TDSQL-C for MySQL

FAQs

Product Documentation

Copyright Notice

Trademark Notice

Tencent Cloud

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

FAQs

Basic Concepts

Last updated：2023-05-31 15:01:45

What is TDSQL-C for MySQL?

TDSQL-C for MySQL is a proprietary new-generation cloud native relational database developed by Tencent Cloud. It combines the strengths of traditional databases, cloud computing, and cutting-edge hardware technologies to provide elastic scalable database services, which features high performance, security, and reliability.

How is TDSQL-C for MySQL different from traditional databases?

TDSQL-C for MySQL combines the strengths of traditional databases and cloud computing. First, it has all the five characteristics of cloud computing:

On-demand self-service

Broad network access

Resource pooling

Rapid elasticity

Measured service

Secondly, TDSQL-C for MySQL truly implements the concept of log as a database through transformation and optimization of the open-source database kernel in combination with the SOA architecture and distributed storage. It optimizes the system performance at critical paths at the software level to reduce the costs of use.

Why is TDSQL-C for MySQL better than traditional databases?

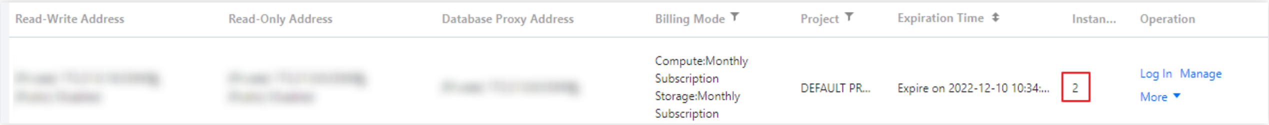
Compared with traditional databases, the cloud-native TDSQL-C for MySQL can store petabytes of data and provide high availability and reliability, fast and elastic configuration adjustment, and instant scaling. For more information, see Strengths.

What are clusters and instances in TDSQL-C for MySQL?

A TDSQL-C for MySQL cluster adopts the multi-node architecture with one read-write instance and multiple read-only instances. A single cluster supports cross-AZ management and billing.

A cluster has its ID and name. It consists of multiple instances: one read-write instance and one or multiple read-only instances.

An instance has its ID and name. It can be referred to as a database server.



Read-Write Address	Read-Only Address	Database Proxy Address	Billing Mode	Project	Expiration Time	Instan...	Operation
[redacted]	[redacted]	[redacted]	Compute:Monthly Subscription Storage:Monthly Subscription	DEFAULT PR...	Expire on 2022-12-10 10:34:...	2	Log In Manage More ▾

A cluster is a logical concept, and its configuration and specification are based on read-write and read-only instances. Different read-only instances can have different configurations, which can be separately adjusted.

All instances in a cluster share the same storage space, which doesn't need to be purchased in advance for serverless and pay-as-you-go clusters.

What programming languages are supported in TDSQL-C for MySQL?

TDSQL-C for MySQL supports all languages supported by native MySQL, such as Java, Python, PHP, Go, C, C++, .NET, and Node.js. For more information, see MySQL Connectors.

Is TDSQL-C for MySQL a distributed database?

TDSQL-C for MySQL is a distributed storage cluster implemented with the three-replica strong consistency algorithm at the storage layer. The computing engine consists of 1–16 compute nodes distributed on different servers, and the storage space can reach petabyte level, with up to 88 CPU cores and 710 GB memory supported. The storage space and computing specification can be scaled dynamically online without affecting normal business operations.

What are the use limits of TDSQL-C for MySQL?

TDSQL-C for MySQL imposes certain use limits on naming and operations to ensure the stable and secure operation of your cluster. For more information, see Use Limits.

Are self-built replica instances supported? Are there any recommended implementation methods?

Yes. After the binlog is enabled, you can sync a TDSQL-C for MySQL database to other MySQL databases to form a source-replica architecture. Tencent Cloud offers and recommends Data Transmission Service (DTS) for data sync and subscription. For more information on how to use DTS, see:

Sync from TDSQL-C for MySQL to TencentDB for MySQL

Creating MySQL or TDSQL for MySQL Data Subscription

Between May 17 and December 31, 2022, DTS allowed you to create pay-as-you-go tasks of migration and sync to TDSQL-C for MySQL free of charge.

Purchase and Billing

Last updated：2023-11-22 15:29:10

How do I reduce the costs of TDSQL-C for MySQL?

TDSQL-C for MySQL reduces TCO through pooling, features minimalist software optimizations, and supports common networks and hardware devices to release the hardware dividend. It has an elastic scaling feature and a built-in high availability architecture, where fast scaling capabilities greatly reduce the wastes of computing and storage resources. In addition, it is more cost-effective than traditional commercial databases with the same high performance and reliability.

TDSQL-C for MySQL offers flexible billing modes and specification adjustment capabilities, so you can scale cluster resources largely based on your business conditions and reasonably query the approximate costs of required resources before deploying your business. The computing-storage separation architecture further allows you to adjust the specification on demand with no need to select a certain combination. Instead of upgrading or downgrading both the compute node and storage space specifications within the fixed range, you can easily adjust the compute node and storage space separately as needed. This helps you implement fine-grained cost management to the greatest extent.

What are the billable items of TDSQL-C for MySQL?

The billable items of TDSQL-C for MySQL include compute node, storage space, backup (free of charge currently), and database audit (optional). For more information, see Billing Overview.

What types of files will use the storage space?

Such files include data files, index files, log files (redo, undo, slow, and error logs), temporary files, and a small number of system files. For more information, see Billing Overview.

What billing modes does TDSQL-C for MySQL support?

TDSQL-C for MySQL adopts a computing-storage separation architecture, so the compute nodes and storage space can be billed separately in different billing modes:

The billing modes of compute nodes include monthly subscription, pay-as-you-go, and serverless.

The billing modes of the storage space include monthly subscription (available only when the billing mode of compute nodes is monthly subscription) and pay-as-you-go.

How do I choose an appropriate billing mode?

If your business peak traffic fluctuates greatly, you only use the database infrequently in the development and test environments, or you use SaaS in scenarios such as website building for small and medium-sized enterprises, we recommend you choose the serverless billing mode. In this mode, the database can automatically start, stop, and scale based on the business load in an imperceptible manner without causing disconnections, and the service is billed by the actual computing and storage resource usage.

If your business has a stable traffic and runs for a long term, we recommend that you choose the monthly subscription billing mode. In this mode, the longer the purchase period, the higher the discount as displayed on the purchase page.

If your business often fluctuates greatly and instantaneously, we recommend that you choose the pay-as-you-go billing mode. In this mode, instances can be released immediately after use to reduce the costs.

If you need a large storage space, you can choose the monthly subscription billing mode. In this mode, the storage space is billed in a tiered manner. If the used storage space exceeds 3,000 GB, you can enjoy a even lower unit price. For more information on storage space pricing, see Selecting Billing Mode for Storage Space.

Which billing modes does a TDSQL-C for MySQL read-only instance support?

The computing and storage billing modes of a TDSQL-C for MySQL read-only instance must be the same as those of its source instance.

If both the computing and storage space billing modes of the source instance are monthly subscription, you can only add read-only instances with the monthly subscription billing mode for both computing and storage space.

If the computing and storage space billing modes of the source instance are monthly subscription and pay-as-you-go respectively, you can only add read-only instances with the monthly subscription billing mode for computing and pay-as-you-go billing mode for storage space.

If both the computing and storage space billing modes of the source instance are pay-as-you-go, you can only add read-only instances with the pay-as-you-go billing mode for both computing and storage space.

In serverless billing mode, you cannot add read-only instances.

Does TDSQL-C for MySQL charge for the backup space?

The backup storage space of TDSQL-C for MySQL is free of charge currently and will be billed on a pay-as-you-go basis as needed in the future. Therefore, plan your backup storage files reasonably.

Will my pay-as-you-go cluster be eliminated immediately when my account balance becomes negative?

No.

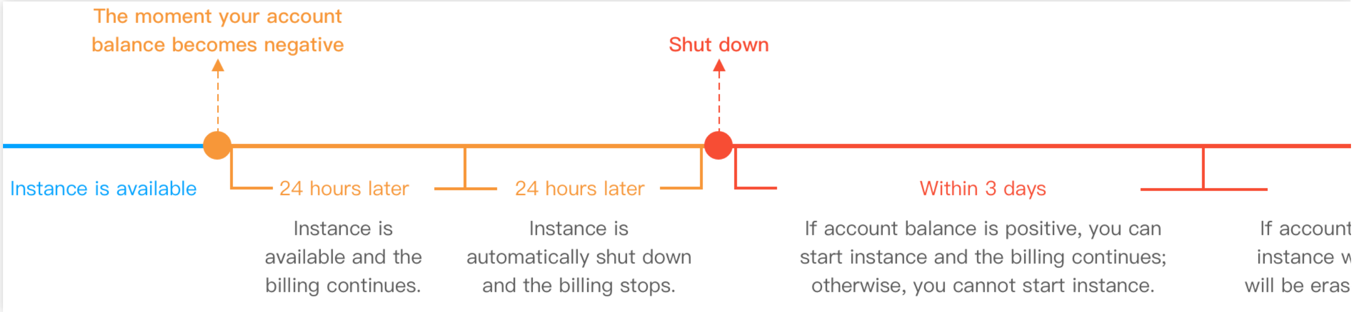
The moment your account balance becomes negative:

You can continue to use your TDSQL-C for MySQL cluster in 24 hours. We will continue to bill you for this period.

After 24 hours, your TDSQL-C for MySQL cluster will be automatically isolated and moved into the recycle bin, and the billing will stop.

If you top up your account within 3 days after the isolation to a positive balance, the billing will continue, and the cluster will be automatically recovered for normal use.

If your account balance remains negative after 3 days, the isolated cluster will be deactivated and put into the repossession queue. After it is repossessed, all data in it will be cleared and cannot be recovered.


The moment your account balance becomes negative
Shut down
Instance is available
24 hours later
Instance is available and the billing continues.
24 hours later
Instance is automatically shut down and the billing stops.
Within 3 days
If account balance is positive, you can start instance and the billing continues; otherwise, you cannot start instance.
If account
instance w
will be eras

How do I purchase TDSQL-C for MySQL?

You can purchase TDSQL-C for MySQL in the following two ways:

Create a cluster in the console.

You can click **Create** in the **Cluster List** to quickly purchase a cluster. This method is suitable for existing TencentDB users.

Directly purchase the service on the purchase page.

You can purchase a cluster on the purchase page. This method is suitable for new TencentDB users.

Before using either method, you must register a Tencent Cloud account and complete identity verification.

To register a Tencent Cloud account:Click here to sign up for a Tencent Cloud account.

To complete identity verification:Click here to verify your identity.

You need to set the database configuration, basic information, and advanced configuration in both purchase methods.

For more information on the configuration, see Purchase Methods.

How do I change the billing mode of a pay-as-you-go cluster to monthly subscription?

TDSQL-C for MySQL implements this change by generating a renewal order. You can select **More** > **Pay-as-You-Go to Monthly Subscription** in the **Operation** column in the Cluster List for change.

Rest assured that access to your business will not be affected during the change from pay-as-you-go billing to monthly subscription billing. However, a monthly subscribed cluster cannot be changed back to a pay-as-you-go cluster.

How do I change the billing mode of a pay-as-you-go cluster to serverless?

TDSQL-C for MySQL converts the cluster type on the backend to change the billing mode from pay-as-you-go to serverless. After conversion, the billing mode is a postpaid billing mode, where you can set the maximum and minimum computing power values as needed first and will be charged based on the actual usage of computing and storage resources upon settlement.

During the change from pay-as-you-go billing to serverless billing, the database can be accessed normally but will experience a momentary disconnection at the time point of change. Therefore, we recommend you configure an automatic reconnection feature for your application. However, a serverless cluster cannot be changed back to a pay-as-you-go cluster. For detailed directions, see Change from Pay-as-You-Go to Serverless Billing.

How do I renew a TDSQL-C for MySQL cluster?

You can manually renew or set automatic renewal for one or multiple TDSQL-C for MySQL clusters in the console. If you have many resources under your account, you can configure renewal settings in the Renewal Management Center, such as batch cluster renewal, automatic renewal, and unified expiration date.

How do I delete a cluster?

If you are sure that you no longer need a cluster, you can find it in the **Cluster List** and select **More** > **Delete** in the **Operation** column to delete it.

Note:

After a cluster is deleted, all instances (including read-write and read-only instances) in it will also be automatically deleted.

After a pay-as-you-go cluster is deleted, its billing will stop automatically.

If a monthly subscribed cluster is deleted before it expires, the fees of all instances in it will be calculated again by the usage duration at the pay-as-you-go price, and your original payment will be refunded after fees incurred are deducted from it.

How do I restore an instance?

You can restore accidentally deleted, expired, or suspended instances from the recycle bin before they are terminated.

Note:

Read-only instances can be restored only after the read-write instance is restored.

You can only restore an instance before it is terminated.

If you click **Terminate** in the **Operation** column, the instance will be eliminated immediately and cannot be restored. Therefore, proceed with caution.

Compatibility and Format

Last updated：2023-05-31 16:10:39

Is TDSQL-C for MySQL compatible with open-source MySQL?

Yes. TDSQL-C for MySQL is fully compatible with MySQL 5.7 and 8.0.

Which transaction isolation levels are supported?

TDSQL-C for MySQL supports two isolation levels: `READ_COMMITTED` (default) and `REPEATABLE_READ` . You can modify the `tx_isolation` parameter in **Parameter Settings** to change the transaction isolation level. For detailed directions, see Setting Instance Parameters.

Is the binlog format different from the native format of MySQL?

There is no difference between them.

Are performance schema and sys schema supported?

Yes.

Are the collected table statistics different from those in open-source MySQL?

No. To ensure that the execution plans of read-write and read-only instances are consistent, every time the read-write instance updates the statistics, the change will be synced to read-only instances.No. To ensure that the execution plans of read-write and read-only instances are consistent, every time the read-write instance updates the statistics, the change will be synced to read-only instances.

Does TDSQL-C for MySQL support XA transactions? Are there any differences in this regard between TDSQL-C for MySQL and open-source MySQL?

Yes, and there are no differences.

Does TDSQL-C for MySQL support full-text indexing?

Yes.

Is Percona Toolkit supported?

Yes. However, we recommend you use online DDL.

Is gh-ost supported?

Yes. However, we recommend you use online DDL.

Connection and Network

Last updated：2023-05-15 15:33:27

How do I access TDSQL-C for MySQL?

You can connect to a TDSQL-C for MySQL instance in the following ways:

Private network connection: A CVM instance can be used to connect to the private network address of a TDSQL-C for MySQL instance. This method utilizes the high-speed private network of Tencent Cloud and features low delay. Note that the CVM and database instances must be under the same account and in the same VPC in the same region.

Public network connection: You can connect to your TDSQL-C for MySQL cluster at its public network address. You need to manually enable the public network address, which you can view on the instance details page in the console and disable if no longer needed.

DMC connection: You can access TDSQL-C for MySQL through Database Management Center (DMC).

For directions, see Connecting to Cluster via the Console.

How do I enable the public network addresses of a TDSQL-C for MySQL instance?

TDSQL-C for MySQL public network addresses include public read-write address and public read-only address. You can enable/disable them on the **Cluster Details** page in the console as instructed in Enabling/Disabling Public Network Address.

Can Lighthouse instance directly access TDSQL-C for MySQL?

The Lighthouse instance cannot directly connect to the TDSQL-C for MySQL through private network address. If you need to do so, see Getting Started with the CCN.

Can TDSQL-C for MySQL directly interconnect with servers of other cloud vendors over the private network?

No.

You can enable public network access for a TDSQL-C for MySQL instance on the **Cluster Details** page, and then connect to the instance by using a server from another cloud vendor.

Can I change the public network address of a TDSQL-C for MySQL instance?

No.

How do I change the network?

TDSQL-C for MySQL allows you to change the VPC in the **Basic Info** section on the **Cluster Details** page in the console.

How many read-only instances can a TDSQL-C for MySQL cluster contain at most?

A cluster can contain only one read-write instance and up to 15 read-only instances.

How do I lower the load of a read-write instance?

TDSQL-C for MySQL allows you to create one or more read-only instances in a cluster, which are suitable for read/write separation and one-write-multiple-read application scenarios and capable of lowering the load of the read-write instance. For more information, see Creating Read-Only Instance.

Does TDSQL-C for MySQL support the JDBC protocol?

TDSQL-C for MySQL is fully compatible with MySQL, so it supports the JDBC protocol.

Features

Last updated：2024-03-26 17:05:53

How long does it take to adjust the database configuration? Will this affect my business?

TDSQL-C for MySQL configuration adjustment includes computing specification adjustment and storage space adjustment.

Computing specification adjustment

In a provisioned resource instance mode, computing specification adjustment can be completed in seconds, but there may be a momentary disconnection of 3–5 seconds. Make sure that your business has an automatic reconnection mechanism. We recommend you perform this operation during off-peak hours.

In serverless instance mode, the upper and lower limits of the computing power are adjusted automatically, and no momentary disconnections will occur to affect the business.

Storage space adjustment

In pay-as-you-go billing mode, you don't need to manually adjust the storage space size, and the system will bill you hourly by actual usage. In monthly subscription billing mode, the prepaid storage space to be changed to must be greater than the used storage space. Changing the billing mode of the storage space won't affect the business.

How long does it take to add a read-only instance? Will this affect my business?

It takes less than 30 seconds to add a read-only instance, which has no impact on your business. The time it takes has nothing to do with the instance storage space. For detailed directions, see Creating Read-Only Instance.

Is there a replication delay between a read-write instance and a read-only instance?

Traditional cloud databases have a delay of seconds, but TDSQL-C for MySQL uses redo logs for replication, which greatly reduces the delay to milliseconds.

In which cases will the replication delay increase?

The replication delay in TDSQL-C for MySQL is much shorter than that in traditional databases, and you can view the **Redo Log Based Replication Lag of Replica Instance** metric of the instance to check the actual delay.

The replication delay will increase in the following cases:

The write load of the read-write instance is too high, and too many redo logs are generated, which cannot be applied by the read-only instance in time.

The load of the read-only instance is too high, which preempts too many resources that should be used for redo logs.

The I/O bottleneck is hit, severely slowing down redo log reads/writes.

How do I perform automatic failover?

TDSQL-C for MySQL adopts a high-availability cluster architecture. The system automatically detects failed nodes and starts a new compute node to replace the failed one and continue providing read-write access.

Can the RPO be guaranteed at zero when a single node fails?

Yes. TDSQL-C for MySQL uses the smart storage tiering mechanism based on three replicas to guarantee the data integrity after failover, where hot data is stored in fast storage (NVMe + SSD replica), warm data is stored in low-redundancy storage (SSD EC), and cold data is stored in slow storage (HHD).

How do I add fields and indexes online?

TDSQL-C for MySQL supports native MySQL online DDL and open-source tools pt-osc and gh-ost. In addition, it also supports instance DDL operations for special scenarios. For more information, see Instant DDL Overview.

Is BULK INSERT supported when there is only data write? How many values at most can be inserted at a time?

Yes. The maximum number of values that can be inserted at a time is determined by the `max_allowed_packet` parameter. You can modify it in the console to specify the limit. The value must be a multiple of 1,024 between 1,024 and 1,073,741,824. For detailed directions, see Setting Instance Parameters.

Console Operations

Last updated：2024-03-26 17:14:48

Version Upgrade

How do I upgrade the version of TDSQL-C for MySQL?

Currently, TDSQL-C for MySQL doesn't support database version upgrade, such as from MySQL 5.7 to 8.0. You can create a new instance on the target version and migrate the source data to it through DTS. For directions, see Migrating to TDSQL-C for MySQL.

For detailed directions on how to upgrade the kernel minor version, such as from v2.0.12 of MySQL 5.7 to v2.0.14, see Upgrading Kernel Minor Version.

Will version upgrade affect the data?

No.

TDSQL-C for MySQL adopts quick in-place upgrade. It uses redo logs to store the data to implement super-fast upgrade. A momentary disconnection may occur in rare cases. Make sure that your business has an automatic reconnection mechanism.

How do I view the differences between different kernel minor versions of TDSQL-C for MySQL?

TDSQL-C for MySQL releases kernel minor version updates from time to time, including new features, performance optimizations, and bug fixes. For more information on updates, see Kernel Version Release Notes.

Data Import and Export

How do I import data into TDSQL-C for MySQL?

TDSQL-C for MySQL allows you to import data in the following ways:

Import incremental data continuously by using DTS to sync the data. For directions, see Sync to TDSQL-C for MySQL.

Import data through DMC. You can import existing local files or files stored in COS of up to 10 GB in size at a time. For import files from COS, you need to upload the data to COS first and get the COS file address as instructed in Uploading Objects and Downloading Objects.

How do I import a SQL file greater than 10 GB in size?

You can upload a file greater than 10 GB in size in the following ways:

Split the SQL file into multiple files smaller than 10 GB and upload them in batches.

Open the SQL file, directly copy the commands in it, and paste and run them in the database.

How do I migrate data in TDSQL-C for MySQL?

TDSQL-C for MySQL allows you to migrate data through DTS or a command line tool.

For using DTS, see Migrating to TDSQL-C for MySQL.

For using a command line tool, see Migrating with Command Line Tool.

How do I export data from TDSQL-C for MySQL?

You can export data from TDSQL-C for MySQL in the following two ways:

Use DMC to export data.

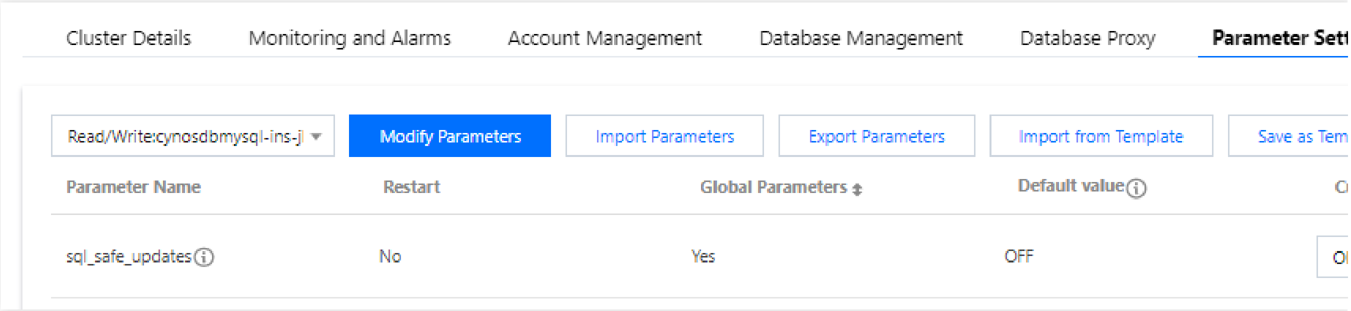
TDSQL-C for MySQL is fully compatible with native MySQL at the computing layer, so you can use MySQL's native tools such as MySQLDumper. It also supports open-source data migration tools. For more information, see Migrating with Command Line Tool.

Parameters/Metrics

Can TDSQL-C for MySQL implement a feature similar to safe-updates?

safe-updates is a safe mode of the database that limits the update (UPDATE or DELETE) of an entire table when an application bug occurs or the database admin performs a maloperation. Similarly, TDSQL-C for MySQL provides the `sql_safe_updates` parameter to achieve this.

You can enable the `sql_safe_updates` parameter in **Parameter Settings** in the console as shown below:



Parameter Name	Restart	Global Parameters	Default value	C
sql_safe_updates	No	Yes	OFF	O

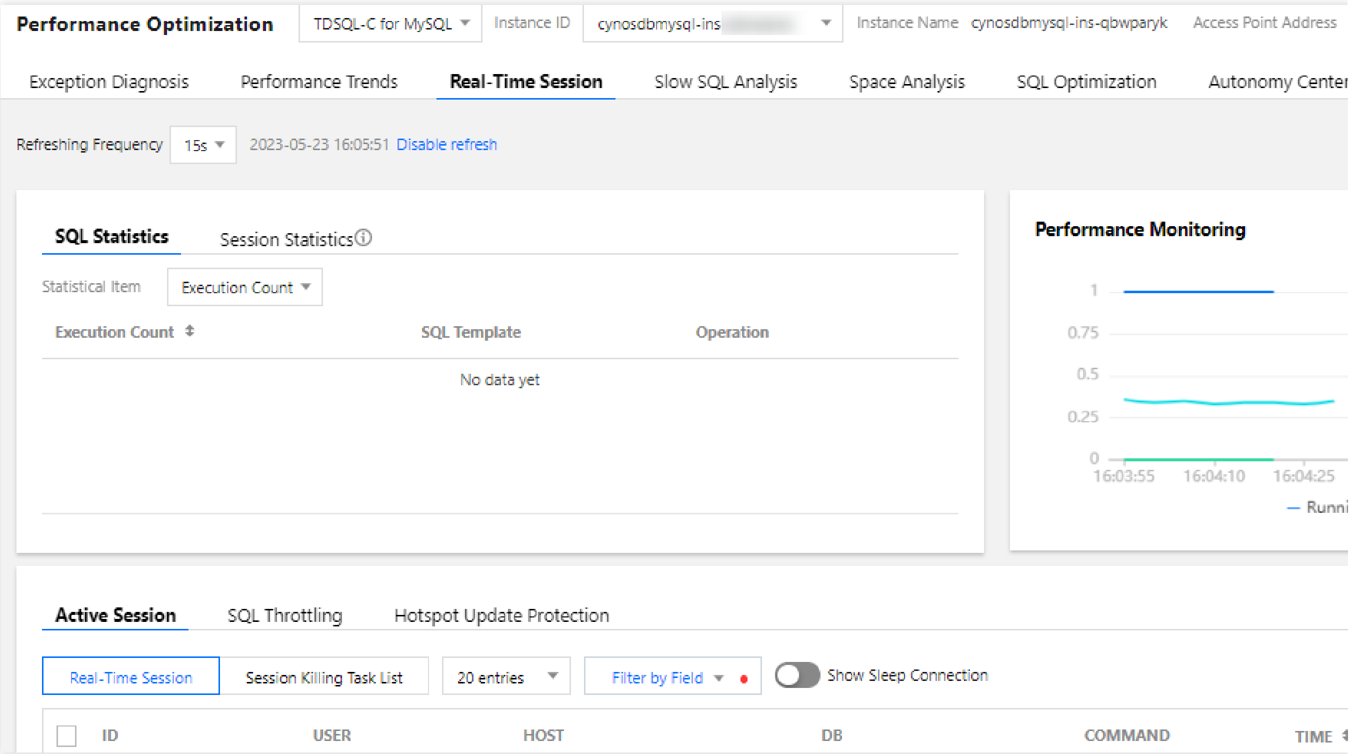
Why does the value of `Threads_created` in TDSQL-C for MySQL keep increasing?

`Threads_created` indicates the number of running threads. If its value keeps increasing, a small amount of memory will be consumed, and more connections will be occupied. You can perform the following operations to observe and end unnecessary sessions.

Observe how the value of `Threads_created` changes on the **Instance Monitoring** page.

Run the `show processlist` command or use the real-time session feature of DBbrain to check the details of running threads.

If the value of `Threads_created` keeps increasing, you can run the `Kill` command to end some sessions based on your actual business conditions. You can also click **Kill Session** or **Kill Sessions during a Period** on the **Real-Time Session** tab in the DBbrain console.



Instance ID cynosdbmysql-ins[redacted]
Instance Name cynosdbmysql-ins-qbwparyk
Access Point Address
Refreshing Frequency 15s 2023-05-23 16:05:51
Statistical Item Execution Count

Execution Count	SQL Template	Operation
No data yet		

20 entries

ID	USER	HOST	DB	COMMAND	TIME

How do I modify the `lower_case_table_names` parameter?

The `lower_case_table_names` parameter indicates the table name case sensitivity. You can set this parameter when purchasing a cluster or modify it in the console.

If `lower_case_table_names` is set to `0`, table names will be in the specified letter case and be case-sensitive during comparison. If it is set to `1`, table names will be stored in lowercase and be case-insensitive during comparison. For detailed directions on how to modify this parameter, see Setting Instance Parameters.

Note:

In MySQL 8.0, the table name case sensitivity can be set only during instance purchase but cannot be modified after the purchase.

How do I modify the maximum number of connections?

You can modify the maximum number of connections in TDSQL-C for MySQL by modifying the `max_connections` parameter in the console. The value range of this parameter is 1–10240, indicating the maximum number of allowed concurrent client connections. For more information, see Setting Instance Parameters.

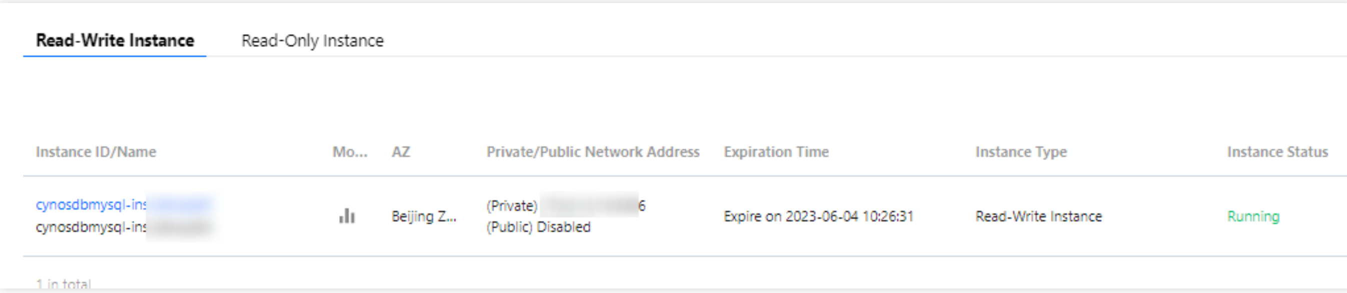
Configuration Change

How do I adjust the configuration of a TDSQL-C for MySQL instance?

TDSQL-C for MySQL uses an architecture where computing and storage resources are separated and all compute nodes share the same data. In cross-server configuration adjustment, no data migration is required; therefore, configuration upgrade/downgrade can be completed within seconds. When the current performance or storage capacity of an instance cannot meet the needs of business changes, or the instance performance is surplus and you want to reduce the costs, you can adjust the instance configuration accordingly.

TDSQL-C for MySQL configuration adjustment includes compute node adjustment and storage space adjustment.

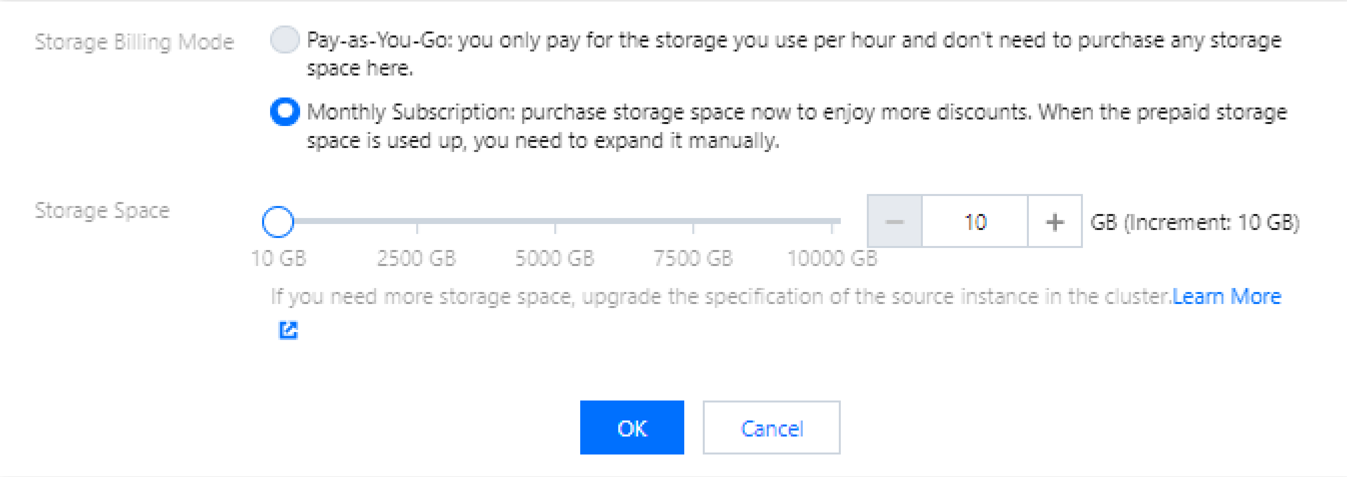
Compute node configuration adjustment: You can adjust the configuration of compute nodes in the **Instance List**.



Instance ID/Name	Mo...	AZ	Private/Public Network Address	Expiration Time	Instance Type	Instance Status
cynosdbmysql-ins[redacted] cynosdbmysql-ins[redacted]	ılı	Beijing Z...	(Private) [redacted]6 (Public) Disabled	Expire on 2023-06-04 10:26:31	Read-Write Instance	Running

1 in total

Storage space configuration adjustment: You can adjust the configuration of the storage space on the **Cluster Details** page.



Storage Billing Mode
○ Pay-as-You-Go: you only pay for the storage you use per hour and don't need to purchase any storage space here.
● Monthly Subscription: purchase storage space now to enjoy more discounts. When the prepaid storage space is used up, you need to expand it manually.

Storage Space: 10 GB (Increment: 10 GB)

If you need more storage space, upgrade the specification of the source instance in the cluster.Learn More

Note:

If the storage space billing mode is pay-as-you-go, you don't need to expand the disk capacity, and the maximum disk capacity you can use is the maximum disk space on the compute node of the source instance. To use a larger storage space, upgrade the compute node of the source instance. For more information on compute node specifications and corresponding maximum storage spaces, see Product Specifications.

If the storage space billing mode is monthly subscription, you can adjust the cluster storage space in the console. To use a storage space larger than the maximum storage space of the current compute specifications, upgrade the compute specifications of the read-write instance instance. For more information on compute node specifications and corresponding maximum storage spaces, see Product Specifications.

If the storage space billing mode is monthly subscription, the valid billing duration of the new storage capacity will start at the adjustment time and end at the cluster expiration time.

For directions, see Adjusting Compute Configuration.

Does TDSQL-C for MySQL support automatic scaling?

TDSQL-C for MySQL supports automatic scaling for the storage space and computing specification.

In serverless instance mode, the computing specification and storage space can be automatically scaled based on the actual usage, and you are charged by the used computing and storage resources.

In a provisioned resource instance mode, for example, if the computing specification is monthly subscribed and the storage space is pay-as-you-go, then the storage space will be automatically scaled based on the actual usage.

The performance doesn't support automatic scaling. You can create read-only instances to mitigate the pressure on the source instance.

Backup and Restoration

What backup types does TDSQL-C for MySQL support?

TDSQL-C for MySQL supports logical backup and snapshot backup. The former is a full backup, while the latter is a full + incremental backup. For more information, see Backup and Rollback Overview.

Logical backup: It supports full backup only, where the logical structure and content of a database are stored as SQL statements. It backs up database objects, including tables, indexes, and stored procedures. This mode features a more refined backup granularity at the database or table level and a lower impact on the performance; however, it is slow and space-intensive.

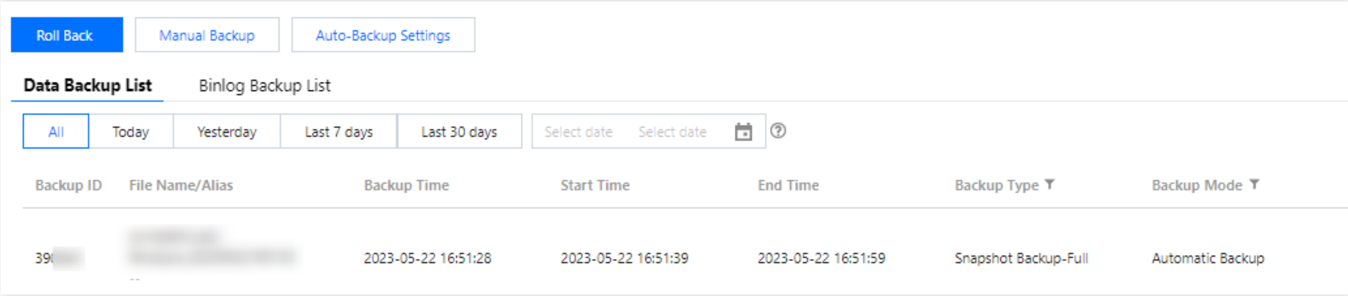
Snapshot backup: It uses the redirect-on-write (ROW) technology to take snapshots of the disks at the storage layer for backup. It features fast backup in seconds, imperceptibility to the computing layer, and low space usage.

Full backup: It copies all the data at a specific time point.

Incremental backup: It backs up only new or modified files based on the last backup.

How do I restore data in TDSQL-C for MySQL?

You can use the rollback feature to restore database tables to the original cluster.



Backup ID	File Name/Alias	Backup Time	Start Time	End Time	Backup Type	Backup Mode
39[redacted]	[redacted] --	2023-05-22 16:51:28	2023-05-22 16:51:39	2023-05-22 16:51:59	Snapshot Backup-Full	Automatic Backup

You can use the clone feature to restore data. This feature allows you to clone the data of the entire cluster to a new cluster. It can restore a cluster to any time point in the log backup retention period or from the specified backup file through clone.

You can manually generate a logical backup file in the console and use it to restore data.

Other Operations in the Console

How long does it take to create a TDSQL-C for MySQL cluster?

Generally, a cluster can be created within five minutes.

In addition, a read-only instance generally can be created within 30 seconds. The creation time has nothing to do with the data volume.

How do I modify the table name case sensitivity after initializing a TDSQL-C for MySQL instance?

You need to adjust the `lower_case_table_names` parameter of the database.

If the compatible database version is MySQL 5.7, you can click the cluster ID in the **Cluster List** to enter the cluster management page, select the **Parameter Settings** tab, and modify the `lower_case_table_names` parameter (`0` : case-sensitive; `1` : case-insensitive).

If the compatible database version is MySQL 8.0, you can only choose whether to enable case sensitivity for table names when creating an instance on the purchase page. You cannot modify the `lower_case_table_names` parameter after creating the instance.

How do I return a database?

Select **More** > **Delete** in the **Operation** column in the Cluster List to return an instance. For more information, see Deleting Instance.

How do I restore a TDSQL-C for MySQL cluster?

After a cluster is returned, a monthly subscribed instance will be retained in the recycle bin for 7 days, and a pay-as-you-go instance for 1 day. During the retention period, terminated instances can be recovered from the recycle bin.

What do I do if I accidentally delete an account or forget the password?

If you delete an account accidentally, you can click a cluster ID in the **Cluster List** to enter the instance management page, and then click **Account Management** > **Create Account** or use a SQL statement to create an account.

If you forgot the root password, select the **Account Management** tab and click **Reset Password** in the **Operation** column.. For more information, see Resetting Password.

The above operations can also be performed through the CreateAccounts API.

What are the requirements for accounts and passwords in TDSQL-C for MySQL?

The account name in TDSQL-C for MySQL can contain 1–16 letters, digits, and underscores (_) and must begin with a letter and end with a letter or digit.

A password in TDSQL-C for MySQL can contain 8–64 characters in at least three of the following character types: uppercase letters, lowercase letters, digits, and special symbols `~!@#$%^&*_-+=|\\\\(){}[]:;'<>,.?/` .

How do I set an alarm for the storage space usage?

The TencentDB storage space is monitored in the monitoring center. You can query the used storage space and utilization on the **Instance Monitoring** page.

When the disk utilization is over the percentage you set, SMS and email alarms will be triggered. You can configure alarm recipients in Tencent Cloud Observability Platform to receive such alarms. For more information on how to configure, see Alarm Policies.

How do I query the task execution status?

You can click the ☰ icon on the right of the **Cluster List** to view running tasks. You can also click **Task List** on the left to view the details of all tasks.

Database and Table

Last updated：2023-05-31 16:10:39

What are the strengths of TDSQL-C for MySQL's big table storage over traditional databases using local disks?

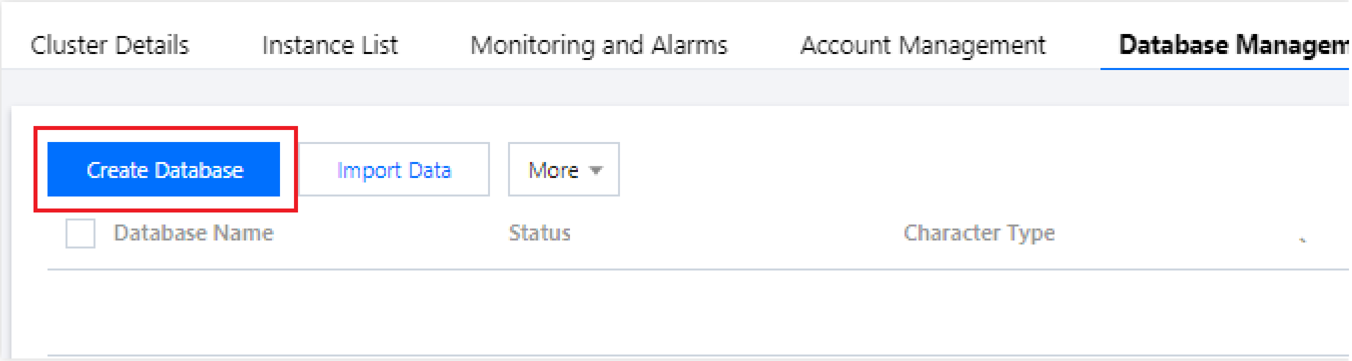
A table in TDSQL-C for MySQL is sharded physically for storage on N storage servers. Therefore, the I/O of a table is allocated to multiple disks, and the overall I/O throughput is much higher than that of centralized databases using local disks.

Are there any limits on the numbers of databases and tables that I can create?

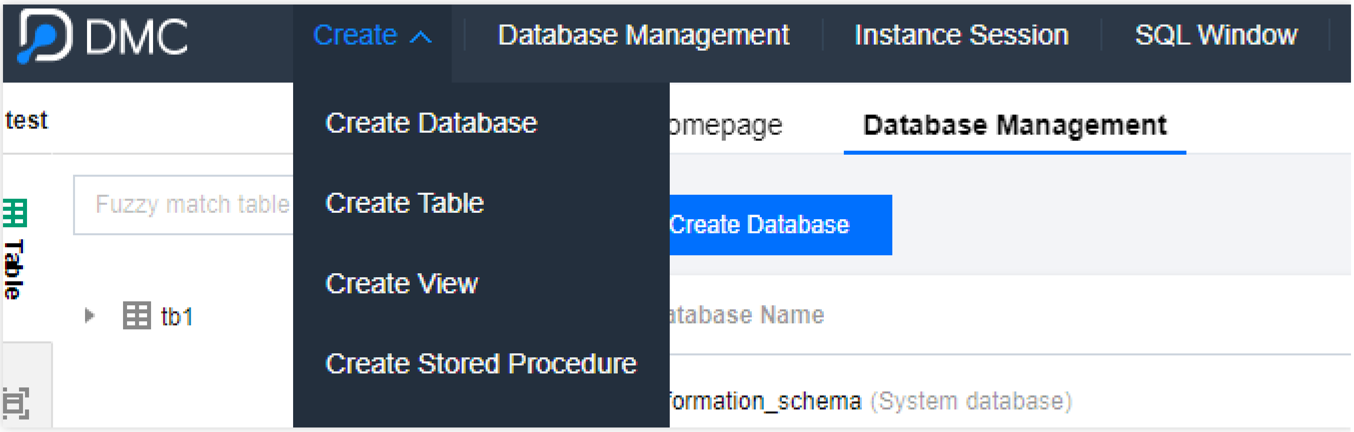
TDSQL-C for MySQL doesn't limit the numbers of databases and tables that can be created. Theoretically, you can create an unlimited number of databases and tables as long as there is enough space. For more information on how to view the upper limit of the storage space, see Product Specifications.

How do I create a database/table?

TDSQL-C for MySQL allows you to create a database/table in many ways:
Quickly create a database/table in the TDSQL-C for MySQL console as instructed in Creating Database.



Create a custom database/table through DMC.



Log in to the client as instructed in Connecting to Cluster and run SQL commands to create a database/table.

The SQL statement to create a database is `create database` , and the command is:

```
create database <database name>;
```

The SQL statement to create a table is `create table`, and the command is:

```
create table <table name> (<table definition option>)<table option><partition optio
The `<table definition option>` is in the format of `column name 1 type 1 [,...] co
```

Example: Create the `tb_emp1` table in the `test_db` database:

```
mysql> USE test_db;
Database changed
mysql> CREATE TABLE tb_emp1
  -> (
  -> id INT(11),
  -> name VARCHAR(25),
  -> deptId INT(11),
  -> salary FLOAT
  -> );
Query OK, 0 rows affected (0.37 sec)
```

Which way is more appropriate to replicate a very big table in a TDSQL-C for MySQL database (for example, to replicate the entire table A to table B)?

You can use the following SQL statement for replication:

```
create table B as select * from A
```

Does TDSQL-C for MySQL support table partitioning?

Yes. You can use partitioned tables to trim large tables to control the volume of data queried and accessed while remaining imperceptible to business code (with no modifications required).

How do I quickly change the structure of a big table?

TDSQL-C for MySQL supports the instant DDL feature to quickly modify columns in big tables while avoiding data replication. This feature can implement changes in seconds without replicating data or using disk capacity or I/O during peak hours. For more information, see Instant DDL Overview.

Performance and Log

Last updated：2023-11-01 17:34:24

How do I test and compare the performance of TDSQL-C for MySQL and TencentDB for MySQL?

Before comparing the performance of TDSQL-C for MySQL and TencentDB for MySQL, note the following to get more accurate and reasonable comparison results:

Use TDSQL-C for MySQL and TencentDB for MySQL instances with the same specification configuration.

Use TDSQL-C for MySQL and TencentDB for MySQL instances on the same version.

The implementation mechanism varies by version. For example, MySQL 8.0 optimizes the performance of multiple CPU cores and separately abstracts threads such as `log_writer` , `log_fluser` , `log_checkpoint` , and `log_write_notifier` , but its performance is lower than MySQL 5.6 or 5.7 when there are only a few CPU cores.

Compare the actual performance in a scenario simulating the pressure in the production environment or use sysbench for comparison. In this way, the obtained test data will be more realistic.

Do not use an individual SQL statement to compare the read performance.

As TDSQL-C for MySQL adopts the compute-storage separation architecture, if only one SQL statement is tested, it may be affected by the network latency, and the read performance will be lower than that of TencentDB for MySQL. The cache hit rate of a database in the production environment is generally above 99%, and only the first read will call I/O, which lowers the read performance. However, as subsequent data is in the cache pool and requires no I/O calls, the performance will become the same.

Do not use an individual SQL statement to compare the write performance either. We recommend that you perform the stress test in a simulated production environment.

For the results of performance comparison between TDSQL-C for MySQL and TencentDB for MySQL, see Test Results - Full Cache Scenario.

How do I prevent some inefficient SQL statements from lowering the overall database performance?

If your TDSQL-C for MySQL cluster is on v8.0, you can use the statement concurrency control feature to limit the number of executions of specified statements.

Does TDSQL-C for MySQL support idle session timeout?

Yes. You can modify the `wait_timeout` parameter to customize the timeout period of idle sessions.

How do I discover and optimize slow SQL statements?

You can discover and optimize slow SQL statements in the following two ways:

Set an alarm policy for the monitoring metric of slow queries on the instance monitoring page to observe slow SQL statements, and then use the slow SQL analysis feature of DBbrain in the console to analyze the performance of such statements and optimize them based on the suggestions provided by DBbrain. For more information, see Slow SQL Analysis.

After connecting to the database cluster, run `show processlist;` to find SQL statements with a long execution time, run `EXPLAIN` to analyze their execution plans and locate root causes, and then optimize them accordingly. For more information on how to connect to a database cluster, see Connecting to Cluster.

Can I improve the query performance of TDSQL-C for MySQL by partitioning tables?

Generally, if a query SQL statement can fall in a partition, its performance can be improved.

Is there any relationship between the compute instance specification and the maximum IOPS? Can I adjust the specification of a compute instance to increase its maximum IOPS?

Yes. For specific compute instance specifications and their maximum IOPS values, see Product Specifications.

How is the IOPS limited and isolated? Will I/O preemption occur between multiple nodes in a TDSQL-C for MySQL cluster?

In a TDSQL-C for MySQL cluster, the IOPS values of nodes are subject to the specification and isolated without affecting each other.

Will the performance be affected after I enable binlog?

After binlog is enabled, the query (SELECT) performance won't be affected, but the write (INSERT, UPDATE, and DELETE) performance will. Generally, in a database with balanced reads/writes, enabling binlog will compromise the performance by 10%–20%.

Will the performance be affected after I enable database audit?

Enabling database audit will compromise the performance by 3%–5%.

What high-speed network protocol does TDSQL-C for MySQL use?

Between TDSQL-C for MySQL compute nodes and storage nodes and between multiple replicas of the stored data, the dual-25 Gbps RDMA technology is leveraged to deliver a low-latency high-throughput I/O performance.

Database Audit

Last updated：2023-07-27 15:39:42

How is audit billed?

Database audit is billed by the stored log size for every clock-hour, and usage duration shorter than one hour will be calculated as one hour. For more information, see Database Audit Billing Overview.

How do I disable database audit after enabling it?

Log in to the TDSQL-C for MySQL console. In the audit instance list, select the target instance, and click **More** > **Disable** in the **Operation** column.

Note：

After it is disabled, the instance's audit policies, log records, and generated log files will be cleared and cannot be recovered. Please save the corresponding logs and files in advance.

How long can audit data be retained?

Audit data can be retained for seven days to five years. You can set the retention period when enabling audit in the TDSQL-C for MySQL console. You can also find the target instance in the audit instance list after enabling database audit, select **More** > **Modify Audit Service** in the **Operation** column, and then modify the audit log retention period.

Between TDSQL-C for MySQL and TDSQL for MySQL

Last updated：2023-11-09 15:34:55

TDSQL-C for MySQL and TDSQL for MySQL are two different database products offered by Tencent Cloud. TDSQL-C for MySQL is designed based on cloud-native architecture, integrating the strengths of traditional databases, cloud computing, and cutting-edge hardware technologies. It supports a wide range of advanced features and capabilities to deliver highly elastic and reliable database services, which features high performance and PB-level storage. While TDSQL for MySQL is designed based on a source-replica architecture. It is a distributed database deployed on Tencent Cloud that supports automatic horizontal sharding and adopts a Shared Nothing architecture, where complete logical databases and tables are obtained and evenly distributed across multiple physical shard nodes in the backend. It provides a complete set of solutions for disaster recovery, backup, recovery, monitoring, and migration, making it suitable for scenarios with massive databases.

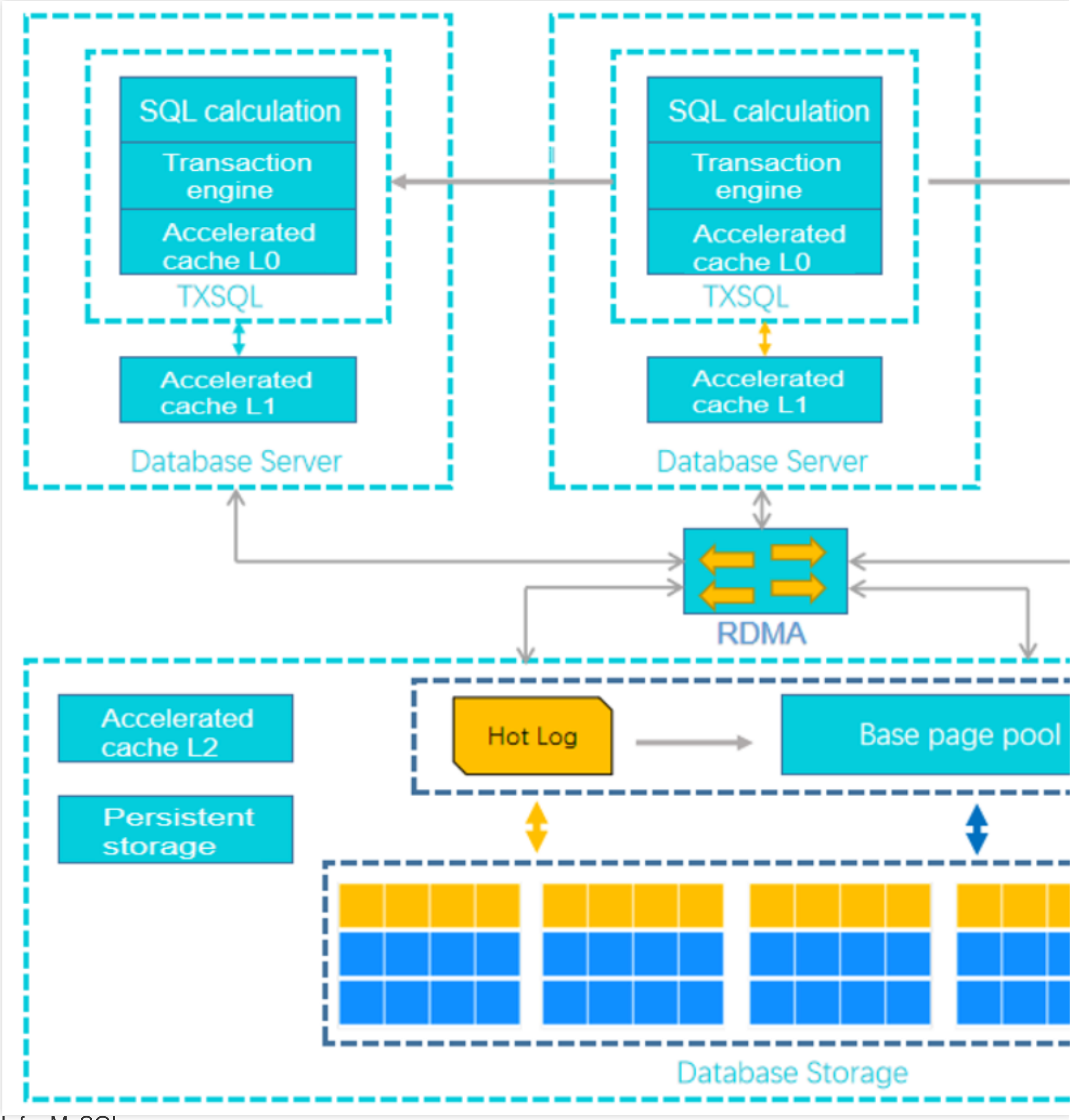
Detailed comparison

Comparison Item	TDSQL-C for MySQL	TDSQL for MySQL
Database type	A new-generation relational database that is born cloud-native	A distributed database that supports automatic horizontal sharding and adopts a Shared Nothing architecture
Architecture	Cluster: A cluster can contain only one read-write instance and up to 15 read-only instances.	InnoDB engine: Physical shard, with one shard containing 1 source and 1 replica or 1 source and 2 replicas
Engine	InnoDB	InnoDB TDStore
Version	MySQL 5.7 MySQL 8.0	MySQL 8.0 Percona 5.7 MariaDB 10.1
Application scenarios	Businesses with great changes, frequent scaling, or needs of read-only instances to improve the read performance. Game projects with frequent quick rollback required Storage and access of petabytes of data	Ecommerce, finance, O2O, social applications, retail, SaaS service providers Online business systems with a large data volume

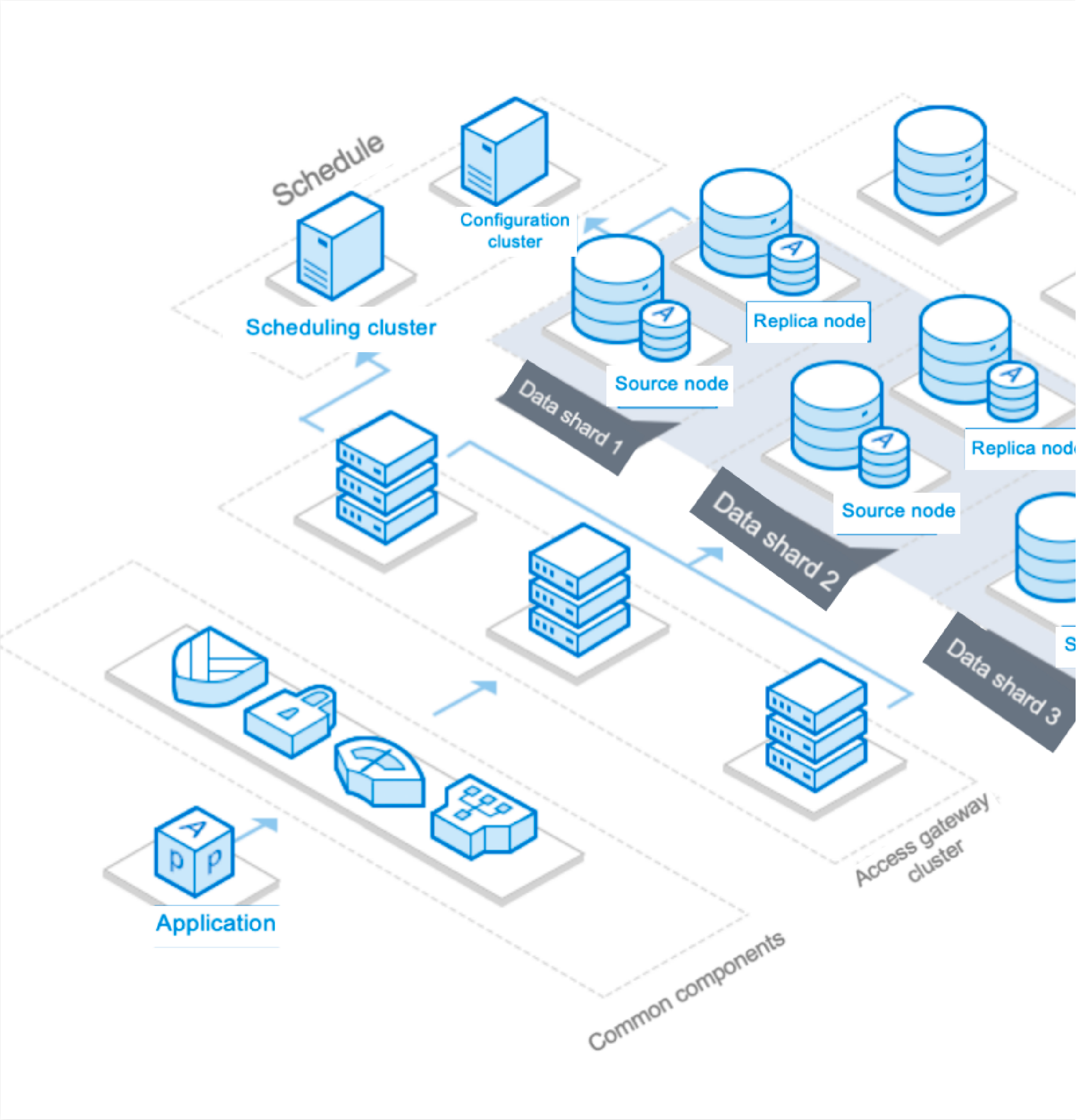
	Scenarios requiring a high write QPS Scenarios that are sensitive to source-replica delay	
Specification	Up to 88 cores and 710 GB memory per instance	Up to 32 cores and 128 GB memory per shard
Automatic backup	The default retention time is 7 days, and the maximum retention time can be set to 1830 days.	The default retention time is 7 days, and the maximum retention time can be set to 365 days.
Manual backup	Supported	Unsupported
Backup file format	Logical Snapshot	Logical Physical
Serverless	Supports serverless elastic expansion without manual intervention, and scales up or down automatically and quickly.	Unsupported
Maximum of creatable tables	There are no limits on the number of databases or tables that can be created. In theory, as long as there is enough space, more databases and tables can be created.	The maximum number of creatable tables is 5000 by default.
Read/write separation	Supported	Supported

Architecture comparison

TDSQL-C for MySQL



TDSQL for MySQL


Schedule
Configuration cluster
Scheduling cluster
Replica node
Source node
Data shard 1
Replica node
Source node
Data shard 2
S
Data shard 3
Application
Access gateway cluster
Common components

Between TDSQL-C for MySQL and TencentDB for MySQL

Last updated：2023-11-09 15:36:04

TDSQL-C for MySQL and TencentDB for MySQL are both relational database products offered by Tencent Cloud. TDSQL-C for MySQL is designed based on cloud-native architecture, integrating the strengths of traditional databases, cloud computing, and cutting-edge hardware technologies. It supports a wide range of advanced features and capabilities to deliver highly elastic and reliable database services, which features high performance and PB-level storage. While TencentDB for MySQL is a high-performance enterprise-grade database service that is specifically tailored based on the open-source MySQL database. It has six enterprise-grade features, including customized kernel, high availability, high reliability, security, scalability, and intelligent Ops.

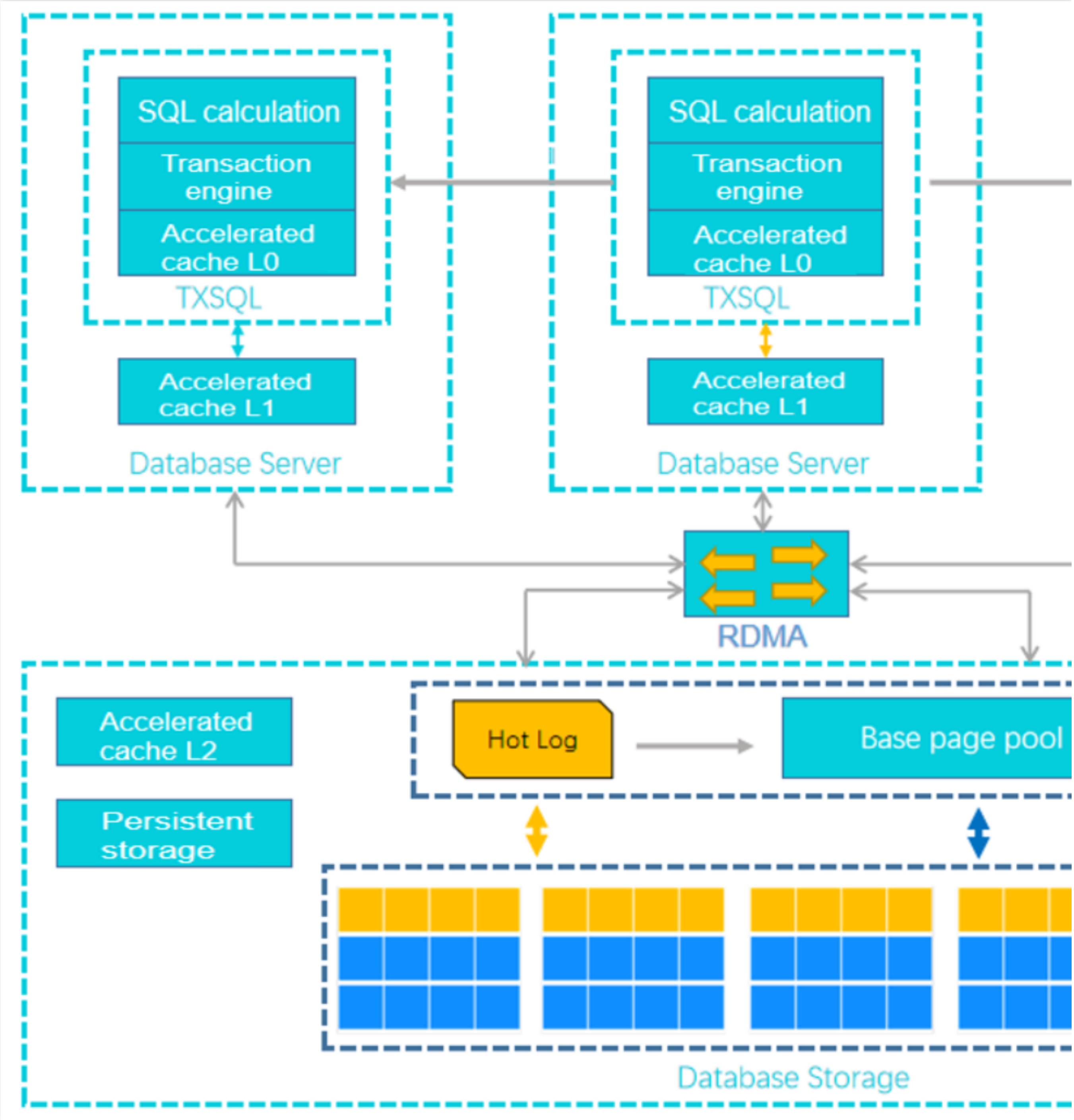
Detailed comparison

Comparison Item	TDSQL-C for MySQL	TencentDB for MySQL
Database type	A new-generation relational database that is born cloud-native	A high-performance enterprise-grade database service that is specifically tailored based on the open-source MySQL database
Architecture	Cluster: A cluster can contain only one read-write instance and up to 15 read-only instances.	Single-node: One single node. Two-node: One source node and one replica node. Three-node: One source node and two replica nodes.
Engine	InnoDB	InnoDB RocksDB
Version	Compatible with MySQL 5.7 Compatible with MySQL 8.0	MySQL 5.6 MySQL 5.7 MySQL 8.0
Application scenarios	Businesses with great changes, frequent scaling, or needs of read-only instances to improve the read performance. Game projects with frequent quick rollback required	Gaming Internet and mobile Apps Finance Ecommerce

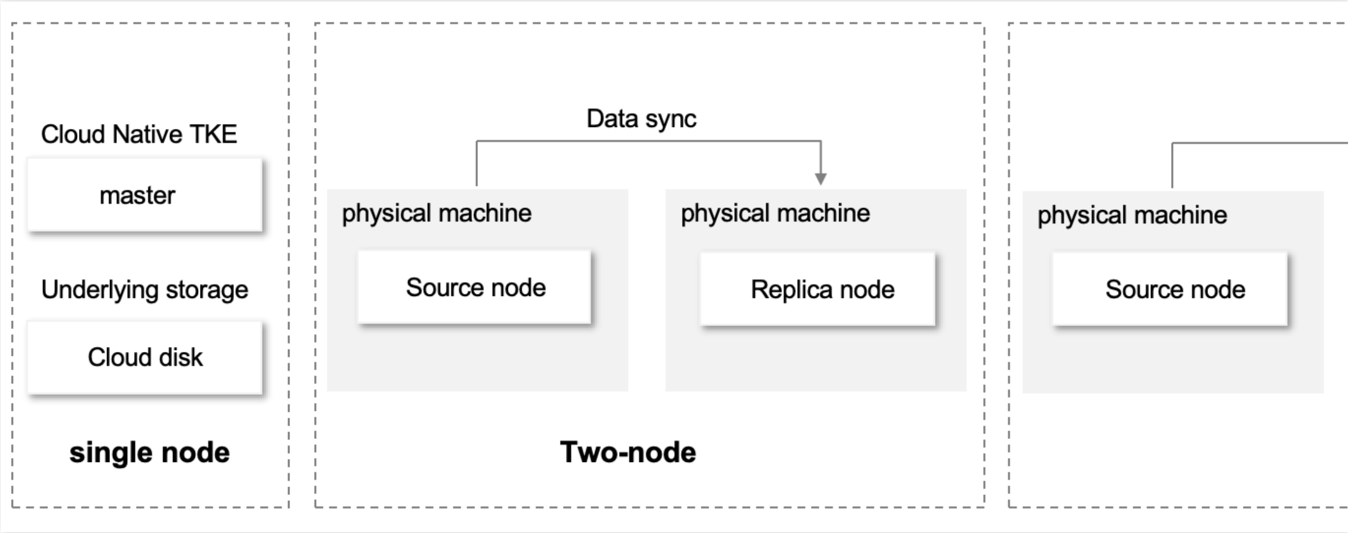
	Storage and access of petabytes of data Scenarios requiring a high write QPS Scenarios that are sensitive to source-replica delay	
Specification	Up to 88 cores and 710 GB memory per instance	Up to 90 cores and 720 GB memory per instance
Automatic backup	The default retention time is 7 days, and the maximum retention time can be set to 1830 days.	The default retention time is 7 days, and the maximum retention time can be set to 1830 days.
Manual backup	Supported	Supported
Backup file format	Logical Snapshot	Logical Physical
Serverless	Supports serverless elastic expansion without manual intervention, and scales up or down automatically and quickly.	Unsupported
Maximum of creatable tables	There are no limits on the number of databases or tables that can be created. In theory, as long as there is enough space, more databases and tables can be created.	The number of tables in a single instance can't exceed 1 million.
Read/write separation	Supported	Supported

Architecture comparison

TDSQL-C for MySQL



TencentDB for MySQL



TDSQL-C for MySQL vs. TencentDB for MySQL

TencentDB for MySQL (Weaknesses)	TDSQL-C for MySQL (Strengths)
The upper limit of data storage is limited by a single physical machine. Upgrading specifications and adding read-only instances takes a relatively long time. Strong data consistency requires expensive methods. Binlog-based data sync can't completely solve the source-replica delay. Full logs (redo log, binlog) and data page updates cause write performance bottlenecks. Data rollback and recovery takes too long.	Compute nodes are stateless, and read-only nodes can be upgraded, switched, and added within seconds. Only redo logs are written to implement lightweight writing, delivering a higher write performance (increased by 140%). The source and replica instances are synced based on redo logs. The source instance pushes redo logs, and the replica instances only replay data in the memory without storing it in disks. The replica instance latency is reduced to milliseconds. Strong data consistency is implemented at the block level based on three copies. Distributed storage is supported with up to 400 TB capacity for each instance. Disk capacity expansion is imperceptible to and has no impact on the business. Gigabytes of data can be backed up/rolled back per second to solve the core problems of slow backup and rollback. Serverless capabilities are available.