dcdb".

Syntax for creating a custom policy

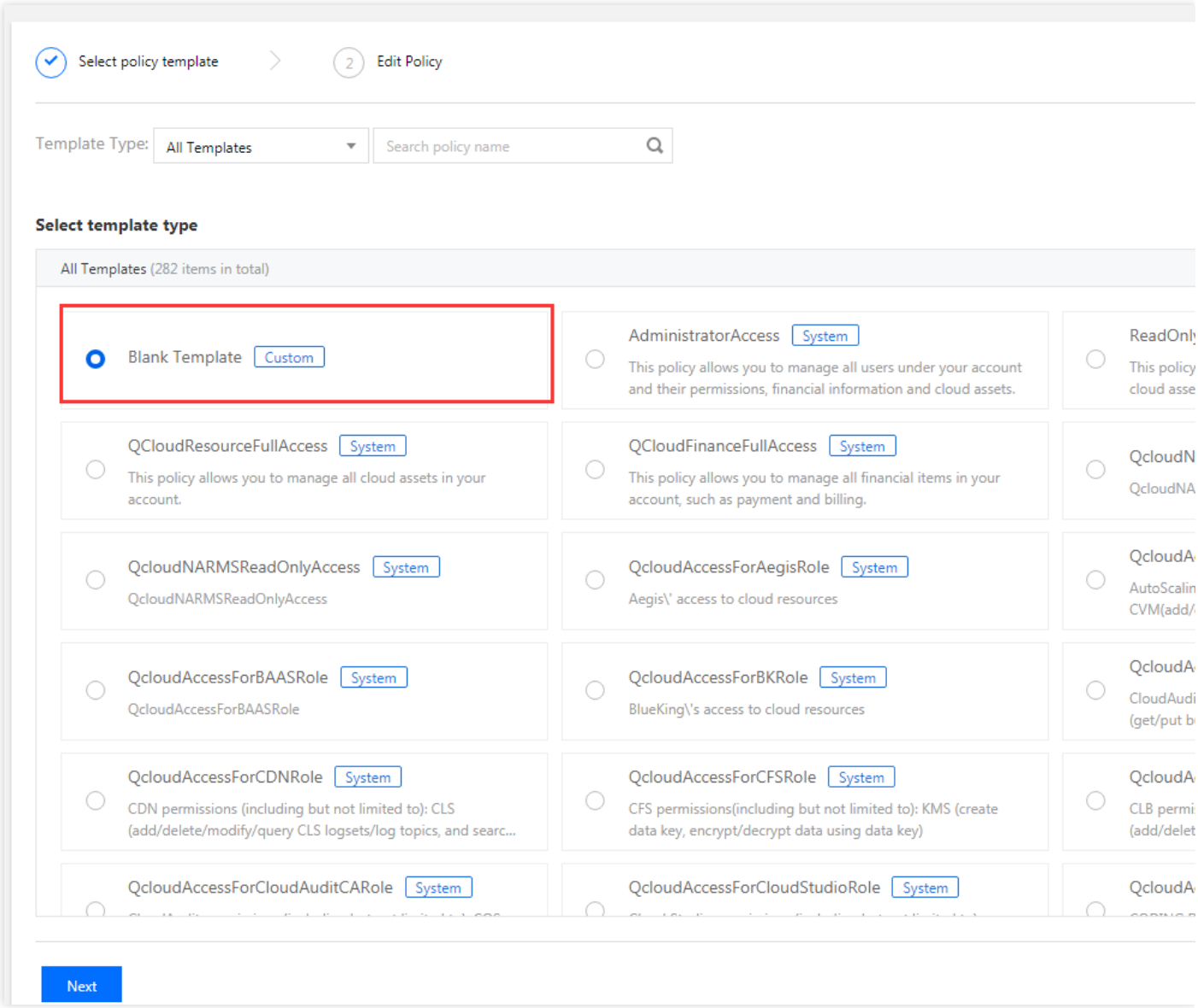
1. Enter the Policy Syntax configuration page and click **Create Custom Policy**.



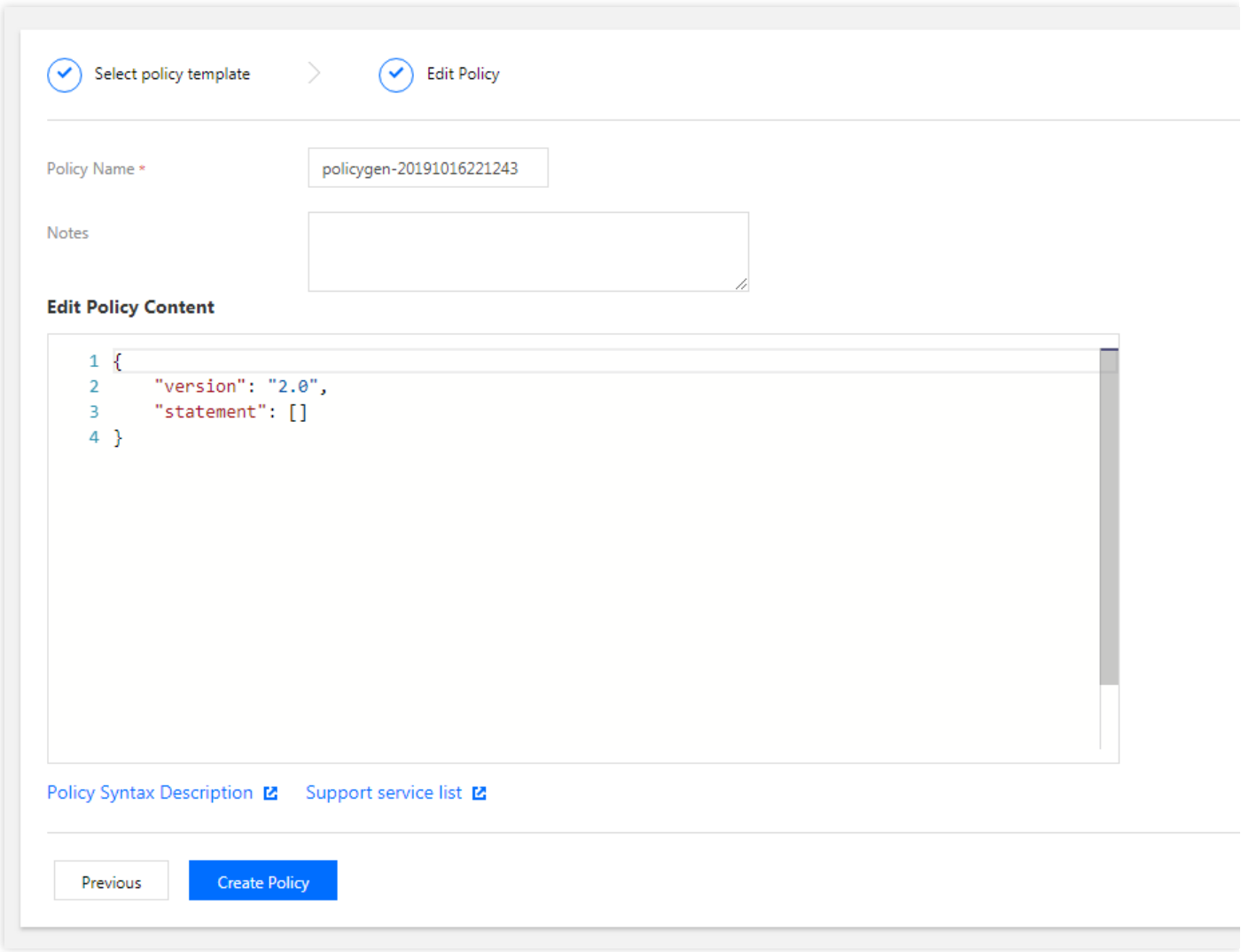
2. Click **Create by Policy Syntax** in the pop-up dialog box.



3. Select "Blank Template" and click **Next**.



4. Enter the corresponding policy syntax.



Associating a sub-account/collaborator and verifying

After the policy is created, associate it with a user/group. After the association is completed, use another browser (or server) to verify whether the sub-account/collaborator can work normally. If the policy syntax is written correctly, you can observe the following:

- You have normal access to the intended target products and resources and can use all the expected features.
- You will be prompted with "You do not have permission to perform this operation" when accessing other unauthorized products or resources.

> To avoid mutual impact of multiple policies, it is recommended to associate only one policy with a sub-account at a time.

The change to account access permission will take effect within 1 minute.

Appendix. Commonly Used Policy Syntax

Policy for authorizing use of all features in all TencentDB instances

To grant a user permission to create and manage TencentDB instances, implement the policy named `QcloudDCDBFullAccess` for the user.

The policy syntax is as follows:

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

Policy for authorizing query of all TencentDB instances

To grant a user permission to view TencentDB instances but not create, delete, or modify them, implement the policy named `QcloudDCDBInnerReadOnlyAccess` for the user.

The policy syntax is as follows:

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

The above policy achieves its goal by allowing the user to separately authorize the use of all operations beginning with "Describe" in TencentDB with the CAM policy.

Note :

As not all functional APIs are covered in the beta test, you may see that a small number of operations are not included in CAM, which is normal.

Policy for granting a user permission to manipulate TencentDB instances in a specific region

To grant a user the permission to manipulate TencentDB instances in a specific region, associate the following policy with the user. For example, the policy below allows the user to manipulate the TencentDB instances in Guangzhou.

```
{
"version": "2.0",
"statement": [
{
"action": "dcdb:*",
"resource": "qcs::dcdb:ap-guangzhou::*",
"effect": "allow"
}
]
}
```

Policy for granting a user permission to manipulate TencentDB instances in multiple specific regions

To grant a user the permission to manipulate TencentDB instances in multiple specific region, associate the following policy with the user. For example, the policy below allows the user to manipulate the TencentDB instances in Guangzhou and Chengdu.

```
{
"version": "2.0",
"statement": [
{
"action": "dcdb:*",
"resource": "qcs::dcdb:ap-guangzhou::*","qcs::dcdb:ap-chengdu::*",
"effect": "allow"
}
]
}
```

Policy for granting a user permission to manipulate a specific TencentDB instance

To grant a user the permission to manipulate a specific database, associate the following policy with the user. For example, the policy below allows the user to manipulate the TencentDB instance "dcdb-xxx" in Guangzhou.

```
{
"version": "2.0",
"statement": [
{
"action": [
"dcdb:*"
],
"resource": "qcs::dcdb:ap-chengdu::instance/dcdb-fwr62n3i",
"effect": "allow"
}
]
}
```

Policy for granting a user permission to manipulate multiple TencentDB instances

To grant a user the permission to manipulate TencentDB instances in batches, associate the following policy with the user. For example, the policy below allows the user to manipulate the TencentDB instances "dcdb-xxx" and "dcdb-yyy" in Guangzhou and "dcdb-zzz" in Beijing.

```
{
"version": "2.0",
"statement": [
{
"action": "dcdb:*",
"resource": ["qcs::dcdb:ap-guangzhou::instance/dcdb-xxx", "qcs::dcdb:ap-guangzhou::instance/dcdb-yyy", "qcs::dcdb:ap-beijing::instance/dcdb-zzz"],
"effect": "allow"
}
]
}
```

Policy for granting a user different permissions to manipulate multiple TencentDB instances

To grant a user the permission to manipulate TencentDB instances in batches, associate the following policy with the user. For example, the policy below allows the user to manipulate the TencentDB

instances "dcdb-xxx" and "dcdb-yyy" in Guangzhou and "dcdb-zzz" in Beijing.

```
{
"version": "2.0",
"statement": [
{
"action": "dcdb:Describe*","dcdb:Create*",
"resource": ["qcs::dcdb:ap-guangzhou::instance/dcdb-xxx", "qcs::dcdb:ap-guangzhou::instance/dcdb-yyy", "qcs::dcdb:ap-beijing::instance/dcdb-zzz"],
"effect": "allow"
}
]
}
```

Note :

For all currently supported APIs, please see the list at the end of this document.

Denying a user permission to create TencentDB accounts

To deny a user permission to create TencentDB accounts, configure `"effect": "deny"` .

```
{
"version": "2.0",
"statement": [
{
"action": "dcdb:CreateAccount",
"resource": "*",
"effect": "deny"
}
]
}
```

Other custom policies

If preset policies cannot meet your requirements, you can create custom policies as shown below:

```
{
"version": "2.0",
"statement": [
{
"action": [
"Action"
],
```

```
"resource": "Resource",
"effect": "Effect"
}
]
}
```

- Replace "Action" with the operation to be allowed or denied.
- Replace "Resource" with the resources that you want to authorize the user to manipulate.
- Replace "Effect" with "Allow" or "Deny".

Supported APIs

Operation Name	API Name	Effective in Console After Configuration
Querying the upgrade price of an instance	DescribeDCDBUpgradePrice	No
Renewing an instance	RenewDCDBInstance	No
Querying the renewal price of an instance	DescribeDCDBRenewalPrice	No
Scaling an instance	UpgradeDCDBInstance	No
Viewing the instance list	DescribeDCDBInstances	Yes
Getting the log list	DescribeDBLogFiles	Yes
Initializing instances	InitDCDBInstances	No
Creating an account	CreateAccount	Yes
Querying the account list	DescribeAccounts	Yes
Deleting an account	DeleteAccount	Yes
Setting account permission	GrantAccountPrivileges	Yes
Querying account permission	DescribeAccountPrivileges	Yes
Copying account permission	CopyAccountPrivileges	No
Modifying database account remarks	ModifyAccountDescription	No
Resetting account password	ResetAccountPassword	Yes

Viewing database parameters	DescribeDBParameters	No
Modifying database parameters	ModifyDBParameters	No
Cloning an account	CloneAccount	Yes
Getting SQL logs	DescribeSqlLogs	No

CAM-enabled Operations

Last updated : 2019-12-06 16:46:52

The following operations support resource-level permission control

Operation Name	API Name	Effective in Console After Configuration
Recovering a dedicated instance	ActiveDedicatedDBInstance	Yes
Binding security groups	AssociateSecurityGroups	Yes
Checking IP status	CheckIpStatus	Yes
Cloning an account	CloneAccount	Yes
Disabling public network access for an instance	CloseDBExtranetAccess	Yes
Copying account permission	CopyAccountPrivileges	Yes
Creating an account	CreateAccount	Yes
Creating an instance	CreateDCDBInstance	Yes
Deleting an account	DeleteAccount	Yes
Querying account permission	DescribeAccountPrivileges	Yes
Querying the account list	DescribeAccounts	Yes
Querying audit logs	DescribeAuditLogs	Yes
Querying audit rule details	DescribeAuditRuleDetail	Yes
Querying the audit rule list	DescribeAuditRules	Yes
Querying audit policies	DescribeAuditStrategies	Yes
Querying the price for batch instance renewal	DescribeBatchDCDBRenewalPrice	Yes
Querying instance objects	DescribeDatabaseObjects	Yes
Querying instance database names	DescribeDatabases	Yes

Operation Name	API Name	Effective in Console After Configuration
Querying column information of an instance table	DescribeDatabaseTable	Yes
Getting the log list	DescribeDBLogFiles	Yes
Querying monitoring information	DescribeDBMetrics	Yes
Viewing database parameters	DescribeDBParameters	Yes
Querying security group information of an instance	DescribeDBSecurityGroups	Yes
Getting slow log recording details	DescribeDBSlowLogAnalysis	Yes
Getting the slow log list	DescribeDBSlowLogs	Yes
Querying instance sync mode	DescribeDBSyncMode	Yes
Getting instance details	DescribeDCDBInstanceDetail	Yes
Viewing the instance list	DescribeDCDBInstances	Yes
Querying price	DescribeDCDBPrice	Yes
Querying the renewal price of an instance	DescribeDCDBRenewalPrice	Yes
Querying purchasable AZs	DescribeDCDBSaleInfo	Yes
Querying instance shards	DescribeDCDBShards	Yes
Querying the upgrade price of an instance	DescribeDCDBUpgradePrice	Yes
Querying dedicated cluster specification	DescribeFenceShardSpec	Yes
Querying flow status	DescribeFlow	Yes
Querying the latest DBA check result	DescribeLatestCloudDBAReport	Yes
Viewing backup log settings	DescribeLogFileRetentionPeriod	Yes
Querying order information	DescribeOrders	Yes
Querying projects	DescribeProjects	Yes

Operation Name	API Name	Effective in Console After Configuration
Querying security group information of a project	DescribeProjectSecurityGroups	Yes
Querying instance specification	DescribeShardSpec	Yes
Getting SQL logs	DescribeSqlLogs	Yes
Unbinding security groups from Tencent Cloud resources in batches	DisassociateSecurityGroups	Yes
Setting account permission	GrantAccountPrivileges	Yes
Initializing instances	InitDCDBInstances	Yes
Isolating a dedicated instance	IsolateDedicatedDBInstance	Yes
Modifying database account remarks	ModifyAccountDescription	Yes
Setting auto-renewal in batches	ModifyAutoRenewFlag	Yes
Renaming an instance	ModifyDBInstanceName	Yes
Modifying security groups bound to a TencentDB instance	ModifyDBInstanceSecurityGroups	Yes
Modifying instance project	ModifyDBInstancesProject	Yes
Modifying database parameters	ModifyDBParameters	Yes
Modifying instance sync mode	ModifyDBSyncMode	Yes
Modifying instance network	ModifyInstanceNetwork	Yes
Modifying instance VIP	ModifyInstanceVip	Yes
Modifying backup log settings	ModifyLogFileRetentionPeriod	Yes
Enabling public network access	OpenDBExtranetAccess	Yes
Renewing an instance	RenewDCDBInstance	Yes
Resetting account password	ResetAccountPassword	Yes
Enabling smart DBA	StartSmartDBA	Yes

Operation Name	API Name	Effective in Console After Configuration
Scaling an instance	UpgradeDCDBInstance	Yes
Upgrading a dedicated instance	UpgradeDedicatedDCDBInstance	Yes

Security Group Configuration

Last updated : 2021-01-04 15:47:46

A security group is a stateful virtual firewall capable of filtering. As an important means for network security isolation provided by Tencent Cloud, it can be used to set network access controls for one or more TencentDB instances. Instances in VPC with the same network security isolation demands in one region can be put into the same security group, which is a logical group (not supported for instances in the classic network currently). TencentDB and CVM share the security group list and are matched with each other within the security group based on rules. Rules not supported by TencentDB will not take effect.

Note :

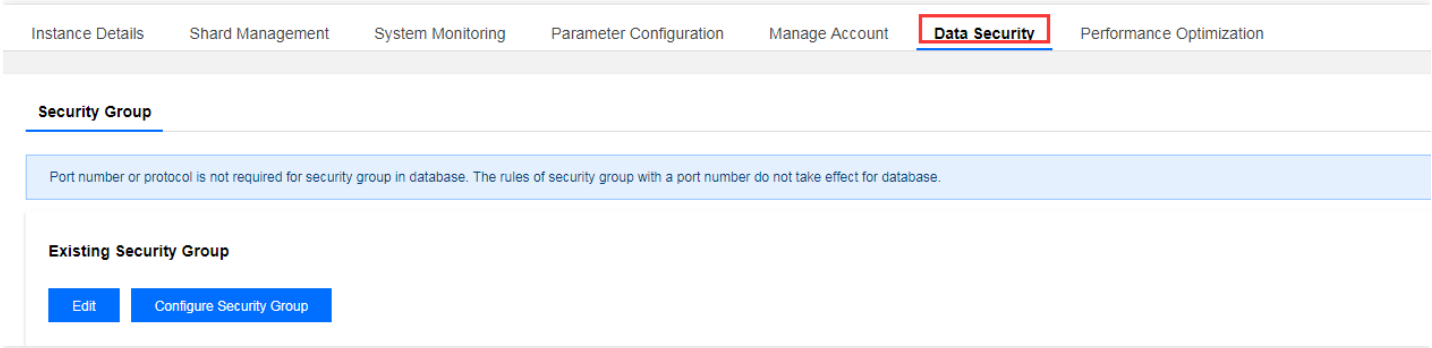
Currently, TencentDB security groups only support network access control for VPC. Neither classic network nor public network is supported.

TencentDB Security Group Management

Log in to the TDSQL console. In the instance list, click an instance name/ID and enter the management page, and select **Data Security** > **Security Group** to manage security groups.

Note :

- TencentDB shares the security group rules of CVM. You can match or adjust the rule priority as needed on the TencentDB security group management page.
- You cannot create or delete security group rules on the security group management page in the TencentDB console.



Security Group Policy

Security group policies are divided into "allowing" and "rejecting" traffic. You can configure security group rules to allow or reject inbound traffic of instances deployed in VPC.

Default Policy of a TencentDB Security Group

Currently, if you select VPC as the network type when purchasing a TencentDB instance, there is no need to associate a security group. In this case, the default policy is to "open all IPs and ports to internet".

Security Group Templates

You can create a custom security group, or create a security group from a template. You can control the inbound and outbound packets of CVMs by configuring security group rules.

Security Group Rules

Security group rules are used to control the inbound and outbound traffic of instances associated with the security group (filtered based on the rules from top to bottom). By default, a new security group rejects all traffic (All Drop). You can modify security group rules at any time, and the new rules take effect immediately.
Each security group rule involves the following items:

- Protocol port: for TencentDB, only **ALL** is supported for protocol port. As TencentDB only provides access over fixed ports, there is no need to specify a port. If a port is specified, the rule will not take effect for TencentDB.
- Authorization type: access based on address ranges (CIDR/IP).
- Source (inbound rules) or target (outbound rules): choose one of the following options:
  - Specify a single IP in CIDR notation.
  - Specify an IP range in CIDR notation.
- Policy: allow or reject the access request.

Security Group Priority

You can set security group priority in the TencentDB console, and the smaller the number, the higher the priority. If an instance is associated with multiple security groups, the priority is used as a basis for evaluating the security rules for this instance.
In addition, if the last policy in multiple security groups associated with an instance is **ALL Traffic Denied**, then the last policy **ALL Traffic Denied** of all security groups except the one with the lowest priority will not take effect.

Security Group Restrictions

- Security groups apply to TencentDB instances in VPC Network Environment.
- Each user can set up to 50 security groups under the same project in the same region.
- A maximum of 100 inbound or outbound rules can be configured for a security group. As TencentDB does not have active outbound traffic, outbound rules are not applicable to TencentDB.
- A TencentDB instance can be associated with multiple security groups, and a security group can be associated with multiple TencentDB instances. No limit is imposed on the number.

> ⚠ **Note :**
>
> We do not recommend associating too many instances with a security group, although no limit is imposed on the number of instances.

Feature	Quantity
Security group	50/region
Access policy	100 (inbound/outbound)
Number of security groups associated with an instance	No limit
Number of instances associated with a security group	No limit

Creating, Managing, and Deleting Security Group Rules

TencentDB shares the security group rules of CVM. You can match or adjust the rule priority as needed on the security group management page in the TencentDB console. To create, manage, or delete security group rules, please do so on the security group management page in the VPC console.

Performance Test

Last updated : 2021-03-02 16:06:14

Overview

Performance test is a comprehensive analysis service for database instance performance and health. It can analyze SQL statement performance, CPU utilization, IOPS utilization, memory utilization, disk utilization, connections, locks, hotspot tables, and transactions, helping you identify and address existing and potential health issues in your database through smart diagnosis and optimization.

This feature is currently available in the following instance editions:

- TDSQL
- TencentDB for MariaDB

> ⚠ **Note :**
>
> For certain test items, the performance test report provides a series of optimization suggestions. Please carefully test the suggested measures before applying them so as to prevent the instance performance problems from getting worse.

Feature Overview

Log in to the Console, click an instance name in the instance list to enter the management page, and select the **Performance Optimization** > **Performance Test** tab where you can perform a performance test.

- **Health rating:** you can view the current database performance score out of 100 points. If your database scores below 60 for a long time, please optimize your business or database configuration.
- **Report generation, viewing, and saving:** you can create a report or view the last report as desired. The report can be saved as a webpage for download.

Main Test Items

Resource analysis

It analyzes the usage of database instance resources (CPU, disk, and connections) in a certain period of time and displays an overall score.

> **Note :**
>
> As most instances adopt the policy of overuse of idle resources by default, you may observe that the maximum CPU utilization exceeds 100%. If this is persistent and the average is higher than the recommended value, you are recommended to scale up your instance.

System status

It sorts out key instance metrics, lists their status and time of occurrence, and suggests corresponding modifications.

Table capacity distribution

It lists the current top 10 tables in reverse order in terms of data capacity to help you identify oversized tables.

Redundant index detection

It lists the current possible redundant indices (with redundancy discrimination below 1%) and suggests optimizations.

> **Note :**
>
> Because a query statement must first query the indices before querying tables through indices, if there are too many identical data entries in the index column, the performance to reduce the amount of data to be filtered may be compromised and is not as fast as full-table scan.

Deadlock diagnosis

Deadlock analysis uses the `show engine innodb status` command to retrieve the last information, which will only be displayed if the deadlock occured within the selected diagnostic period.

> **Note :**
>
> If deadlock occurs frequently, it means that the SQL in the transactions is prone to generate locking loops in concurrent execution scenarios. A fundamental solution is to modify the SQL running logic order and optimize the locking mechanism to reduce the probability of deadlock. A temporary solution is to kill the blocking session leader.

Lock wait diagnosis

It reports lock waits lasting over 60 seconds in the current time period.

Note :

- Lock waits are normal, but sometimes your business may display lock wait timeout errors such as `Lock wait timeout exceeded;try restarting transaction`. MySQL's InnoDB lock information is saved in three tables (`innodb_trx` , `innodb_lock_waits` , and `innodb_locks`) in `information_schema` . Lock wait diagnosis analyzes the lock dependencies in the three tables in the set's master database, finds the session leader that holds a lock for longer than the specified time and blocks other sessions, and then kills it.
- Currently, lock wait is only supported for the InnoDB engine.

Long session diagnosis

It lists the sessions whose command is not "sleep" but execution time exceeds 10 seconds by diagnosing `information_schema.processlist` in the set's master database.

Note :

The best solution to long sessions is to optimize SQL and proactively place session invalidation configuration in your business code. Of course, you can also make expired sessions automatically invalid by adjusting the `interactive_timeout` and `wait_timeout` parameters.

Slow query analysis

It lists the current top 20 slow query statements based on the number of executions in reverse order.

Note :

The slow query threshold can be adjusted in the `long_query_time` parameter. Slow queries may occur for many reasons. Generally, if your instance consumes reasonable amounts of resources but a lot of slow queries occur, you are recommended to check whether your business SQL and indices are appropriate. If your instance has high performance overhead and a lot of slow queries occur, you are recommended to check whether your instance configuration is appropriate and optimize your business SQL and indices. You can query more details of slow queries by using the slow query analysis feature.

Database status check

It checks the health status of the database layer in the current database.

Others

Other values that require DBA's attention are listed.

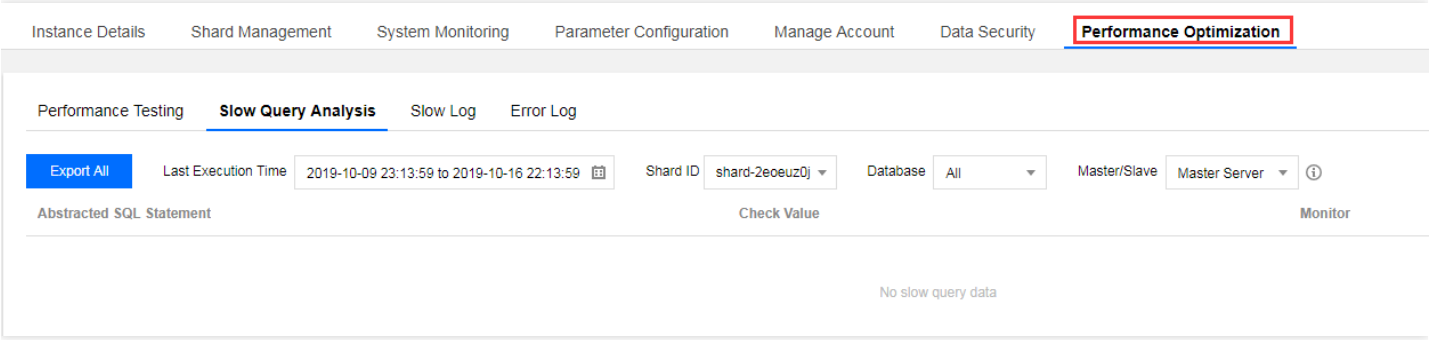
Slow Query Analysis

Last updated : 2021-03-02 17:45:07

Feature Description

An SQL statement query that takes more time than the specified value is referred to as a "slow query", and the corresponding statement is called a "slow query statement". The process where a database administrator (DBA) analyzes slow query statements and finds out the reasons why slow queries occur is known as "slow query analysis".

Log in to the TDSQL for MySQL Console, click an instance name in the instance list to enter the management page, and select the **Performance Optimization** > **Slow Query Analysis** tab where you can perform slow query analysis.



⚠ Note :

Currently, slow query analysis can only be performed and viewed in each shard separately.

Descriptions of main parameters

Main default settings

- Slow query feature: enabled by default.
- Slow query time (long_query_time): 1 second by default, that is, only slow query statements that exceed 1 second will be logged.
- Analyzed data output latency: 1-5 minutes.
- Logging duration: 30 days, depending on the backup and log settings.

Descriptions of analysis list fields

- Checksum (checksum): a sequence of digits used to identify a slow query statement (64-bit by default).
- Abstracted slow query statement (fingerprint): slow query statement with user data hidden.
- Database: the database in which a slow query statement occurs.
- Account: the account under which a slow query statement occurs.
- Last execution time (last_seen): the time when a slow query statement last occurred within the time range.
- First execution time (first_seen): the time when a slow query statement first occurred within the time range.
- Total number of occurrences (ts_cnt): number of occurrences of a slow query statement within the time range.
- Occurrence percentage: occurrence percentage of a slow query statement in relation to all slow query statements within the time range.
- Total time (query_time_sum): total time of a slow query statement within the time range.
- Total time percentage: total time percentage of a slow query statement within the time range.
- Average time (query_time_avg): average time calculated by dividing the total time of a slow query statement with total number of occurrences.
- Minimum time (query_time_min): minimum occurrence time of a slow query statement.
- Maximum time (query_time_max): maximum occurrence time of a slow query statement.
- Total lock time (lock_time_sum): total lock time of a slow query statement.
- Total lock time percentage: time percentage of a slow query statement in relation to total slow query statement lock time within the time range.
- Average lock time (lock_time_avg): average time calculated by dividing total slow query statement lock time with total number of locks.
- Minimum lock time (lock_time_min): minimum slow query statement lock time.
- Maximum lock time (lock_time_max): maximum slow query statement lock time.
- Number of rows sent (Rows_sent_sum): total number of data rows sent by a slow query statement.
- Number of rows scanned (Rows_examined_sum): total number of data rows scanned by a slow query statement.

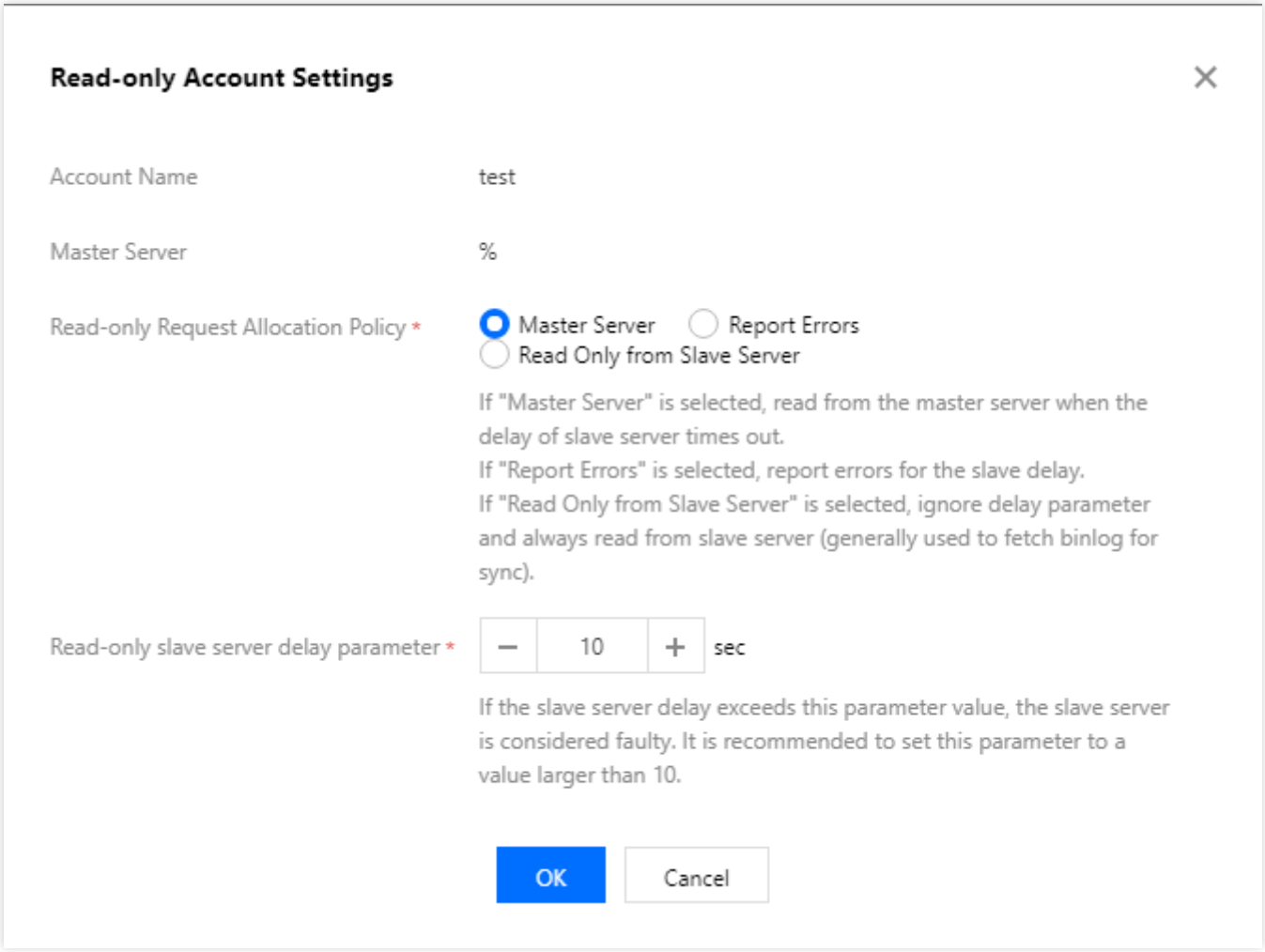
Configuring Read/Write Separation

Last updated : 2021-03-02 17:39:36

Read/Write Separation Based on Read-only Account

1. Log in to the TDSQL for MySQL Console. In the instance list, click an instance name or **Manage** in the "Operation" column to enter the instance management page.
2. Select the **Manage Account** tab and click **Create**.
3. In the pop-up dialog box, set the account information, set **Create as read-only account** to **Yes**, and click **Confirm and Go Next**.
4. In the pop-up dialog box, you can set **Read-Only Request Allocation Policy** to define the read policy when a secondary server failure (or long delay) occurs and configure the " Read-Only Secondary Server Delay Parameter", and then click **OK**.
   - Select **Primary Server** to read from the primary server when the delay of secondary server exceeds the limit.
   - Select **Report Errors** to report an error when the delay of secondary server exceeds the limit.
   - Select **Read Only from Secondary Server** to ignore the delay parameter and always read from the secondary server (this is generally used to pull binlogs for sync).
   - Set the **Read-Only Secondary Server Delay Parameter** to define the data sync delay threshold, which is used together with **Primary Server** and **Report Errors** under the **Read-**

Only Request Allocation Policy.



Read/Write Separation Based on Comment

Add the `/*slave*/` field before each SQL statement to be "read" by the secondary server, and add the `-c` parameter after "mysql" to parse the comment, such as `mysql -c -e "/*slave*/sql"`, to automatically assign "read" requests to the secondary server. Below are examples:

```
//Read from the primary server//
select * from emp order by sal, deptno desc;
//Read from the secondary server//
/*slave*/ select * from emp order by sal, deptno desc;
```

⚠ Note :

- This feature only supports read from the secondary server (SELECT) rather than other operations. Non-SELECT statements will fail.
- The `-c` parameter needs to be added after `mysql` to parse the comment.
- `/*slave*/` must be in lowercase, and no spaces are needed before and after the statement.
- If the MAR (strong sync) mechanism is affected by a secondary server exception, read from the secondary server will be automatically switched to read from the primary server.

Isolating, Restoring, and Terminating Instance

Last updated : 2021-03-02 17:42:20

Isolating Instance

Once isolated, an instance cannot be used or accessed; however, it is not terminated or deleted. You can recover it in the console. After isolation, resource capacity will not be released, and the most basic data replicas will be retained. After the isolation period elapses, the instance will be completely terminated.

- A pay-as-you-go instance can be returned manually in the console. After the instance is returned, it will be in "isolated" status and retained for 24 hours, during which it cannot be accessed. If you want to restore it, you can do so in the instance list.

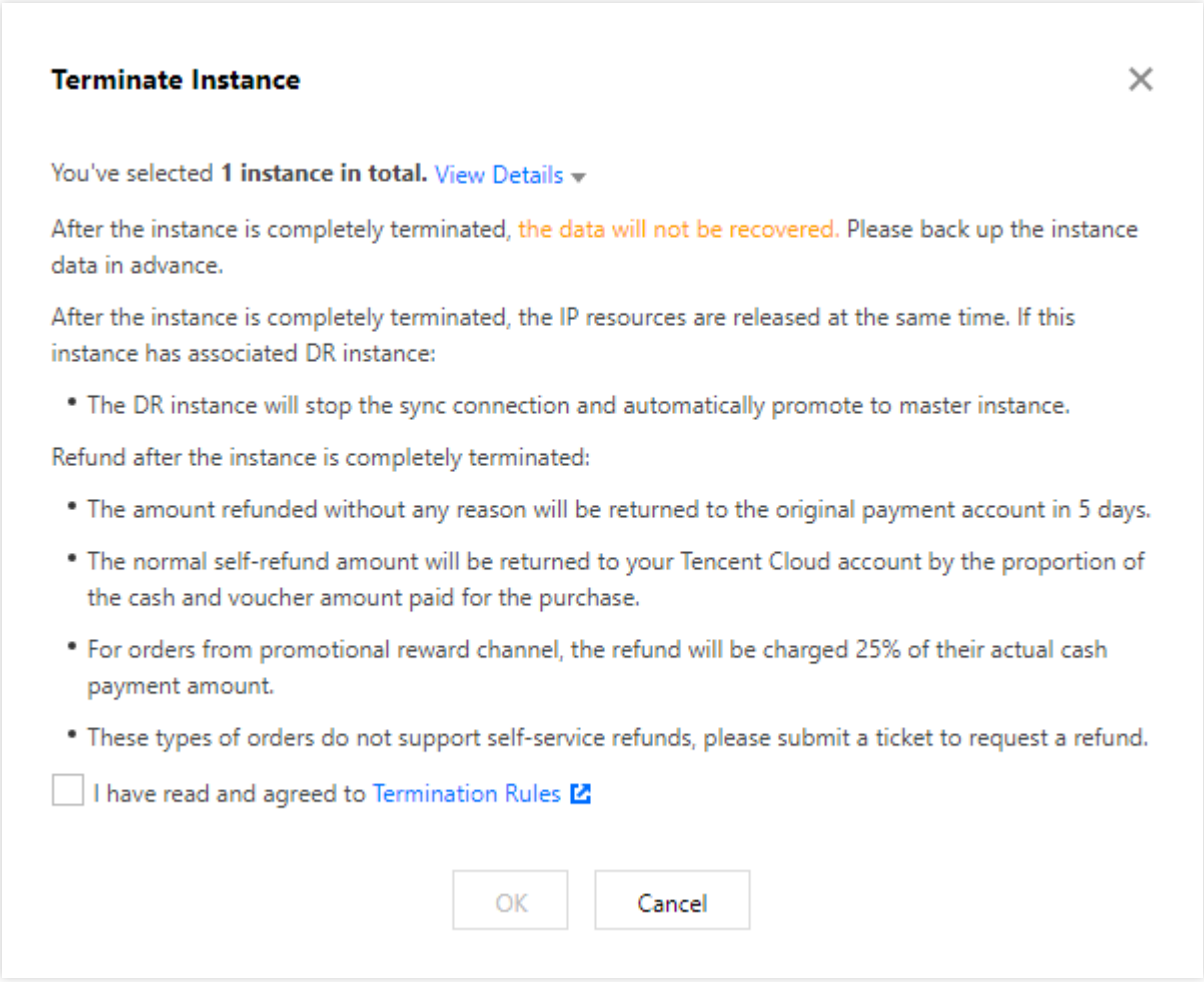
After an instance is returned, once its status changes to "isolated", no fees related to it will be incurred.

> ⚠ **Note :**
>
> - After an instance is isolated, its IP will be released, and you may not get back the original IP after the instance is recovered.
> - After an instance is isolated, you cannot upgrade it, modify its parameters, create or modify an account for it, roll it back, add sets to it, or rename it.

Directions

1. Log in to the TDSQL for MySQL Console, select an instance and click **More** > **Terminate/Return** above the instance list.

2. In the pop-up dialog box, indicate your consent and click **OK**



3. Return to the instance list. The status of the instance has changed to **isolated**, and you can choose to **restore/start up** the instance.

Recovering Instance

Instance recovery is to recover an isolated instance to its normal running status. The recovery may take several minutes to complete. In addition, the recovered instance may have a new IP rather than the original IP before isolation.

Terminating Instance

If you do not need an instance any longer, you can return it, and it will be isolated. An isolated instance will be completely terminated after expiration.

Notes

- After an instance is terminated completely, its data will not be recoverable. Please back up the data in advance.
- When the instance is terminated, its IP resources will be released simultaneously. If it has a disaster recovery instance, the disaster recovery instance will stop the sync connection and automatically promote to master instance.

Back up
Backup Mode

Last updated : 2021-03-02 17:28:52

TDSQL for MySQL supports full backups and incremental backups.

Backup Type

Full backup

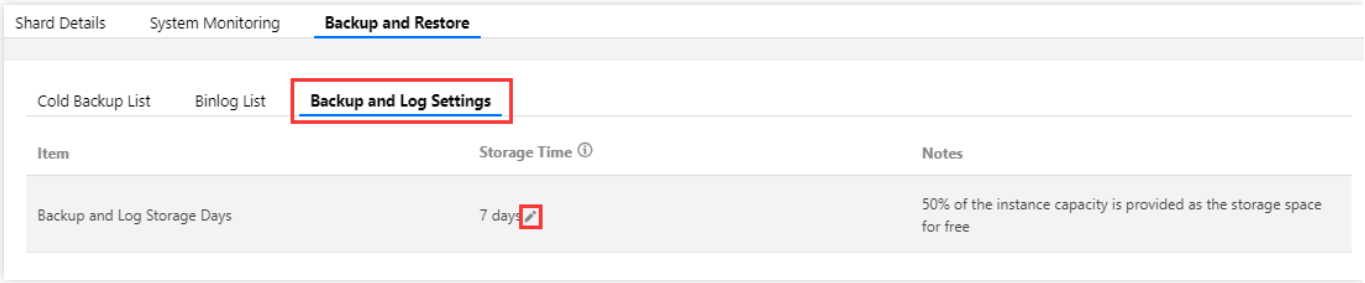
You can set the start time and retention period for full backups. By default, the backup starts at 00:00–05:00 AM, during which the performance is relatively low. The retention period is 7 days by default.

Incremental backup

Incremental backup is implemented based on binlogs, which are generated in real time. The binlogs use a certain amount of disk capacity, and are periodically uploaded to the TencentDB backup system.

Custom Backup Time

1. Log in to the TDSQL for MySQL Console and click the instance name or **Manage** in the "Operation" column to enter the instance management page.
2. Select **Shard Management** and click the shard ID to enter the shard management page.
3. Select **Backup and Restore** > **Backup and Log Settings** and click the icon as shown below to set the storage period.
   - Backup cycle: the backup task is performed every day by default.
   - Storage time: data and log backups can be retained for 1 to 7 days. Retention time is set to 7 days by default.



Downloading Backup File

Last updated : 2021-01-11 15:05:16

Overview

You can download TencentDB cold backup data and binlogs in the TDSQL for MySQL Console.

Directions

1. Log in to the TDSQL Console and click the instance name or **Manage** in the "Operation" column to enter the instance management page.
2. Select **Shard Management** and click the shard ID to enter the shard management page.
3. Go to the **Backup and Restore** tab and select **Cold Backup List** or **Binlog List** and click **Download** in the "Operation" column.
4. A VPC address is provided in the download dialog box that pops up. You can click **Copy** to get the address.

   - The address is valid for 15 minutes. After it expires, refresh the page to get a new one. Access the VPC address in the VPC network.
   - We recommend that you copy the download address, log in to a (Linux) CVM instance in the same VPC as the TencentDB instance, and run the `wget` command for download.