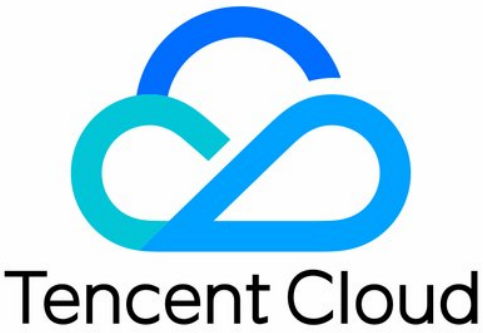
TencentDB for TcaplusDB

Getting Started

Product Documentation



©2013-2022 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Getting Started
- Basic Concepts
  - Cluster
  - Table Group
  - Table
  - Index
  - Data Types
  - Read/Write Capacity Mode
  - Table Definition in ProtoBuf
  - Table Definition in TDR
- Creating Cluster
- Creating Table Group
- Creating Table
- Get Connection Information
- Accessing TcaplusDB

©2013-2022 Tencent Cloud. All rights reserved.

Getting Started
Basic Concepts
Cluster

Last updated：2021-12-29 18:03:51

Cluster Overview

A cluster is the basic TcaplusDB management unit, which provides independent TcaplusDB service for business. From the perspective of gaming business, a cluster can provide service for either a game's independent submodules or the entire game.

Creating Cluster

For detailed directions, please see Creating Cluster.

Cluster Details

You can view the cluster configuration and attributes details by logging in to the TcaplusDB Console and then clicking the ID of the target cluster.

Cluster details consist of three parts:

Basic information

- Cluster ID: unique cluster ID, which can be used to locate the cluster and control access to the cluster but cannot be used for database connection.
- Access ID: cluster connection ID, which identifies the cluster when you access the database through the SDK or client tool.
- Connection protocol: table definition protocols supported by cluster. Currently, only Protocol Buffers is supported.
- Cluster name: cluster name, which can be customized as needed.
- Region: cluster region.
- RESTful API : address used for database access through a RESTful API in the same VPC subnet.

Network information

- Network: information of the VPC where the current cluster resides.

©2013-2022 Tencent Cloud. All rights reserved.

- Subnet: information of the subnet where the current cluster resides.
- Private address: private IP address assigned to the current cluster. CVM instances in the same VPC subnet as the cluster can access this IP address.
- Private port: access port number assigned to the current cluster.

Other information

- Creation time: creation time of the current cluster.
- Connection password: if you forgot the password, you can view the cluster access password in the console.

©2013-2022 Tencent Cloud. All rights reserved.

Table Group

Last updated：2020-07-31 11:16:00

Table Group Overview

As a logic isolation method in TcaplusDB, a table group represents a data partition that can separate different tables from each other. A cluster can contain multiple table groups, and a table group can contain multiple tables. Different table groups cannot access data of each other.

From the perspective of gaming business, if a gaming business supports unified servers in all zones and can read/write the same table group in the cluster, then it does not need to be divided into different zones. It can also be server-specific/zone-specific, i.e., the cluster can contain multiple table groups.

Creating Table Group

For detailed directions, please see Creating Table Group.

Table Group Details

You can view the table group configuration and attributes on the cluster list page to understand the overall cluster usage.
You can log in to the TcaplusDB Console and enter the cluster list to view the table group information of the target cluster.

Table group information contains the following four fields:

- ID: it is the ID of the table group in the current cluster, which is required during database connection. Please note that the table group IDs may be repeated in different clusters.
- Table Group Name: you can customize the name of the table group based on its actual purpose.
- Table Count: it indicates the number of tables in the current table group.
- Total Capacity: it indicates the disk capacity used by the tables in the current table group.

©2013-2022 Tencent Cloud. All rights reserved.

Table

Last updated：2021-12-29 17:26:55

Table Overview

Like in other databases, a table in TcaplusDB is a data set as a combination of business data stored based on the table definition.
From the perspective of gaming business, a table can manage a set of correlated data of a business module such as user table and item table.

For example, the player table below can store the information of all players in a game such as game character name, gender, race, level, strength, gears, mount, and items.

```
{
player_id:11474,
player_name: "Test account 2",
gender:0,
ethnicity:"Elf",
FightingPower:10,
equipment:
{
helmet:0,
Warframe:0,
gloves:0,
necklace:0,
pants:0,
Shoes:0
},
horse:"0"
},
{
player_id:11475,
player_name: "Test account 1",
gender:1,
ethnicity:"Orc",
FightingPower:1477,
equipment:
{
helmet:1478,
Warframe:21,
gloves:554,
necklace:12,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
pants:64,
Shoes:122
},
horse:"3"
}
```

Record

A record is a set of field values of the same object. In the sample player character table above, all information of a player character consists of multiple attribute field values, which together represent the set of the player's character information in the game. A record can contain up to 1 MB data.

Field

A field describes an attribute of a specified object. In the sample player character table above, an independent attribute of a player character is called a field, such as `player_name` and `ethnicity` .

Primary key field

As can be found from the sample table above, each record has a field that marks its uniqueness. This field is called primary key field. In the player character table above, the primary key fields are `player_id` and `player_name` . TcaplusDB supports up to 4 primary key fields.

General field

In addition to the fields marking the record uniqueness, each record also has other attributes called general fields. In the sample player character table above, the general fields include `equipment` and `FightingPower` . TcaplusDB supports up to 128 general fields.

Sharding field

TcaplusDB is a distributed database system that can provide higher concurrent request capability for tables. If you set a sharding field, TcaplusDB will calculate its hash value to automatically adjust the table shards based on the current table access conditions; otherwise, the system will use a primary key field as the sharding field.

The setting of the sharding field is subject to the backend data distribution. You need to evaluate whether the values of the field used as the sharding field are discrete. You are not recommended to set a field with limited values such as gender and weekday as the sharding field; instead, you are recommended to use fields such as ID and name.

Field type

TcaplusDB supports multiple data types as detailed below:

Field Type	Description

©2013-2022 Tencent Cloud. All rights reserved.

Field Type	Description
int32	Variable-length signed integer that can represent a value between -2^31 and +2^31. It is relatively inefficient to represent a negative number.
int64	Variable-length signed integer that can represent a value between -2^63 and +2^63.
uint32	Variable-length unsigned integer that can represent a value between 0 and 2^32.
uint64	Variable-length unsigned integer that can represent a value between 0 and 2^64.
sint32	Variable-length signed `int` value which is more efficient to represent a negative number than regular `int32` .
sint64	Variable-length signed `int` value which is more efficient to represent a negative number than regular `int64` .
bool	Boolean value representing `True` or `False` .
fixed32	Fixed-length numeric type always taking 4 bytes. It is more efficient than `uint32` if the value is generally above 2^28.
fixed64	Fixed-length numeric type always taking 8 bytes. It is more efficient than `uint64` if the value is generally above 2^56.
sfixed32	Fixed-length numeric type always taking 4 bytes.
sfixed64	Fixed-length numeric type always taking 8 bytes.
float	32-bit binary floating point taking 4 bytes.
double	64-bit binary double-precision floating point taking 8 bytes.
strings	String encoded with UTF-8 or 7-bit ASCII that cannot exceed 2^32 in length.
bytes	Byte sequence of a length below 2^32.
message	Nested type. It supports up to 128 consecutive nesting levels. A high number of nesting levels will affect the performance.

Table definition format

TcaplusDB supports Protocol Buffers, which is a lightweight and efficient structured data storage format and mainly used to store structured data sequentially.

For more information on the table formats, please see Table Description Language.

©2013-2022 Tencent Cloud. All rights reserved.

Index

Last updated：2021-12-29 17:27:47

Index Overview

You can create an index for a table and use it to quickly get desired data from a database, which delivers higher flexibility in data query for your application.

TcaplusDB supports two types of indexes:

- Local index: it must be composed of primary key fields and allows you to quickly query data by using a primary key as a filter.
- Global index: it can be composed of any fields except those in `message` type and allows you to quickly query data by using a field configured in the global index as a filter.

TcaplusDB supports up to 1 global index and 6 local indexes.

Local index

A local index can be defined only during table creation and can consist of only primary key fields. The intersection of all index field sets cannot be empty.
For example, the `HeroInfo` table below has 4 primary key fields: `heroId` , `heroName` , `heroFightingType` , and `heroQuality` .

```
{
heroId:1,
heroName:"Arthur",
heroFightingType:1,
heroQuality:3
heroSkill:{
BasicSkill1:1,
BasicSkill2:2,
SpecialSkill:3
},
heroLevel:12,
heroskin:2,
heroAttackpower:141,
heroPhysicalDefense: 283,
heroMagicdefense:124
}
{
heroId:4,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
heroName:"Shooter",
heroFightingType:3,
heroQuality:4
heroSkill:{
BasicSkill1:1,
BasicSkill2:2,
SpecialSkill:3
},
heroLevel:11,
heroskin:1,
heroAttackpower:225,
heroPhysicalDefense: 57,
heroMagicdefense:41
}
```

For the table above, you can use any combination of the 4 primary key fields to create indexes, but the intersection of all created indexes cannot be empty. For example, if `heroId` is used as the intersection field for index designing, the following indexes can be created:

- heroId,heroName
- heroId,heroFightingType
- heroId,heroQuality
- heroId,heroName,heroFightingType
- heroId,heroFightingType,heroQuality
- heroId,heroName,heroFightingType,heroQuality

  You can query data by the fields in the created indexes. Please create indexes based on your actual business needs; otherwise, the performance will be compromised due to index redundancy.

Global index

In the `HeroInfo` table above, you can query data entries by using a field in a created local index as a filter. For example, if you want to query data by information such as `heroLevel` and `heroskin`, you can add them to the global index and use the fields in it to query data.

Please note that the global index does not support nested types either, such as the `heroSkill` field in the sample table above.

©2013-2022 Tencent Cloud. All rights reserved.

Data Types

Last updated：2023-12-20 16:40:36

TencentDB for TcaplusDB supports Protocol Buffers data types.

Mapping between proto3 and Data Types in Programing Languages

.proto	C++	Java	Python	Go	Ruby	C#	PHP	Dart
double	double	double	float	float64	float	double	float	doubl
float	float	float	float	float32	float	float	float	doubl
int32	int32	int	int	int32	fixnum/ bignum (as required)	int	integer	int
int64	int64	long	int/ long	int64	bignum	long	integer/string	Int64

©2013-2022 Tencent Cloud. All rights reserved.

uint32	uint32	int	int/ long	uint32	fixnum/ bignum (as required)	uint	integer	int
uint64	uint64	long	int/ long	uint64	bignum	ulong	integer/string	Int64
sint32	int32	int	int	int32	fixnum/ bignum (as required)	int	integer	int
sint64	int64	long	int/ long	int64	bignum	long	integer/string	Int64
fixed32	uint32	int	int/ long	uint32	fixnum/ bignum (as required)	uint	integer	int

©2013-2022 Tencent Cloud. All rights reserved.

fixed64	uint64	long	int/ long	uint64	bignum	ulong	integer/string	Int64
sfixed32	int32	int	int	int32	fixnum/ bignum (as required)	int	integer	int
sfixed64	int64	long	int/ long	int64	bignum	long	integer/string	Int64
bool	bool	boolean	bool	bool	trueClass/ falseClass	bool	boolean	bool
string	string	String	str/ unicode	string	string (UTF -8)	string	string	string

©2013-2022 Tencent Cloud. All rights reserved.

bytes	string	ByteString	str	byte	string (ASCII-8BIT)	bytestring	string	-

Mapping between proto2 and Data Types in Programing Languages

.proto	C++	Java	Python	Go	Notes
double	double	double	float	*float64	-
float	float	float	float	*float32	-
int32	int32	int	int	*int32	Uses variable-length encoding. Inefficient for encoding negative numbers. If your field is likely to have negative values, use sint32 instead.
int64	int64	long	int/long	*int64	Uses variable-length encoding. Inefficient for encoding negative numbers. If your field is likely to have negative values, use sint64 instead.
uint32	uint32	int	int/long	*uint32	Uses variable-length encoding.
uint64	uint64	long	int/long	*uint64	Uses variable-length encoding.
sint32	int32	int	int	*int32	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.
sint64	int64	long	int/long	*int64	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.
fixed32	uint32	int	int/long	*uint32	Always four bytes. More efficient than uint32 if values are generally greater than

©2013-2022 Tencent Cloud. All rights reserved.

					2^28.
fixed64	uint64	long	int/long	*uint64	Always eight bytes. More efficient than uint64 if values are generally greater than 2^56.
sfixed32	int32	int	int	*int32	Always four bytes.
sfixed64	int64	long	int/long	*int64	Always eight bytes.
bool	bool	boolean	bool	*bool	-
string	string	String	unicode(Python 2) or str(Python 3)	*string	A string must always contain UTF-8 encoded or 7-bit ASCII text.
bytes	string	ByteString	bytes	byte	May contain any arbitrary sequence of bytes.

©2013-2022 Tencent Cloud. All rights reserved.

Read/Write Capacity Mode

Last updated：2020-07-31 11:16:00

Read/Write Capacity Mode Overview

TcaplusDB uses the read/write capacity mode to calculate the volume of table read/write requests.

Capacity unit (CU)

It is the minimum request unit of resource access and the minimum statistical unit of requested data reads/writes.

Read capacity unit (RCU)

It is the minimum unit of read requests. The response data generated for a request may contain multiple RCUs. The maximum value of an RCU is 4 KB, i.e., an RCU indicates that a read request can be performed on a record that is up to 4 KB in size. If you need to read data of more than 4 KB, you need more RCUs.

The base number of an RCU is 4 KB. The size of a request below 4 KB is calculated as 4 KB, and the size of a request above 4 KB is calculated as a multiple of 4 KB (the remaining part below 4 KB is calculated as 4 KB).
Below is an example:

Read request capacity

You send a request to query a field in a record which is 10 KB in size, and the database returns the value of the specified field of the record. In this case, the response data is 10 KB, which is 3 RCUs.

Write capacity unit (WCU)

It is the minimum unit of write requests. The written data generated for a write request may contain multiple WCUs. The maximum value of a WCU is 4 KB, i.e., every 4 KB of written data will be counted as one WCU.
The base number of a WCU is 4 KB. The size of a request below 4 KB is calculated as 4 KB, and the size of a request above 4 KB is calculated as a multiple of 4 KB (the remaining part below 4 KB is calculated as 4 KB).
You can select whether to enable the result flag feature when performing a write operation. If it is enabled, the system will automatically calculate the returned CUs as WCUs.
Consumed number of WCUs = CUs generated by data write + returned CUs

Below is an example:

- **Write request**
  You send a request to update content of 5 KB, and the backend reads the record (50 KB in total) containing the target data from the disk into the memory.
  Then, the backend updates the record and flushes the 50 KB content into the disk after the update.
  If you have enabled result flag, 5 KB content will be returned.

©2013-2022 Tencent Cloud. All rights reserved.

- **Result flag enabled**
  13 CUs (50 KB data written into disk) + 2 CUs (5 KB returned data) = 15 WCUs generated
- **Result flag not enabled**
  13 CUs (50 KB data written into disk) = 13 WCUs generated

Preset mode

If you select the preset mode, you need to adjust the preset read/write capacity of the table based on access change. This operation can help you control the TcaplusDB use to keep it at or below the defined request rate, so that you can predict the costs.
To better configure the preset capacity, you need to estimate the application traffic in advance.

Reserved read

It refers to the RCU preconfiguration of a table. You need to estimate the highest RCU consumption per second of the table on the day and set the value as the reserved RCUs.

Reserved write

It refers to the WCU preconfiguration of a table. You need to estimate the highest WCU consumption per second of the table on the day and set the value as the reserved WCUs.

Reserved capacity

It refers to the preconfiguration of disk capacity usage of a table. You need to estimate the capacity used by the table on the day.

Request limit and quota

As your configured RCU or WCU value may be too low or the number of access requests may surge, TcaplusDB limits the requests exceeding 40% of the reserved RCUs or WCUs by default. This limit does not block the requests; instead, it makes the requests queue up to reduce the access frequency, which may cause timeout in the program. You can use the monitoring and alarming feature to know whether the current number of requests exceeds the preconfigured capacity.

This limit can prevent your application from consuming too many CUs. However, if requests are limited, they will fail, and many system errors will occur.

You can use TcaplusDB table monitoring to monitor the actual RCUs/WCUs and modify the table quota if necessary.

©2013-2022 Tencent Cloud. All rights reserved.

Table Definition in ProtoBuf

Last updated：2022-02-11 15:04:39

TencentDB for TcaplusDB supports two table definition languages: Protocol Buffers (Protobuf) and Tencent Data Representation (TDR). Protobuf is a method of serializing structured data developed by Google, which emphasizes simplicity and performance. TDR is a platform-neutral data description language developed by Tencent, which combines the advantages of XML, binary language, and object-relational mapping (ORM) and is widely used in data serialization scenarios of Tencent's games. Both languages are equally useful. Use either of them based on your usage habits. This document describes how to define tables in Protobuf.

Table Definition in Protobuf

To create a table in TcaplusDB, you first need to use a table description language to define the table format and write the table definition content into a table IDL description file.
The table IDL description file defines the table based on Protocol Buffers rules.
The definition information is mainly divided into three types:

- File definition information
- Table definition information
- Nested type information

File Definition

The file definition information area mainly defines the common information of the current table description file, which contains the following three types of contents:

Option Name	Description	Sample Value	Required
syntax	It specifies the syntax specification version for writing the current file.	proto2, proto3	Yes
package	It specifies the custom package name of the current file, which helps avoid name conflicts with messages.	Package name information	Yes
importIt	It indicates some common information imported into the TcaplusDB table, which must be imported in the table definition.	tcaplusservice.optionv1.proto	Yes

©2013-2022 Tencent Cloud. All rights reserved.

Table Definition

In table definition information, the table format (including extended information and field information) is defined in a message.

Extended information definition

Table definition information can be extended by `option` on the basis of Protobuf syntax, which can implement richer semantic features. The definable content is as shown below:
The detailed definition format is `option(tcaplusservice.option) = "value";` .

Option Name	Description	Configuration Example	Required
tcaplus_primary_key	Sets the primary key field of the TcaplusDB table.	option(tcaplusservice.tcaplus_primary_key) = "uin,name,region";	Yes
tcaplus_index	Sets the index key field of the TcaplusDB table.	option(tcaplusservice.tcaplus_index) = "index_1(uin,region)";	No
tcaplus_sharding_key	You can customize the table shardkey.	option(tcaplusservice.tcaplus_sharding_key) = "uin";	No
tcaplus_field_cipher_suite	It is required if the field encryption feature is used. To specify your custom encryption algorithm, see the example in the API documentation.	option(tcaplusservice.tcaplus_field_cipher_suite) = "DefaultAesCipherSuite";	No

©2013-2022 Tencent Cloud. All rights reserved.

Option Name	Description	Configuration Example	Required
tcaplus_cipher_md5	MD5 (used to encrypt the password string, which is stored at the user side) should be set if the field encryption feature is used.	option(tcaplusservice.tcaplus_cipher_md5)= "62fee3b53619b7f303c939964c6f2c4b";	No

Field information definition

TcaplusDB field definition format is `field modifier field type field name = identifier[special definition];` .

Field modifier

proto2 supports three limiting modifiers, while proto3 no longer supports the `REQUIRED` modifier and uses `OPTIONAL` as the default modifier.

- REQUIRED: it indicates that this field is required. In proto2, primary key fields must be declared with `REQUIRED` .
- OPTIONAL: it indicates that this field is optional. You can set a default value for an optional field.
- REPEATED: it indicates that this field can contain 0–N elements. Its features are the same as those of `OPTIONAL` , but it can contain multiple values at a time, which can be considered as an array. The special definition of `[packed = true]` must be specified.

Field type

TcaplusDB supports general fields and nested fields. For more information, please see Data Types.

Field name

You can name a field based on its current attribute. The name can contain letters, digits, and underscores and cannot start with a digit. You are recommended to use camelCase for the name.

Identifier

Identifier value range: [1, 2^29 - 1]
Identifiers within [19000, 19999] cannot be used, as they are reserved in Protocol Buffers. If you use any of them, an error will be reported.
Each field will occupy memory when being encoded, and the occupied memory size is subject to the identifier:

- A field with an identifier within [1, 15] occupies 1 byte during encoding.

©2013-2022 Tencent Cloud. All rights reserved.

- A field with an identifier within [16, 2047] occupies 2 bytes during encoding.

Special definition

- You can use `packed=true` to specify a field declared with the `REPEATED` modifier. The syntax is as follows:

```
repeated int64 lockid = 6 [packed = true];
```

- You can use `default = 1` to specify the default value for a field declared with the `OPTIONAL` modifier. The syntax is as follows:

```
optional int32 logintime = 5 [default = 1];
```

- You can specify the fields in string and bytes types as encrypted fields. The syntax is as follows:

```
required string name = 2[(tcaplusservice.tcaplus_crypto) = true];
```

Nested type information

TcaplusDB supports nested types. A nested type can contain another one as its field. You can also define a new nested type in another one.
Similar to table definition, nested type definition is also declared with a message, but the nested type structure cannot contain definition of extended information such as "primary key" and "index".
TcaplusDB supports up to 128 levels of consecutive nesting; however, you are not recommended to use a high number of nesting levels, as it will compromise the data access performance.

Sample Code for Table Description File in proto2

Below is a table description file in compliance with the proto2 syntax specification:

```
syntax = "proto2"; // Indicate compliance with the proto2 syntax specification.
package myTcaplusTable; // Custom package name
import "tcaplusservice.optionv1.proto"; // Define some common information of the
TcaplusDB table, which should be imported in your table definition.
message player { // Use a message to define the table, and the message name is th
e table name.
// Only the message for which the primary key option (tcaplusservice.tcaplus_prim
ary_key) is specified can be recognized as a TcaplusDB business data table. Other
wise, the message is only a structure.
// The primary key option specifies the list of primary key field names that are
separated by commas. Up to four fields can be specified as primary keys.
option(tcaplusservice.tcaplus_primary_key) = "uin,name,region";
// Use the "tcaplusservice.tcaplus_index" option to specify the TcaplusDB index.
// The index keys included in each index must be primary keys, and the intersecti
```

©2013-2022 Tencent Cloud. All rights reserved.

```
on of all index field sets cannot be empty.
option(tcaplusservice.tcaplus_index) = "index_1(uin,region)";
option(tcaplusservice.tcaplus_index) = "index_2(uin,name)";
// "option(tcaplusservice.tcaplus_sharding_key) = "uin";". You can explicitly set the index sharding field. If you do not explicitly set it, the system will use a primary key field as the sharding field by default.
option(tcaplusservice.tcaplus_field_cipher_suite) = "DefaultAesCipherSuite"; // Use the default "DefaultAesCipherSuite" encryption function. This parameter is optional.
option(tcaplusservice.tcaplus_cipher_md5)= "62fee3b53619b7f303c939964c6f2c4b"; // Set the MD5 value of the encryption password string. This parameter is optional.
// The following shows the definition of table fields.
// A TcaplusDB table supports the following data types:
// Non-nested types: int32, int64, uint32, uint64, sint32, sint64, bool, fixed64, sfixed64, double, fixed32, sfixed32, float, string, bytes
// Nested type: message
// TcaplusDB supports three field modifiers: REQUIRED, OPTIONAL and REPEATED.
// (Up to four) primary key fields
required int64 uin = 1; // The primary key fields must be declared with the REQUIRED modifier. Nested data types are not supported.
required string name = 2[(tcaplusservice.tcaplus_crypto) = true]; // The string-type and bytes-type fields in a message can be specified as encrypted fields. This parameter is optional.
required int32 region = 3;
// A table can contain up to four primary key fields.
// General value field.
required int32 gamesvrid = 4; // General fields can be declared with REQUIRED, OPTIONAL, and REPEATED modifiers.
optional int32 logintime = 5 [default = 1];
repeated int64 lockid = 6 [packed = true]; // The "packed=true" option should be specified for the fields declared with the REPEATED modifier.
optional bool is_available = 7 [default = false]; // You can specify a default value for an OPTIONAL field.
optional pay_info pay = 8; // Value fields can be in custom structure types.
}
message pay_info { // Use a message to define the structure.
required int64 pay_id = 1;
optional uint64 total_money = 2;
optional uint64 pay_times = 3;
optional pay_auth_info auth = 4;
message pay_auth_info { // Structure types support nested definition.
required string pay_keys = 1;
optional int64 update_time = 2;
}
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Sample Code for Table Description File in proto3

Below is a table description file in compliance with the proto3 syntax specification:

```
syntax = "proto3"; // Indicate compliance with the proto3 syntax specification.
package myTcaplusTable; // Custom package name
import "tcaplusservice.optionv1.proto"; // Define some common information of the TcaplusDB table, which should be imported in your table definition.
message player { // Use a message to define the table, and the message name is the table name.
// Only the message for which the primary key option (tcaplusservice.tcaplus_primary_key) is specified can be recognized as a TcaplusDB business data table. Otherwise, the message is only a structure.
// The primary key option specifies the list of primary key field names that are separated by commas. Up to four fields can be specified as primary keys.
option(tcaplusservice.tcaplus_primary_key) = "uin,name,region";
// Use the "tcaplusservice.tcaplus_index" option to specify the TcaplusDB index.
// The index keys included in each index must be primary keys, and the intersection of all index field sets cannot be empty.
option(tcaplusservice.tcaplus_index) = "index_1(uin,region)";
option(tcaplusservice.tcaplus_index) = "index_2(uin,name)";
// "option(tcaplusservice.tcaplus_sharding_key) = "uin";". You can explicitly set the sharding field. If you do not explicitly set it, the system will use a primary key field as the sharding field by default.
option(tcaplusservice.tcaplus_field_cipher_suite) = "DefaultAesCipherSuite"; // Use the default "DefaultAesCipherSuite" encryption function. This parameter is optional.
option(tcaplusservice.tcaplus_cipher_md5)= "62fee3b53619b7f303c939964c6f2c4b"; // Set the MD5 value of the encryption password string. This parameter is optional.
// The following shows the definition of table fields.
// A TcaplusDB table supports the following data types:
// Non-nested types: int32, int64, uint32, uint64, sint32, sint64, bool, fixed64, sfixed64, double, fixed32, sfixed32, float, string, bytes
// Nested type: message
// (Up to four) primary key fields
int64 uin = 1; // The primary key fields must be in non-nested types.
string name = 2[(tcaplusservice.tcaplus_crypto) = true]; // The string-type and bytes-type fields in a message can be specified as encrypted fields. This parameter is optional.
int32 region = 3;
// A table can contain up to four primary key fields.
// General value field.
int32 gamesvrid = 4;
int32 logintime = 5;
int64 lockid = 6;
bool is_available = 7;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
pay_info pay = 8; // Value fields can be in custom structure types.
}
message pay_info { // Use a message to define the structure.
int64 pay_id = 1;
uint64 total_money = 2;
uint64 pay_times = 3;
pay_auth_info auth = 4;
message pay_auth_info { // Structure types support nested definition.
string pay_keys = 1;
int64 update_time = 2;
}
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Table Definition in TDR

Last updated：2020-12-21 10:51:22

TencentDB for TcaplusDB supports two table definition languages: Protocol Buffers (Protobuf) and Tencent Data Representation (TDR).
Protobuf is a method of serializing structured data developed by Google, which emphasizes simplicity and performance.
TDR is a platform-neutral data description language developed by Tencent, which combines the advantages of XML, binary language, and object-relational mapping (ORM) and is widely used in data serialization scenarios of Tencent's games.
Both languages are equally useful. Use either of them based on your usage habits.
This document describes how to define tables in TDR.

Table Definition in TDR

TDR supports GENERIC tables and LIST tables. GENERIC tables represent the attributes of elements in the form of tables, such as students, employers, and game players. LIST tables are a list of records, such as game leaderboards and in-game emails (usually the last 100 emails).

You can create different types of tables in one XML file.

Table Definition

- The `metalib` element is the root element of XML files.
- The `tagsetversion` attribute must be one.
- The `struct` element with the `primarykey` attribute needs to be defined as a table, or else it is just a structure.
- Each time the table structure is changed, the version attribute needs to be incremented by 1. The starting value of version attribute is always 1.
- The `primarykey` attribute is used to specify the primary key field. A GENERIC table supports up to four primary key fields and a LIST table supports up to three.
- The `splittablekey` attribute works as a shardkey to shard a TcaplusDB table into multiple shards and store them on multiple nodes. `splittablekey` must be one of the primary key fields. As a highly discrete `splittablekey` has better performance and a wide value range, we recommend a STRING `splittablekey`.
- The `desc` attribute describes the element.

©2013-2022 Tencent Cloud. All rights reserved.

- The `entry` element defines a field. Valid values: int32, string, char, int64, double, short.
- The `index` element defines an index which must contain `splittablekey` . Since the primary key can be used to query tables, the index should have a different attribute from the primary key.

Sample Code for Table Description File in TDR

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>

<metalib name="tcaplus_tb" tagsetversion="1" version="1">

<!-- generic_table `users`, store the user' information -->
<!-- an user may has many roles -->
<struct name="users" version="1" primarykey="user_id,username,role_id" splittable
key="user_id" desc="user table">
<entry name="user_id" type="uint64" desc="user id"/>
<entry name="username" type="string" size="64" desc="login username"/>
<entry name="role_id" type="int32" desc="a user can have multiple roles"/>

<entry name="level" type="int32" defaultvalue="1" desc="role's level"/>
<entry name="role_name" type="string" size="1024" desc="role's name"/>
<entry name="last_login_time" type="string" size="64" defaultvalue="" desc="user
login timestamp"/>
<entry name="last_logout_time" type="string" size="64" defaultvalue="" desc="user
logout timestamp"/>

<index name="index1" column="user_id"/>
</struct>

<!-- list_table `mails`, store the role's mails -->
<struct name="mails" version="1" primarykey="user_id,role_id" desc="mail table">
<entry name="user_id" type="uint64" desc="user id"/>
<entry name="role_id" type="int32" desc="a user may has many roles"/>

<entry name="text" type="string" size="2048" desc="mail text"/>
<entry name="send_time" type="string" size="64" defaultvalue="" desc="timestamp o
f the mail sent"/>
<entry name="read_time" type="string" size="64" defaultvalue="" desc="timestamp o
f the mall read"/>
</struct>

</metalib>
```

Additionally, you can use `union` to create nested type data.

©2013-2022 Tencent Cloud. All rights reserved.

The `union` element defines a simple type as a collection (union) of values from specified simple data types, such as INT and STRING. You can use `union` as a custom data type.

The `Macro` tag is used to define constants.

```
<macro name="DB_MAX_USER_MSG_LEN" value="301" desc="Max length of the message tha
t user can define"/>
<union name="DBPlayerMsg" version="1" desc="DB Player message">
<entry name="SysMsgID" type="uint8" desc="Message ID" />
<entry name="UsrMsg" type="string" size="DB_MAX_USER_MSG_LEN" desc="player create
d message" />
</union>
```

©2013-2022 Tencent Cloud. All rights reserved.

Creating Cluster

Last updated：2024-04-26 09:39:22

Overview

This document describes how to create a TcaplusDB cluster in the console.

Note：

For the consultation or purchase of TencentDB for TcaplusDB, please submit a ticket to contact the Tencent Cloud team.

Prerequisites

You have signed up for a Tencent Cloud account and completed identity verification.

Directions

1. Log in to the TencentDB for TcaplusDB console, select **Cluster List** on the left sidebar, and click **Create Cluster**.
2. On the displayed purchase page, specify the following configurations, and click **Buy Now**.

Network: select a VPC and a subnet. You can select one VPC and one subnet only when creating a cluster, which cannot be modified once created.

Connection Protocol: select a connection protocol (data description protocol).

Cluster Name: set the cluster name, which must be unique under the account and can contain 1–32 characters. You can rename the cluster at any time.

Access Password: set the cluster access password, which must meet the following requirements:

8–64 characters in length. The password of 12 or more characters is recommended.

Cannot start with a slash (/).

At least contain three types:

Lowercase letters (a-z)

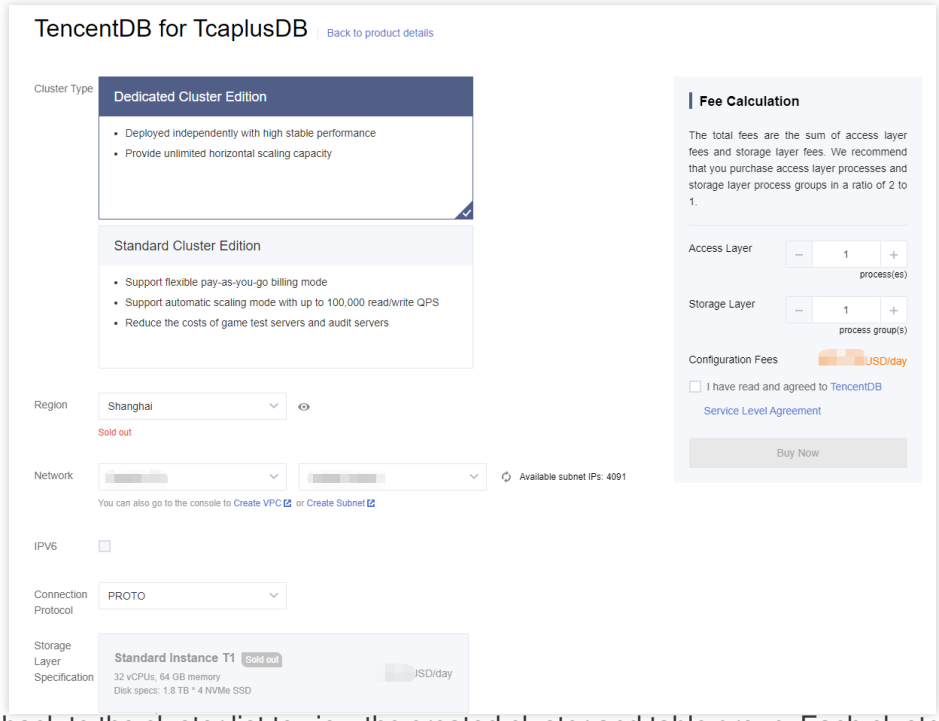
Uppercase letters (A-Z)

Digits (0-9)

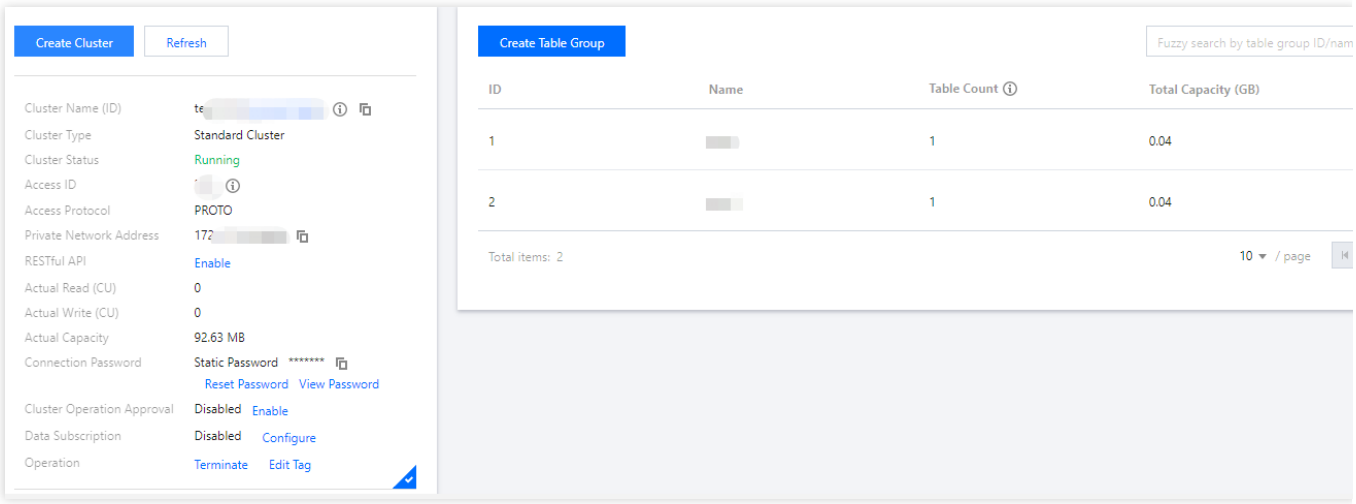
Special symbols `\\~!@#$%^&*_-+=\\|(){}[]:;'<>,.?/` .

Table Group Name and **Table Group ID**: create a default table group for the cluster and name it. The table group ID can be automatically generated or customized.

©2013-2022 Tencent Cloud. All rights reserved.



3. Go back to the cluster list to view the created cluster and table group. Each cluster will be assigned a unique cluster ID.



©2013-2022 Tencent Cloud. All rights reserved.

Creating Table Group

Last updated：2022-03-15 16:01:14

Overview

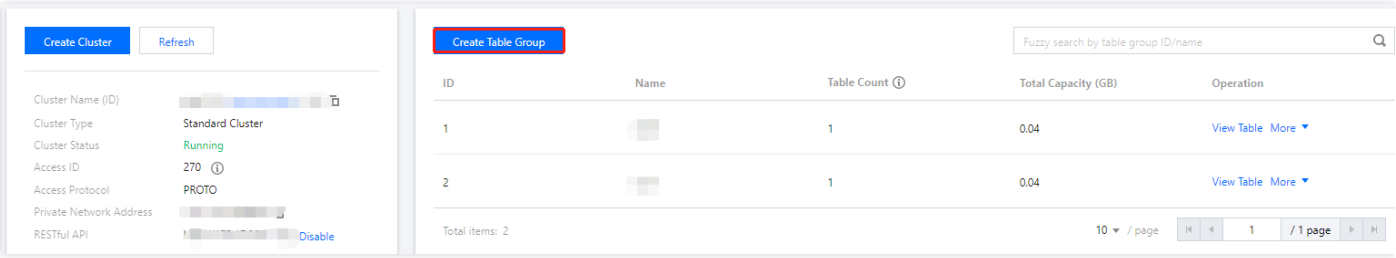
This document describes how to create a table group in the TcaplusDB console.

Prerequisite

You have created a TcaplusDB cluster.

Directions

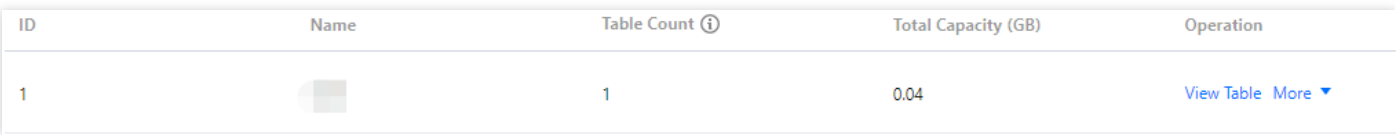
1. Log in to the TcaplusDB console, select **Cluster List** on the left sidebar, select a desired cluster, and click **Create Table Group**.



2. In the pop-up window, specify table group configurations, and click **Create Table Group**. After that, you will be prompted that the creation has been successful.

- **Select Cluster**: You can switch to another cluster.
- **Table Group Name**: The name must contain 4-64 characters.

3. Go back to the table group list to view the name, ID, table quantity, and total capacity of the table group.



©2013-2022 Tencent Cloud. All rights reserved.

Creating Table

Last updated：2022-06-01 14:39:14

Operation Scenarios

This document describes how to create a table in the TcaplusDB Console.

Prerequisites

- You have created a TcaplusDB cluster and table group.
- You have prepared the table file according to the sample table description file.

Directions

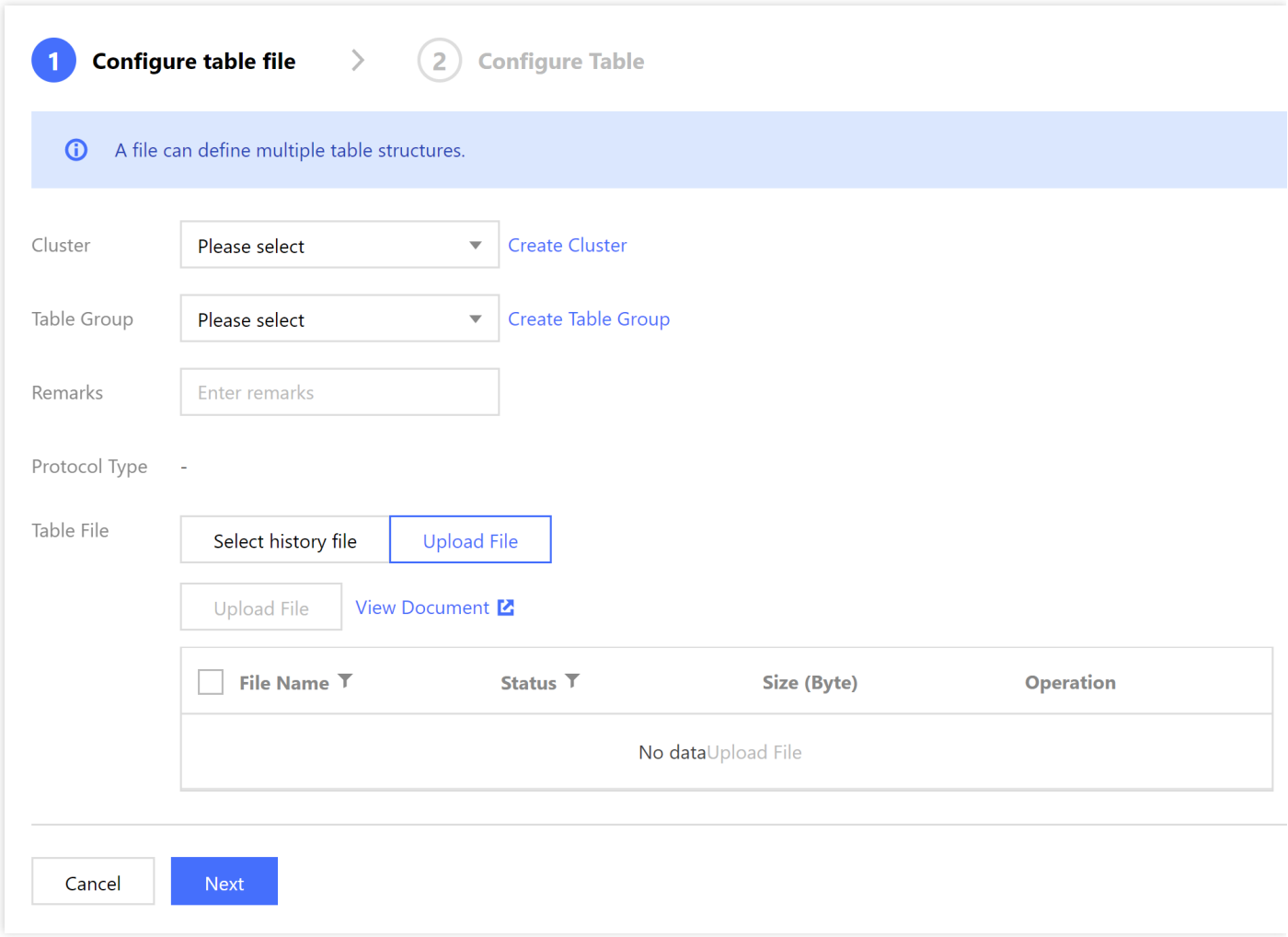
1. Log in to the TcaplusDB Console, select **Table List** on the left sidebar, and click **Create Table**.
2. Configure the table on the table creation page.

-**Cluster and Table Group**: select the target cluster and table group.

- **Table File**: you can upload a table definition file from your local file system, select a previously uploaded one, or mix and match. However, filenames must be unique. For more information, please see the sample PB table file game_players.proto.

©2013-2022 Tencent Cloud. All rights reserved.

- **Remarks**: enter the table remarks.



3. Click **Next** and the system will verify the selected table definition file.

- If the verification fails, an error will be returned, and you should modify the file accordingly and upload it again.
- If the verification is successful, the table metadata defined in the file will be displayed, and then you can proceed to the next step.

4. On the table configuration page, select the table to be created and enter the capacity and reserved read and write parameters. The daily fees of the table will be automatically calculated.
5. After confirming that the table information is correct, click **Create** and the system will prompt that the creation has been successful.

©2013-2022 Tencent Cloud. All rights reserved.

Get Connection Information

Last updated：2022-02-22 10:49:52

This document describes how to get TcaplusDB access information in the console.

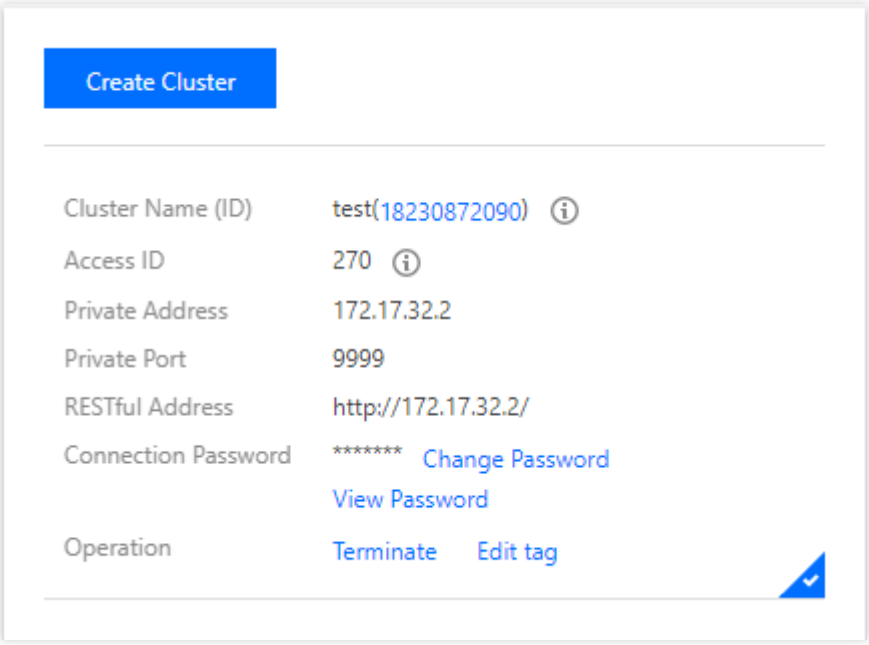
The Cluster List page provides some cluster information. The "Access ID", "Connection password", "Private Address", and "Private Port" displayed on this page are important parameters for a client to connect to TcaplusDB.

Access ID: access ID to which the cluster belongs.

Connection Password: application password.

Private address: the address to connect to TcaplusDB.

Private port: the port to connect to TcaplusDB.



©2013-2022 Tencent Cloud. All rights reserved.

Accessing TcaplusDB

Last updated：2022-03-29 15:49:30

TcaplusDB can be accessed and read through multiple ways such as client tool, SDK toolkits in various programming languages, and RESTful API.

Accessing TcaplusDB Through Client Tool

You can use the tcaplus_client for Linux to access TcaplusDB.

Accessing TcaplusDB Through SDK for C++ API

You can use the SDK for C++ to access TcaplusDB.

Accessing TcaplusDB Through RESTful API

You can access TcaplusDB through RESTful API. For more information, please see Description.

- You can access TcaplusDB through RESTful API for Python as instructed here.
- You can access TcaplusDB through RESTful API for PHP as instructed here.
- You can access TcaplusDB through RESTful API for Go as instructed here.
- You can access TcaplusDB through RESTful API for Java as instructed here.

©2013-2022 Tencent Cloud. All rights reserved.