

TencentDB for CTSDB

FAQs

Product Documentation



Copyright Notice

©2013-2025 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



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.

FAQs

Last updated : 2025-01-03 10:24:49

How do I connect to CTSDB?

For more information, see [Connecting to Instance](#).

Is CTSDB easy to use?

Yes. In the console, you can view the instance details, perform management and control operations such as initializing and renaming instances, and use the monitoring feature to view the instance health status in real time. You can also use RESTful APIs to perform almost all types of data operations, and such APIs are fully compatible with Elasticsearch protocols.

How does CTSDB ensure high query performance?

It implements an inverted index algorithm to accelerate queries in any dimension.

What are CTSDB's advantages over relational databases?

In scenarios with massive amounts of time series data, a relational database will have the following problems:

High storage costs: Its performance in time series data compression is poor, and it needs to use large amounts of machine resources.

High maintenance costs: It uses a standalone system, where databases and tables need to be partitioned manually at the upper layer, incurring high maintenance costs.

Low write throughput: Its write throughput based on a single server is low, which can hardly withstand the pressure of writing tens of millions of time series data points.

Poor query performance: It is more suitable for transaction processing but has poor performance in aggregate analysis of massive amounts of data.

In contrast, CTSDB has the following advantages:

Low storage costs: Considering that the data is usually incremental in time, has repeated dimensions, and has smooth metric value changes, it uses appropriate encoding and compression algorithms to increase the data compression ratio and aggregates historical data through data rollup to save storage space.

Highly concurrent writes: CTSDB adopts the policy where data is written into the memory first and then periodically dumped as immutable files for storage. It can also write data in bulk to reduce network overheads.

Low query delay and high query concurrency: It optimizes common query modes, uses technologies such as indexing to reduce query delay, and leverages technologies such as caching and routing to improve query concurrency.

How do I connect to CTSDB on a CVM instance over the classic network?

Currently, CTSDB can be accessed only in VPCs. If your CVM instance is on the classic network, you can interconnect the classic network with a VPC through Classiclink to connect to CTSDB.