Generated Date: 2026-10-23

Key-Value Core User Manual

Apusic Key-Value Database

Copyright © Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen 2026. All Rights Reserved.

Copyright Notice

All intellectual property rights including software copyrights involved in this document have been legally registered and duly owned by Kingdee Apusic Cloud Computing Co., Ltd. Such rights are protected by Copyright Law of the People's Republic of China, Regulations on Protection of Computer Software, Regulations on Intellectual Property Protection, relevant international copyright conventions, laws, regulations and other intellectual property legislations. Unauthorized use is strictly prohibited.

Disclaimer

All copyright information contained herein is legally owned by Kingdee Apusic Cloud Computing Co., Ltd. and protected by applicable laws. The Company shall bear no liability for any information unrelated to Kingdee Apusic contained in this document. You may view, copy and print this document only within the scope permitted by Copyright Law of the People's Republic of China. No entity or individual may use, modify or republish any part or content of this document without written authorization from Kingdee Apusic Cloud Computing Co., Ltd. Any violation shall constitute infringement, and the Company reserves the right to pursue legal liabilities accordingly.

Updates to this document shall not be notified separately. You may consult the Company regarding issues related to this document. All rights not expressly granted by the Company are reserved.

Trademark Notice

Apusic is a registered trademark filed with the National Intellectual Property Administration of PRC by Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen. Exclusive rights to the registered trademark are legally owned by Kingdee Apusic and protected by law. Without written consent of Kingdee Apusic, no entity or individual shall use, copy, modify, disseminate, transcribe, or bundle and sell any part of this trademark with other products in any manner or for any reason. Kingdee Apusic shall pursue legal liabilities against any infringement of its trademark rights. All other trademarks or registered trademarks mentioned in this document belong to their respective owners.

Contents

- 1 Preface
  - 1.1 Intended Audience
  - 1.2 Related Documents
  - 1.3 Technical Support
- 2 Introduction
- 3 Related Concepts
- 4 Feature List
- 5 Product Installation
  - 5.1 Redis Configuration Compatibility
- 6 Usage Introduction
  - 6.1 Key-Value Core Usage Guide
    - 6.1.1 Operation Commands
    - 6.1.2 KVDB Cluster Usage Guide
      - 6.1.2.1 Master-Slave Mode
        - 6.1.2.1.1 Master-Slave Commands
      - 6.1.2.2 Sentinel Mode
        - 6.1.2.2.1 Sentinel Commands
      - 6.1.2.3 Cluster Mode
        - 6.1.2.3.1 Cluster Commands
        - 6.1.2.3.2 Cluster Operations Commands
    - 6.1.3 Using SSL
      - 6.1.3.1 SSL Configuration Items
      - 6.1.3.2 Generating Certificates with OPENSSL
      - 6.1.3.3 Starting the Server with SSL
      - 6.1.3.4 Client SSL Connection
    - 6.1.4 Using Chinese National Cryptography
      - 6.1.4.1 Generating Certificates with Tongsuo
      - 6.1.4.2 Starting the Server with Chinese National Cryptography
      - 6.1.4.3 Client SSL Connection
    - 6.1.5 MPTCP Multipath Transmission Protocol
      - 6.1.5.1 MPTCP Configuration Items
      - 6.1.5.2 Configuring the MPTCP Environment

- 6.1.5.3 Starting the Server with MPTCP
- 6.1.5.4 Client MPTCP Connection
- 6.1.6 Data Persistence
- 6.1.6.1 RDB
- 6.1.6.1.1 Ways to Generate RDB Files
- 6.1.6.1.2 Ways to Restore Data from RDB
- 6.1.6.1.3 Repairing Corrupted RDB Files
- 6.1.6.2 AOF
- 6.1.6.2.1 Ways to Generate AOF Files
- 6.1.6.2.2 Ways to Restore AOF Data
- 6.1.6.2.3 Repairing Corrupted AOF Files
- 6.1.7 Command Auditing
- 6.2 kvdb-cli Client Usage Introduction
- 6.2.1 Client Parameters
- 6.2.2 Using the Client
- 6.2.2.1 General Operations
- 6.2.2.2 Statistics Operations
- 6.2.2.3 Query Operations
- 6.2.2.4 Test Operations
- 6.2.2.5 LUA Operations
- 6.2.2.6 Cluster Operations
- 6.3 RDB Cluster Data Migration Tool Usage Introduction
- 6.4 Performance Test Tool Usage Introduction
- 6.4.1 KVDB benchmark Parameters
- 7 All Product Configuration Descriptions
- 7.1 All Key-Value Core Configurations
- 7.1.1 Cache Configuration File
- 7.1.2 Sentinel Configuration File
- 7.1.3 Authorization and Authentication Center Configuration
- 8 Product Tools Usage Guide
- 8.1 kvdb-check-rdb Tool User Manual
- 8.1.1 Tool Location and Prerequisites
- 8.1.1.1 Tool Location
- 8.1.1.2 Prerequisites

- 8.1.2 Basic Syntax
- 8.1.3 Core Options and Parameter Descriptions
- 8.1.4 Common Operation Examples
  - 8.1.4.1 Verifying RDB File Integrity
  - 8.1.4.2 Repairing a Corrupted RDB File
- 8.1.5 Notes
- 8.2 kvdb-check-aof Tool User Manual
  - 8.2.1 Tool Introduction
  - 8.2.2 Tool Location and Prerequisites
    - 8.2.2.1 Tool Location
    - 8.2.2.2 Prerequisites
  - 8.2.3 Basic Syntax
  - 8.2.4 Core Options and Parameter Descriptions
  - 8.2.5 Common Operation Examples
    - 8.2.5.1 Verifying AOF File Integrity
    - 8.2.5.2 Repairing a Corrupted AOF File
  - 8.2.6 Notes
- 8.3 kvdb-benchmark Tool User Manual
  - 8.3.1 Tool Introduction
  - 8.3.2 Tool Location and Prerequisites
    - 8.3.2.1 Tool Location
    - 8.3.2.2 Prerequisites
  - 8.3.3 Basic Syntax
  - 8.3.4 Core Options and Parameter Descriptions
    - 8.3.4.1 Connection Options
    - 8.3.4.2 Core Test Options
    - 8.3.4.3 Advanced Extension Parameters (Multi-threading / Cluster / Special Test Modes)
  - 8.3.5 Common Operation Examples (Core Focus)
    - 8.3.5.1 Local Default Test
    - 8.3.5.2 Testing with Specified Concurrency and Request Counts
    - 8.3.5.3 Targeted Testing of Core Commands
    - 8.3.5.4 Time-Based Stress Testing for Targeted Testing of Core Commands
    - 8.3.5.5 Simplified Output for Quick QPS Viewing
    - 8.3.5.6 Test Result Interpretation (Key Indicators)

- 8.3.6 Notes
- 8.4 kvdb-conf-conv Tool User Manual
  - 8.4.1 Tool Introduction
  - 8.4.2 Tool Location and Prerequisites
    - 8.4.2.1 Tool Location
    - 8.4.2.2 Prerequisites
  - 8.4.3 Basic Syntax
  - 8.4.4 Core Options and Parameter Descriptions
  - 8.4.5 Common Operation Examples
    - 8.4.5.1 Converting KVDB V2.0.2 and Earlier KVDB Server Configuration to Apusic KVDB V2.0
    - 8.4.5.2 Converting KVDB V2.0.2 and Earlier KVDB Sentinel Configuration to Apusic KVDB V2.0
  - 8.4.6 Notes
- 8.5 Troubleshooting Common Problems
  - 8.5.1 Tool Cannot Run
  - 8.5.2 Verification / Repair File Failure
  - 8.5.3 kvdb-cli Cannot Connect to the KVDB Service
- 8.6 Summary
- 9 Solutions to Common Problems
  - 9.1 Problem 1: Port Occupation and Solutions
  - 9.2 Modifying the KVDB Key-Value Core Port
  - 9.3 ACL File Loading
  - 9.4 Resolving KVDB Master-Slave Failover
  - 9.5 Request Latency Issues
  - 9.6 How to Detect Which Commands Were Executed
  - 9.7 Process Is Running but Cannot Connect
  - 9.8 dir Path Issue
  - 9.9 Password Configuration Issues
  - 9.10 Log Configuration Issues
  - 9.11 Master-Slave Buffer Configuration Issues

1 Preface

This document is the user manual for the Apusic Key-Value Database (KVDB) V2.0 key-value core product. It describes in detail how to use and configure the features of the KVDB key-value core, as well as management operations.

1.1 Intended Audience

This document is intended for KVDB product operations engineers, IT system operations engineers, development engineers, and other personnel.

1.2 Related Documents

For more information about Apusic KVDB V2.0, please refer to the following Apusic KVDB V2.0 product manual set:

No.	Manual	Description
1	Apusic Key-Value Database V2.0 Quick Start Guide	Briefly introduces how to get started with KVDB quickly.
2	Apusic Key-Value Database V2.0 Installation Guide	Describes in detail how to install KVDB on various operating systems, how to start and stop KVDB services, and the product registration process.
3	Apusic Key-Value Database V2.0 Key-Value Core User Manual	Describes in detail the use, configuration, and management of KVDB features, as well as how to use the supporting tools.
4	Apusic Key-Value Database V2.0 Console User Manual	Describes in detail the use of KVDB console features and operating instructions.
5	Apusic Key-Value Database V2.0 Development Guide	Describes in detail how to develop KVDB client applications in various programming languages.
6	Apusic Key-Value Database V2.0 Migration Guide	Describes in detail how to migrate and upgrade earlier KVDB versions to V2.0, and how to migrate from Redis to KVDB.
7	Apusic Key-Value Database V2.0 Operations Guide	Describes in detail KVDB monitoring, operations, security hardening, and other operations topics.
8	Apusic Key-Value Database V2.0 Performance Tuning Guide	Describes in detail how to tune KVDB performance.

1.3 Technical Support

KVDB provides comprehensive technical support services. You can obtain technical support in the following ways:

- Website: www.apusic.com
- Tel: 400-855-5800
- Email: support@apusic.com
- Kingdee Cloud Community: https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN

When contacting technical support, please provide the following information:

1. Your name
2. Company information and contact details
3. Operating system and its version
4. Product version number
5. Detailed information such as logs and screenshots of exceptions and errors

2 Introduction

Apusic Key-Value Database (`KVDB`) is a key-value database software suitable for all scenarios, with high throughput and data security, providing secure and reliable caching support for large-scale, high-concurrency, and highly available critical applications. The product is compatible with the Redis protocol and persistent data files, and can replace Redis simply, quickly, and smoothly.

3 Related Concepts

Term	Meaning	Usage Notes
Standalone	Stores data, handles data reads/writes, and handles data synchronization	The default mode of KVDB is standalone mode, in which there is only one node
Master-slave	Replicates data from the master node, handles data reads/writes, and does not participate in data synchronization	A real-time replica of the master node's data. It actively connects to the specified master node and receives the data stream sent by the master node, thereby keeping its data synchronized with the master node. By default, replica nodes are read-only, and all write operations must be sent to the master node.
Sentinel	Monitors master and slave nodes, handles master node failover, and handles master-slave node synchronization	Sentinel mode. When the master node goes down, it automatically switches a slave node to master; when a slave node fails, the sentinel marks it as offline.
Cluster	A computer service system composed of multiple independent computers	In this document, all three modes — master-slave, sentinel, and cluster — can be regarded as clusters; the cluster mode specifically refers to the multi-master-node data sharding storage mode.

4 Feature List

Feature	Description	Additional Notes
Multi-data-type caching	Provides data caching types: string, list, hash, set, sortedset, geo, hyperloglog, and stream.	Supports data structures compatible with the Redis protocol; can directly replace Redis without modifying application code
Publish/Subscribe	Implements the publish/subscribe feature, enhancing system functionality	Supports channel subscription and pattern subscription; can be used in scenarios such as message queues and event notifications
ACL permission control	Provides a secure access mechanism for the service, supporting fine-grained access permission control.	Role-based permission management, supporting command-level and keyspace-level access control
Memory data eviction policies	Provides multiple data eviction policies to meet various data eviction requirements and improve memory utilization.	Includes multiple eviction algorithms such as LRU, LFU, and TTL, which can be flexibly selected according to business characteristics
Persistence	Improves key-value core availability and prevents data loss in the event of downtime.	Supports two persistence methods: RDB snapshots and AOF logs, ensuring data security
Lua script support	Supports operating the key-value core with Lua scripts.	Supports atomic operations of complex logic, improving processing efficiency
Multi-threading mode	Supports multi-threaded concurrent request processing to improve system throughput.	Adopts I/O multiplexing technology to fully utilize multi-core CPU performance
Master-slave mode	Supports master-slave backup.	Implements data redundancy, supports read-write splitting, and improves system availability
Sentinel mode	Provides node monitoring, automatic failover, failure notification, and configuration propagation for master-slave mode.	Implements a high-availability architecture that automatically detects failures and switches services
Cluster mode	Supports elastic scaling (memory scale-out/scale-in) and also provides failover functionality.	Data sharding storage, supporting online scale-out and scale-in, meeting the needs of large data volume scenarios

MPTCP multipath transmission protocol	Supports multipath parallel data transmission, aggregating multiple network links to send and receive data simultaneously, breaking through the bandwidth limit of a single link and achieving redundant fault tolerance of transmission links.	Suitable for large-file and large-batch data synchronization and transmission scenarios. Compared with traditional single-TCP-link transmission, performance is improved by about 70%. It can effectively avoid single-link network jitter, congestion, and interruption problems, greatly improving the throughput, stability, and reliability of data transmission.
Monitoring and operations	Performance monitoring	Provides detailed performance indicator monitoring, including QPS, response time, memory usage, etc.
	Slow query log	Records commands whose execution time exceeds the threshold, facilitating performance tuning
	Statistics	Obtains server running status and statistics through the INFO command
High availability	Automatic failure recovery	Automatically detects node health status and implements automatic failover
	Data consistency	Ensures data consistency between master and slave nodes
	Service discovery	Supports automatic discovery of cluster nodes, simplifying configuration management
Security features	Access control	Supports password authentication and SSL/TLS/Chinese national cryptography encrypted transmission
	Audit log	Records user operation logs to meet compliance requirements
Extended features	Transaction support	Supports transaction commands such as MULTI/EXEC
	Pipeline operations	Supports batch command execution, reducing network round-trip time
	Key expiration management	Supports key lifecycle management features such as TTL and EXPIRE

Development support	Client compatibility	Fully compatible with Redis client libraries (Jedis, Lettuce, etc.)
	Protocol compatibility	Supports the RESP protocol and can seamlessly replace Redis
	Toolchain support	Provides CLI tools, benchmark tools, etc.

5 Product Installation

For KVDB key-value core installation, please refer to the Apusic Key-Value Database V2.0 Installation Guide

5.1 Redis Configuration Compatibility

Starting from KVDB v2.0.2, Redis configuration files are compatible; KVDB automatically converts the contents of Redis.conf into KVDB configuration contents.

Usage: `./kvdb-server redis.conf (specify the Redis configuration file)`

6 Usage Introduction

6.1 Key-Value Core Usage Guide

Distributed caching is KVDB's most core capability and the center of the entire product; all other features are built on top of the data caching business. KVDB stores data directly in memory and uses multi-threaded read-write separation to achieve efficient storage, meeting different types of data storage needs, enabling rapid development, and reducing type conversion; it supports multiple data eviction policies to make rational use of memory space.

6.1.1 Operation Commands

Command	Description
ACL LOAD	Reloads the ACL from the configured ACL file
ACL SAVE	Saves the current ACL rules to the configured ACL file
ACL LIST	Lists the current ACL rules in ACL config file format
ACL USERS	Lists the usernames of all configured ACL rules
ACL GETUSER username	Gets the rules of a specific ACL user
ACL SETUSER username [rule [rule ...]]	Modifies or creates the rules of a specific ACL user
ACL DELUSER username [username ...]	Deletes the specified ACL users and related rules
ACL CAT [categoryname]	Lists the ACL categories or the commands inside a category
ACL GENPASS [bits]	Generates a pseudo-random secure password for ACL users
ACL WHOAMI	Returns the name of the user associated with the current connection
ACL LOG [count or RESET]	Lists the latest ACL security events
ACL HELP	Shows helpful text about the different subcommands
APPEND key value	Appends a value to a key
ASKING	Sent by a cluster client after an -ask redirection

AUTH [username] password	Authenticates to the server
BGREWRITEAOF	Asynchronously rewrites the append-only file
BGSAVE [SCHEDULE]	Asynchronously saves the dataset to disk
BITCOUNT key [start end [BYTE|BIT]]	Counts the set bits in a string
BITFIELD key [GET encoding offset] [SET encoding offset value] [INCRBY encoding offset increment] [OVERFLOW WRAP|SAT|FAIL]	Performs arbitrary bitfield integer operations on strings
BITFIELD_RO key GET encoding offset	Performs arbitrary bitfield integer operations on strings; the read-only variant is disabled
BITOP operation destkey key [key ...]	Performs bitwise operations between strings
BITPOS key bit [start [end [BYTE|BIT]]]	Finds the first bit set or clear in a string
BLPOP key [key ...] timeout	Removes and returns the first element in a blocking manner
BRPOP key [key ...] timeout	Removes and returns the last element in a blocking manner
BRPOPLPUSH source target timeout	Takes the last element from a list and inserts it at the head of another list; if the list has no elements, it blocks until the wait times out or a poppable element is found.
BLMOVE source destination FROM-TOP|FROM-BOTTOM TO-TOP| TO-BOTTOM timeout	Blocking variant that returns and removes the first or last element stored in the source list (head or tail depends on the wherefrom parameter), and pushes that element to the first or last position of the destination list (head or tail depends on the whereto parameter)
LMPOP numkeys key [key ...] LEFT|RIGHT [COUNT count]	Pops one or more elements from the first non-empty list key among the provided key names
BLMPOP timeout numkeys key [key ...] LEFT|RIGHT [COUNT count]	Blocking variant that pops one or more elements from the first non-empty list key among the provided key names, or blocks the connection until another client pushes to it or the timeout is reached

BZPOPMIN key [key ...] timeout	Blocking variant that removes and returns the member with the lowest score from one or more sorted sets, or blocks the connection until another client pushes to it or the timeout is reached
BZPOPMAX key [key ...] timeout	Blocking variant that removes and returns the member with the highest score from one or more sorted sets, or blocks the connection until another client pushes to it or the timeout is reached
CLIENT CACHING YES|NO	Instructs the server whether to track keys in the next request
CLIENT ID	Returns the client ID of the current connection
CLIENT INFO	Returns information about the current client connection
CLIENT KILL [ip:port] [ID client-id] [TYPE normal|master|slave|pubsub] [USER username] [ADDR ip:port] [LADDR ip:port] [SKIPME yes/no]	Kills the connection of a client
CLIENT LIST [TYPE normal|master|replica|pubsub] [ID client-id [client-id ...]]	Gets the list of client connections
CLIENT GETNAME	Gets the current connection name
CLIENT GETREDIR	Gets the tracking notification redirection client ID (if any)
CLIENT UNPAUSE	Resumes processing of paused clients
CLIENT PAUSE timeout [WRITE|ALL]	Stops processing commands from clients for a period of time
CLIENT REPLY ON|OFF|SKIP	Instructs the server whether to reply to commands
CLIENT SETNAME connection-name	Sets the current connection name
CLIENT TRACKING ON|OFF [REDIRECT client-id] [PREFIX prefix [PREFIX prefix ...]] [BCAST] [OPTIN] [OPTOUT] [NOLOOP]	Enables or disables server-assisted client-side caching support
CLIENT TRACKINGINFO	Returns information about server-assisted client-side caching for the current connection
CLIENT UNBLOCK client-id [TIMEOUT|ERROR]	Unblocks a client blocked in a blocking command

	from a different connection
COMMAND	Gets an array of Redis command details
COMMAND COUNT	Gets the total number of Redis commands
COMMAND GETKEYS	Extracts the keys given a full Redis command
COMMAND INFO command-name [command-name ...]	Gets an array of details of specific Redis commands
CONFIG GET parameter [parameter ...]	Gets the values of configuration parameters
CONFIG REWRITE	Rewrites the configuration file with the in-memory configuration
CONFIG SET parameter value [parameter value ...]	Sets configuration parameters to the given values
CONFIG RESETSTAT	Resets the statistics returned by INFO
COPY source destination [DB destination-db] [REPLACE]	Copies a key
DBSIZE	Returns the number of keys in the selected database
DEBUG OBJECT key	Gets debugging information about a key
DEBUG SEGFAULT	Crashes the server
DECR key	Decrements the integer value of a key by one
DECRBY key decrement	Decrements the integer value of a key by the given number
DEL key [key ...]	Deletes a key
DISCARD	Discards all commands issued after MULTI
DUMP key	Returns a serialized version of the value stored at the specified key
DELEX key	Conditionally deletes a key based on the comparison result of the value (value equal/not equal, hash digest match/mismatch)
DIGEST key	Computes the digest of the value corresponding to the key (SM3/SHA256, etc.)

ECHO message	Echoes the given string
EVAL script numkeys [key [key ...]] [arg [arg ...]]	Executes a Lua script server-side
EVALSHA sha1 numkeys [key [key ...]] [arg [arg ...]]	Executes a Lua script server-side
EXEC	Executes MULTI
EXISTS key [key ...]	Determines whether a key exists
EXPIRE key seconds [NX|XX|GT|LT]	Sets a key's time to live in seconds
EXPIREAT key timestamp [NX|XX|GT|LT]	Sets the expiration time of a key as a UNIX timestamp
EXPIRETIME key	Gets the expiration Unix timestamp of a key
FAILOVER [TO host port [FORCE]] [ABORT] [TIMEOUT milliseconds]	Starts a coordinated failover between this server and one of its replicas
FLUSHALL [ASYNC|SYNC]	Removes all keys from all databases
FLUSHDB [ASYNC|SYNC]	Removes all keys from the current database
FUNCTION LOAD	Loads a server-side function (replacing Lua scripts)
FUNCTION DELETE	Deletes the specified function
FUNCTION FLUSH	Flushes all functions
FUNCTION LIST	Lists all functions
FUNCTION STATS	Views function execution statistics
GEOADD key [NX|XX] [CH] longitude latitude member [longitude latitude member ...]	Adds one or more geospatial items to the geospatial index represented by a sorted set
GEOHASH key member [member ...]	Returns members of a geospatial index as standard geohash strings
GEOPOS key member [member ...]	Returns the longitude and latitude of geospatial index members
GEODIST key member1 member2 [m|km|ft|mi]	Returns the distance between two members of a geospatial index
GEORADIUS key longitude latitude radius m|km|ft|mi [WITHCOORD] [WITHDIST] [WITHHASH] [COUNT count [ANY]] [ASC|DESC] [STORE key] [STOREDIST key]	Queries a sorted set representing a geospatial index for members matching a given maximum distance from a point

GEORADIUSBYMEMBER key member radius m|km|ft|mi [WITHCOORD] [WITHDIST] [WITHHASH] [COUNT count [ANY]] [ASC|DESC] [STORE key] [STOREDIST key]	Queries a sorted set representing a geospatial index for members matching a given maximum distance from a member
GEOSEARCH key [FROMMEMBER member] [FROMLONLAT longitude latitude] [BYRADIUS radius m|km|ft|mi] [BYBOX width height m|km|ft|mi] [ASC|DESC] [COUNT count [ANY]] [WITHCOORD] [WITHDIST] [WITHHASH]	Queries a sorted set representing a geospatial index for members within a box or circle area
GEOSEARCHSTORE destination source [FROMMEMBER member] [FROMLONLAT longitude latitude] [BYRADIUS radius m|km|ft|mi] [BYBOX width height m|km|ft|mi] [ASC|DESC] [COUNT count [ANY]] [STOREDIST]	Queries a sorted set representing a geospatial index for members within a box or circle area, and stores the result in another key
GET key	Gets the value of a key
GETBIT key offset	Returns the value stored at key
GETDEL key	The bit value at the offset in the string value; gets the value of a key and deletes the key
GETRANGE key start end	Gets a substring of the string stored at a key
GETSET key value	Sets the string value of a key and returns its old value
HDEL key field [field ...]	Deletes one or more hash fields
HELLO [protover [AUTH username password] [SETNAME clientname]]	Handshakes with Redis
HEXISTS key field	Determines whether a hash field exists
HGET key field	Gets the value of a hash field
HGETALL key	Gets all fields and values in a hash
HINCRBY key field increment	Increments the integer value of a hash field by the given number
HINCRBYFLOAT key field increment	Increments the float value of a hash field by the given amount
HKEYS key	Gets all fields in a hash
HLEN key	Gets the number of fields in a hash

HMGET key field [field ...]	Gets the values of all given hash fields
HMSET key field value [field value ...]	Sets multiple hash fields to multiple values
HSET key field value [field value ...]	Sets the string value of a hash field
HSETNX key field value	Sets the value of a hash field, only if the field does not exist
HRANDFIELD key [count [WITHVALUES]]	Gets one or more random fields from a hash
HSTRLEN key field	Gets the length of the value of a hash field
HVALS key	Gets all values in a hash
HOTKEYS [[value] [opt] ...]	Detects and lists current hot keys
INCR key	Increments the integer value of a key by one
INCRBY key increment	Increments the integer value of a key by the given amount
INCRBYFLOAT key increment	Increments the float value of a key by the given amount
INFO [section]	Gets information and statistics about the server
LOLWUT [VERSION version]	Displays some computer art and the Redis version
KEYS pattern	Finds all keys matching the given pattern
LASTSAVE	Gets the UNIX timestamp of the last successful save to disk
LINDEX key index	Gets an element from a list by its index
LINSERT key BEFORE|AFTER pivot element	Inserts an element before or after another element in a list
LLEN key	Gets the length of a list
LPOP key [count]	Removes and gets the first element in a list
LPOS key element [RANK rank] [COUNT num-matches] [MAXLEN len]	Returns the index of matching elements in a list
LPUSH key element [element ...]	Adds one or more elements to a list
LPUSHX key element [element ...]	Adds an element to a list, only if the list exists

LRANGE key start stop	Gets a range of elements from a list
LREM key count element	Removes elements from a list
LSET key index element	Sets the value of an element in a list by its index
LTRIM key start stop	Trims a list to the specified range
MEMORY DOCTOR	Outputs a memory problems report
MEMORY HELP	Shows helpful text about the different subcommands
MEMORY MALLOC-STATS	Shows allocator internal statistics
MEMORY PURGE	Asks the allocator to release memory
MEMORY STATS	Shows memory usage details
MEMORY USAGE key [SAMPLES count]	Estimates the memory usage of a key
MGET key [key ...]	Gets the values of all given keys
MIGRATE host port key| destination-db timeout [COPY] [REPLACE] [AUTH password] [AUTH2 username password] [KEYS key [key ...]]	Atomically transfers a key from a Redis instance to another one
MONITOR	Listens in real time to all requests received by the server
MOVE key db	Moves a key to another database
MSET key value [key value ...]	Sets multiple keys to multiple values
MSETNX key value [key value ...]	Sets multiple keys to multiple values, only if none of the keys exist
MULTI	Marks the start of a transaction block
OBJECT ENCODING key	Inspects the internal encoding of a Redis object
OBJECT FREQ key	Gets the logarithmic access frequency counter of a Redis object
OBJECT IDLETIME key	Gets the time since the last access to a Redis object
OBJECT REFCOUNT key	Gets the number of references of the value of a key
OBJECT HELP	Shows helpful text about the different

	subcommands
PERSIST key	Removes the expiration time from a key
PEXPIRE key milliseconds [NX|XX|GT|LT]	Sets a key's time to live in milliseconds
PEXPIREAT key milliseconds-timestamp [NX|XX|GT|LT]	Sets the expiration time of a key as a UNIX timestamp specified in milliseconds
PEXPIRETIME key	Gets the expiration Unix timestamp of a key in milliseconds
PFADD key [element [element ...]]	Adds the specified elements to the specified HyperLogLog.
PFCOUNT key [key ...]	Returns the approximated cardinality of the set(s) observed by the HyperLogLog at key(s)
PFMERGE destkey sourcekey [sourcekey ...]	Merges N different HyperLogLogs into a single one
PING [message]	Pings the server
PSETEX key milliseconds value	Sets the value and expiration time of a key in milliseconds
PSUBSCRIBE pattern [pattern ...]	Listens for messages published to channels matching the given patterns
PUBSUB CHANNELS [pattern]	Lists active channels
PUBSUB NUMPAT	Gets the count of unique pattern subscriptions
PUBSUB NUMSUB [channel [channel ...]]	Gets the number of subscribers of channels
PUBSUB HELP	Shows helpful text about the different subcommands
PTTL key	Gets the time to live of a key in milliseconds
PUBLISH channel message	Publishes a message to a channel
PUNSUBSCRIBE [pattern [pattern ...]]	Stops listening for messages published to channels matching the given patterns
QUIT	Closes the connection
RANDOMKEY	Returns a random key from the keyspace
READONLY	Enables read queries for a connection to a cluster

	replica node
READWRITE	Disables read queries for a connection to a cluster replica node
RENAME key newkey	Renames a key
RENAMENX key newkey	Renames a key, only if the new key does not exist
RESET	Resets the connection
RESTORE key ttl serialized-value [REPLACE] [ABSTTL] [IDLETIME seconds] [FREQ frequency]	Creates a key using the provided serialized value, previously obtained with DUMP
ROLE	Returns the role of the instance in the context of replication
RPOP key [count]	Removes and gets the last element in a list
RPOPLPUSH source destination	Removes the last element in a list, prepends it to another list and returns it
LMOVE source destination LEFT|RIGHT LEFT|RIGHT	Pops an element from a list, pushes it to another list and returns it
RPUSH key element [element ...]	Appends one or more elements to a list
RPUSHX key element [element ...]	Appends an element to a list, only if the list exists
SADD key member [member ...]	Adds one or more members to a set
SAVE	Synchronously saves the dataset to disk
SCARD key	Gets the number of members in a set
SCRIPT DEBUG YES|SYNC|NO	Sets the debug mode for executed scripts
SCRIPT EXISTS sha1 [sha1 ...]	Checks existence of scripts in the script cache
SCRIPT FLUSH [ASYNC|SYNC]	Removes all scripts from the script cache
SCRIPT KILL	Kills the currently executing script
SCRIPT LOAD script	Loads the specified Lua script into the script cache
SDIFF key [key ...]	Subtracts multiple sets
SDIFFSTORE destination key [key ...]	Subtracts multiple sets and stores the resulting set in a key

SELECT index	Changes the selected database for the current connection
SET key value [EX seconds|PX milliseconds|EXAT timestamp|PXAT milliseconds-timestamp|KEEPTTL] [NX|XX] [GET]	Sets the string value of a key
SETBIT key offset value	Sets or clears the bit stored at key
SETEX key seconds value	Sets the value and expiration time of a key
SETNX key value	Sets the value of a key, only if the key does not exist
SETRANGE key offset value	Overwrites part of a string at the key starting from the specified offset
SHUTDOWN [NOSAVE|SAVE]	Synchronously saves the dataset to disk, then shuts down the server
SINTER key [key ...]	Returns the members of the set resulting from the intersection of all given sets
SINTERCARD numkeys key [key ...] [LIMIT limit]	Intersects multiple sets and returns the cardinality of the result
SINTERSTORE destination key [key ...]	Intersects multiple sets and stores the resulting set in a key
SISMEMBER key member [member ...]	Returns whether each member is a member of the set stored at key
REPLICAOF host port	Makes the server a replica of another instance, or promotes it as master
SLOWLOG GET [count]	Gets the entries of the slow log
SLOWLOG LEN	Gets the length of the slow log
SLOWLOG RESET	Clears all entries from the slow log
SLOWLOG HELP	Shows helpful text about the different subcommands
SMEMBERS key	Gets all members
SMOVE source destination member	Moves a member from one set to another

SORT key [BY pattern] [LIMIT offset count] [GET pattern [GET pattern ...]] [ASC|DESC] [ALPHA] [STORE destination]	Sorts the elements in a list, set or sorted set
SORT_RO key [BY pattern] [LIMIT offset count] [GET pattern [GET pattern ...]] [ASC|DESC] [ALPHA]	Sorts the elements in a list, set or sorted set; read-only variant of SORT
SPOP key [count]	Removes and returns one or more random members from a set
SRANDMEMBER key [count]	Gets one or more random members from a set
SREM key member [member ...]	Removes one or more members from a set
STRLEN key	Gets the length of the value stored in a key
SUBSCRIBE channel [channel ...]	Listens for messages published to the given channels
SUNION key [key ...]	Returns the members of the set resulting from the union of all given sets
SUNIONSTORE destination key [key ...]	Returns the union of all given sets and stores the result in the specified key
SWAPDB index1 index2	Swaps two Redis databases
SYNC	Internal command used to initiate a replication stream from the master
PSYNC replicationid offset	Internal command used to initiate a replication stream from the master
TIME	Returns the current server time
TOUCH key [key ...]	Changes the last access time of keys; returns the number of specified existing keys
TTL key	Gets the time to live of a key in seconds
TYPE key	Determines the type stored at key
UNSUBSCRIBE [channel [channel ...]]	Stops listening for messages published to the given channels
UNLINK key [key ...]	Deletes a key asynchronously in another thread; otherwise it is just like DEL, but non-blocking

UNWATCH	Forgets all watched keys
WAIT numreplicas timeout	Waits for the synchronous replication of all write commands sent in the context of the current connection
WATCH key [key ...]	Watches the given keys to determine execution of the MULTI/EXEC block
ZADD key [NX|XX] [GT|LT] [CH] [INCR] score member [score member ...]	Adds one or more members to a sorted set, or updates its score if it already exists
ZCARD key	Gets the number of members in a sorted set
ZCOUNT key min max	Counts the members in a sorted set with scores within the given values
ZDIFF numkeys key [key ...] [WITHSCORES]	Subtracts multiple sorted sets
ZDIFFSTORE destination numkeys key [key ...]	Subtracts multiple sorted sets and stores the resulting sorted set in a new key
ZINCRBY key increment member	Increments the score of a member in a sorted set
ZINTERCARD numkeys key [key ...] [LIMIT limit]	Intersects multiple sorted sets and returns the cardinality of the result
ZINTERSTORE destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX]	Intersects multiple sorted sets and stores the resulting sorted set in a new key
ZLEXCOUNT key min max	Counts the number of members in a sorted set within a given lexicographical range
ZPOPMAX key [count]	Removes and returns the member with the highest score in a sorted set
ZPOPMIN key [count]	Removes and returns the member with the lowest score in a sorted set
ZMPOP numkeys key [key ...] MIN|MAX [COUNT count]	Removes and returns the members with scores in a sorted set
ZRANDMEMBER key [count [WITHSCORES]]	Gets one or more random elements from a sorted set
ZRANGESTORE dst src min max [BYSCORE|BYLEX] [REV] [LIMIT offset count]	Stores a range of members from a sorted set into another key

ZRANGE key min max [BYSCORE|BYLEX] [REV] [LIMIT offset count] [WITHSCORES]	Returns a range of members from a sorted set
ZRANGEBYLEX key min max [LIMIT offset count]	Returns a range of members in a sorted set, ordered lexicographically
ZREVRANGEBYLEX key max min [LIMIT offset count]	Returns a range of members in a sorted set, by lexicographical range, ordered from higher to lower strings
ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count]	Returns a range of members in a sorted set, by score
ZRANK key member	Determines the index of a member in a sorted set
ZREM key member [member ...]	Removes one or more members from a sorted set
ZREMRANGEBYLEX key min max	Removes all members in a sorted set within the given lexicographical range
ZREMRANGEBYRANK key start stop	Removes all members in a sorted set within the given indexes
ZREMRANGEBYSCORE key min max	Removes all members in a sorted set within the given scores
ZREVRANGE key start stop [WITHSCORES]	Returns a range of members in a sorted set, by index, with scores ordered from high to low
ZREVRANGEBYSCORE key max min [WITHSCORES] [LIMIT offset count]	Returns a range of members in a sorted set, by score, with scores ordered from high to low
ZREVRANK key member	Determines the index of a member in a sorted set, with scores ordered from high to low
ZSCORE key member	Gets the score associated with the given member in a sorted set
ZMSCORE key member [member ...]	Gets the scores associated with the given members in a sorted set
ZUNIONSTORE destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX]	Adds multiple sorted sets and stores the resulting sorted set in a new key
SCAN cursor [MATCH pattern] [COUNT count] [TYPE type]	Incrementally iterates the keyspace

SSCAN key cursor [MATCH pattern] [COUNT count]	Incrementally iterates Set elements
HSCAN key cursor [MATCH pattern] [COUNT count]	Incrementally iterates hash fields and associated values
ZSCAN key cursor [MATCH pattern] [COUNT count]	Incrementally iterates sorted set elements and associated scores
XINFO CONSUMERS key groupname	Lists the consumers in a consumer group
XINFO GROUPS key	Lists the consumer groups of a stream
XINFO STREAM key [FULL [COUNT count]]	Gets information about a stream
XINFO HELP	Shows helpful text about the different subcommands
XADD key [NOMKSTREAM] [MAXLEN|MINID [=|~] threshold [LIMIT count]] *|ID field value [field value ...]	Appends a new entry to a stream
XTRIM key MAXLEN|MINID [=|~] threshold [LIMIT count]	Trims the stream to (approximately if '~' is passed) a certain size
XDEL key ID [ID ...]	Removes the specified entries from a stream; returns the number of items actually deleted, which may differ from the number of IDs passed if some IDs do not exist
XRANGE key start end [COUNT count]	Returns a range of elements in a stream, with IDs matching the specified IDs interval
XREVRANGE key end start [COUNT count]	Compared to XRANGE, returns a range of elements in a stream in reverse order (from larger to smaller IDs), with IDs matching the specified IDs interval
XLEN key	Returns the number of entries in a stream
XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]	Returns never-seen elements from multiple streams, with IDs greater than the ones reported by the caller for each stream; can block
XGROUP CREATE key groupname id|$ [MKSTREAM]	Creates a consumer group
XGROUP CREATECONSUMER key groupname consumername	Creates a consumer in a consumer group

XGROUP DELCONSUMER key groupname consumername	Deletes a consumer from a consumer group
XGROUP DESTROY key groupname	Destroys a consumer group
XGROUP SETID key groupname id|$	Sets a consumer group to an arbitrary last delivered ID value
XGROUP HELP	Shows helpful text about the different subcommands
XREADGROUP GROUP group consumer [COUNT count] [BLOCK milliseconds] [NOACK] STREAMS key [key ...] ID [ID ...]	Returns new entries from a stream using a consumer group, or accesses the history of pending entries of a given consumer; can block
XACK key group ID [ID ...]	Marks a pending message as correctly processed, effectively removing it from the pending entries list of the consumer group; the return value of the command is the number of messages successfully acknowledged, that is, the IDs we were actually able to resolve in the PEL
XCLAIM key group consumer min-idle-time ID [ID ...] [IDLE ms] [TIME ms-unix-time] [RETRYCOUNT count] [FORCE] [JUSTID]	In the context of a stream consumer group, this command changes the ownership of a pending message, so that the new owner is the consumer specified as the command argument
XAUTOCLAIM key group consumer min-idle-time start [COUNT count] [JUSTID]	In the context of a stream consumer group, this command automatically changes the ownership of pending messages, so that the new owner is the consumer specified as the command argument
XPENDING key group [[IDLE min-idle-time] start end count [consumer]]	Returns information and entries from the pending entries list of a stream consumer group, that is, messages that were fetched but never acknowledged
XIDMPRECORD key pid iid stream-id	Persists the stream IDMP state

6.1.2 KVDB Cluster Usage Guide

An KVDB cluster is a highly available and scalable cluster. The cluster modes are master-slave mode, sentinel mode, and cluster mode.

6.1.2.1 Master-Slave Mode

KVDB master-slave mode, i.e. (Master-Slave Replication), uses one KVDB instance as the master and the rest as backup machines. The data on the master and backup machines is exactly the same. The master supports various operations such as data writing and reading, while the slaves support data synchronization with the master and reading. When one of the KVDB instances fails, simply access the KVDB key-value core on other nodes.

6.1.2.1.1 MASTER-SLAVE COMMANDS

- Become the slave node of a node: slaveof [ip] [port] (replicaof no one will turn a slave node into a master node).

6.1.2.2 Sentinel Mode

A sentinel is an KVDB key-value core-side program that automatically monitors and handles failover of faulty nodes between KVDB key-value cores. KVDB provides sentinel commands; the sentinel sends commands and waits for responses from the KVDB key-value cores, thereby monitoring multiple running KVDB instances.

6.1.2.2.1 SENTINEL COMMANDS

1. Command to view the status of sentinel:
   info
2. Command to get all master nodes monitored by sentinel:
   sentinel masters
3. Command to get the status information of the master-name master node:
   sentinel master [master-name]
4. Command to get the status information of all slaves under the master-name node:
   sentinel slaves [master-name]
5. Command to get the IP address by node name in sentinel:
   sentinel get-master-addr-by-name [master-name]
6. Command to add a node:
   sentinel monitor [name] [ip] [port] [quorum]
7. Command to reset kvdb nodes matching the specified name:
   sentinel reset [master-name]
8. Command to delete a node:
   sentinel remove [master-name]
9. Command to force a node into subjective down state:
   sentinel failover [master-name]
10. Get information about other sentinel nodes in the sentinel cluster:
    sentinel sentinels
11. Set the access password for the specified master node:
    sentinel set [master-name] auth-pass [password]

6.1.2.3 Cluster Mode

Cluster mode implements elastic scaling of KVDB master nodes. It is mainly used to increase or decrease the memory capacity available to KVDB, meeting the caching needs of business by adding nodes instead of increasing server memory; moreover, cluster mode has automatic failover functionality similar to the sentinel, providing high availability, making it a better choice. In cluster mode, all nodes can communicate with each other and perceive each other's status information. The cluster can automatically assign master and slave nodes, or these nodes can be specified manually.

6.1.2.3.1 CLUSTER COMMANDS

1. Assign new hash slots to the receiving node:
   CLUSTER ADDSLOTS slot [slot ...]
2. This command triggers an increment of the cluster config epoch from the connected node. If the node's config epoch is zero, or less than the cluster's greatest epoch, the epoch will be incremented:
   CLUSTER BUMPEPOCH
3. Returns the number of active failure reports for the given node:
   CLUSTER COUNT-FAILURE-REPORTS node-id
4. Returns the number of local keys in the specified hash slot:
   CLUSTER COUNTKEYSINSLOT slot
5. Sets the hash slots as unbound in the receiving node:
   CLUSTER DELSLOTS slot [slot ...]
6. Forces a replica to perform a manual failover of its master:
   CLUSTER FAILOVER [FORCE|TAKEOVER]
7. Deletes a node's own slots information:
   CLUSTER FLUSHSLOTS
8. Removes a node from the nodes table:
   CLUSTER FORGET node-id
9. Returns local key names in the specified hash slot:
   CLUSTER GETKEYSINSLOT slot count
10. Provides information about the state of KVDB Cluster nodes:
    CLUSTER INFO
11. Returns the hash slot of the specified key:
    CLUSTER KEYSLOT key
12. Forces a node cluster to handshake with another node:
    CLUSTER MEET ip port
13. Returns the node id:
    CLUSTER MYID
14. Gets the cluster configuration of the node:
    CLUSTER NODES
15. Sets the current node as a slave of the specified master node:
    CLUSTER REPLICATE [master-node-id]

16. Atomically migrates a key to another instance:
    MIGRATE [host] [port] [key] [destination-db] [timeout] [COPY] [REPLACE] [KEYS key1 key2...]
17. Sets the state of a slot (importing, exporting, ownership):
    CLUSTER SETSLOT [STATE] []
18. Forces the current node to immediately save the current cluster configuration to the nodes.conf file:
    CLUSTER SAVECONFIG
19. Resets the cluster configuration of the node:
    CLUSTER RESET [HARD|SOFT]
20. Sets the configuration epoch of the node:
    CLUSTER SET-CONFIG-EPOCH [epoch]
21. Returns information about all replica nodes of the specified master node:
    CLUSTER REPLICAS [node-id]
22. Returns the mapping information of all slots in the cluster, including the master node and replica nodes of each slot:
    CLUSTER SLOTS

6.1.2.3.2 CLUSTER OPERATIONS COMMANDS

The following commands are auxiliary tools provided by KVDB for daily cluster operations, executed through kvdb-cli (not Cluster protocol commands).

Command	Description
`kvdb-cli --cluster create ip:port1 ... --cluster-replicas 1`	Create a cluster (automatic meet + slot assignment + master-slave)
`kvdb-cli --cluster check ip:port`	Check cluster health status
`kvdb-cli --cluster info ip:port`	View cluster statistics
`kvdb-cli --cluster fix ip:port`	Fix problems such as uncovered slots (experimental)
`kvdb-cli --cluster reshard ip:port`	Online resharding (migrating slots)
`kvdb-cli --cluster add-node new_ip:new_port existing_host:existing_port`	Add a new node
`kvdb-cli --cluster del-node host:port node_id`	Delete a node

> *Note: These commands are mainly used to simplify cluster creation, checking, maintenance, and expansion operations, and are suitable for administrators performing cluster management. Among them, `--cluster fix` and `--cluster reshard` are advanced features; it is recommended to fully understand their behavior before using them.*

6.1.3 Using SSL

SSL is KVDB's built-in encrypted communication protocol. Enabling SSL-related configuration implements SSL mutual authentication encrypted communication with clients. You can use OPENSSL to generate the corresponding certificate and key files.

SSL configuration exists in both kvdb.yaml and sentinel.yaml, with the same usage. For detailed configuration explanations, refer to the Cache Configuration File section or the following:

6.1.3.1 SSL Configuration Items

Based on the common parameters of TLS configuration, the following is a complete supplementary table:

Parameter	Explanation	Usage
tls-port	SSL listening port. If only SSL listening is enabled, the port under Network needs to be set to 0	port: 6369
tls-cert-file	Server SSL certificate	tls-cert-file: "./certs/ssl_tls_cert/server.crt"
tls-key-file	Key of the server SSL certificate	tls-key-file: "./certs/ssl_tls_cert/server.key"
tls-key-file-pass	Password of the server SSL certificate key	tls-key-file-pass: "your_password"
tls-ca-cert-file	Certificate file of the certificate authority, i.e. the trusted root certificate	tls-ca-cert-file: "./certs/ssl_tls_cert/ca.crt"
tls-ca-cert-dir	Certificate file path of the certificate authority. If there are multiple trusted root certificates, use this parameter	tls-ca-cert-dir: ""
tls-client-cert-file	Client SSL certificate, used for cluster/master-slave mode	tls-client-cert-file: "./certs/ssl_tls_cert/client.crt"

tls-client-key-file	Key of the client SSL certificate, used for cluster/master-slave mode	tls-client-key-file: "./certs/ssl_tls_cert/client.key"
tls-client-key-file-pass	Password of the client SSL certificate key	tls-client-key-file-pass: "your_password"
tls-dh-params-file	Diffie-Hellman parameter file, used for forward secrecy	tls-dh-params-file: "./certs/ssl_tls_cert/dhparam.pem"
tls-protocols	TLS protocol versions, specifying the allowed TLS protocol versions	tls-protocols: "TLSv1.2 TLSv1.3"
tls-ciphers	SSL/TLS encryption cipher suites (for TLSv1.2 and below)	tls-ciphers: "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256"
tls-ciphersuites	TLSv1.3 cipher suites (for TLSv1.3)	tls-ciphersuites: "TLS_CHACHA20_POLY1305_SHA256:TLS_AES_256_GCM_SHA384"
tls-prefer-server-ciphers	Whether to prefer the cipher suites defined by the server rather than those defined by the client	tls-prefer-server-ciphers: "yes"
tls-session-caching	Whether to enable TLS session caching	tls-session-caching: "yes"
tls-session-cache-size	Maximum size of the TLS session cache	tls-session-cache-size: 1000
tls-session-cache-timeout	Timeout of the TLS session cache (seconds)	tls-session-cache-timeout: 300
tls-auth-clients	Whether the server verifies client certificates. By default, clients are required to provide certificates and the server verifies	tls-auth-clients: ""

	them. If not needed, it can be configured as "no"; if set to optional, the client may provide a certificate for verification or not	
tls-replication	Whether master-slave mode uses TLS communication. When this parameter is true on the master, slave nodes must also be true	tls-replication: false
tls-cluster	Whether cluster mode uses TLS communication	tls-cluster: false

6.1.3.2 Generating Certificates with OPENSSL

Other methods can be used to obtain the required certificates, as long as they are legitimate and usable.

For SSL mutual authentication:

- The server needs a CA certificate, server certificate, and server private key;
- The client needs a CA certificate, client certificate, and client private key.

Steps:

1. Download and install openssl: official website /source/index.html (openssl.org)
2. openssl.cnf file:

```
[ server_cert ]
keyUsage = digitalSignature, keyEncipherment
nsCertType = server

[ client_cert ]
keyUsage = digitalSignature, keyEncipherment
nsCertType = client
```

3. Create a CA certificate:
   - `openssl genrsa -out ca.key 4096`
   - `openssl req -x509 -new -nodes -sha256 -key ca.key -days 3650 -subj "/O=APUSIC /CN=KVDB" -out ca.crt`
4. Create the server private key and certificate
   - `openssl genrsa -out kvdb.key 2048`
   - `openssl req -new -sha256 -subj "/O=APUISC /CN=KVDB" -key kvdb.key | openssl x509 -req -CA ca.crt -CAkey ca.key -CAcreateserial -days 365 -extfile openssl.cnf -extensions server_cert -out kvdb.crt`
5. Create the client private key and certificate
   - `openssl genrsa -out client.key 2048`
   - `openssl req -new -sha256 -subj "/O=APUISC /CN=KVDB" -key client.key | openssl x509 -req -CA ca.crt -CAkey ca.key -CAcreateserial -days 365 -extfile openssl.cnf -extensions client_cert -out client.crt`

6.1.3.3 Starting the Server with SSL

1. Modify kvdb.yaml (key section)

```
 TLS/SSL:
   # TLS listening port; use together with port 0 to enable TLS only
   tls-port: 0
   # Server TLS certificate file path (PEM format)
   tls-cert-file: "./cert/kvdb.crt"
   # Server TLS private key file path (PEM format)
   tls-key-file: "./cert/kvdb.key"
   # Password of the TLS private key file (if encrypted)
   tls-key-file-pass: ""
   # Client TLS certificate file path (for client authentication)
   tls-client-cert-file: ""
   # Client TLS private key file path
```

```
    tls-client-key-file: ""
    # Password of the client TLS private key file (if encrypted)
    tls-client-key-file-pass: ""
    # DH parameter file path
    tls-dh-params-file: ""
    # CA certificate file path used to verify clients and nodes
    tls-ca-cert-file: "./cert/cat.crt"
```

2. Start kvdb `./kvdb-server kvdb.yaml`

6.1.3.4 Client SSL Connection

- Connect using kvdb-cli: `./kvdb-cli -p 6369 --tls --cert ./client.crt --key ./client.key --cacert ./ca.crt`
- Connect using Go:

```go
func TestTls(t *testing.T) {
    caCert, err := ioutil.ReadFile("./certs/ca.crt")
    if err != nil {
        panic(err)
    }
    caCertPool := x509.NewCertPool()
    caCertPool.AppendCertsFromPEM(caCert)
    cliCert, err := tls.LoadX509KeyPair("./certs/client.crt", "./certs/client.key")
    if err != nil {
        panic(err)
    }
    cli := redis.NewClient(&redis.Options{
        Addr: "127.0.0.1:6369",
        TLSConfig: &tls.Config{
            Certificates:       []tls.Certificate{cliCert}, //客户端证书，双向认证必须携带
            RootCAs:            caCertPool,                 //校验服务器证书【CA证书】
            InsecureSkipVerify: true,                       //不用校验服务器证书
        },
        DialTimeout: time.Minute,
    })

    fmt.Println(cli.Info("server").String())
}
```

6.1.4 Using Chinese National Cryptography

Chinese national cryptography is KVDB's built-in encrypted communication protocol. Enabling the related configuration implements mutual authentication encrypted communication with clients using Chinese national cryptography. You can use the Tongsuo cryptography library to generate the corresponding certificate and key files.

The Chinese national cryptography encryption configuration applies uniformly to the kvdb.yaml and sentinel.yaml configuration files, and the usage rules of the cryptography parameters in both files are exactly the same. Compared with regular SSL encryption configuration, TLS configuration with Chinese national cryptography requires specifying the corresponding cipher suites according to the protocol version: when using the TLSv1.2 protocol, configure the parameter tls-ciphers: ECC-SM2-SM4-GCM-SM3:ECC-SM2-SM4-CBC-SM3; when using the TLSv1.3 protocol, configure the parameter tls-ciphersuites: TLS_SM4_GCM_SM3, thereby achieving standard full-link encrypted communication with Chinese national cryptography.

6.1.4.1 Generating Certificates with Tongsuo

Tongsuo is a modern open-source cryptography foundation library launched by the OpenAtom open-source community. Its predecessor is BabaSSL. It is compatible with the OpenSSL interface system, purpose-built for domestic commercial cryptography compliance scenarios, and has complete Chinese national cryptographic algorithm and secure communication protocol capabilities. It is currently the mainstream underlying dependency library for TLS Chinese national cryptography transformation of localized middleware.

For mutual authentication with Chinese national cryptography:

- The server needs a CA certificate, server certificate, and server private key;
- The client needs a CA certificate, client certificate, and client private key.

Steps:

1. Download, install, and compile Tongsuo; the cryptography version of openssl is located in the Tongsuo/app directory
2. openssl.cnf file:

```
[ server_cert ]
keyUsage = digitalSignature, keyEncipherment
nsCertType = server

[ client_cert ]
keyUsage = digitalSignature, keyEncipherment
nsCertType = client
```

3. Create a CA certificate:
   - `Tongsuo/app/openssl ecparam -genkey -name SM2 -out ca.key`

- `Tongsuo/app/openssl req -x509 -new -nodes -sm3 -key ca.key -days 3650 -subj "/O=APUSIC /CN=KVDB" -out ca.crt`

4. Create the server private key and certificate

- `Tongsuo/app/openssl ecparam -genkey -name SM2 -out kvdb.key`
- `Tongsuo/app/openssl -new -sm3 -subj "/O=APUISC /CN=KVDB" -key kvdb.key | openssl x509 -req -sm3 -CA ca.crt -CAkey ca.key -CAcreateserial -days 365 -out kvdb.crt`

5. Create the client private key and certificate

- `Tongsuo/app/openssl ecparam -genkey -name SM2 -out client.key`
- `Tongsuo/app/openssl -new -sm3 -subj "/O=APUISC /CN=KVDB" -key client.key | openssl x509 -req -sm3 -CA ca.crt -CAkey ca.key -CAcreateserial -days 365 -extfile openssl.cnf -extensions client_cert -out client.crt`

6.1.4.2 Starting the Server with Chinese National Cryptography

1. Modify kvdb.yaml (key section)

```
 TLS/SSL:
   # TLS listening port; use together with port 0 to enable TLS only
   tls-port: 6369
   # Server TLS certificate file path (PEM format)
   tls-cert-file: "./cert/kvdb.crt"
   # Server TLS private key file path (PEM format)
   tls-key-file: "./cert/kvdb.key"
   # Password of the TLS private key file (if encrypted)
   tls-key-file-pass: ""
   # Client TLS certificate file path (for client authentication)
   tls-client-cert-file: ""
   # Client TLS private key file path
   tls-client-key-file: ""
   # Password of the client TLS private key file (if encrypted)
   tls-client-key-file-pass: ""
```

```
    # DH parameter file path
    tls-dh-params-file: ""
    # CA certificate file path used to verify clients and nodes
    tls-ca-cert-file: "./cert/cat.crt"
    # Cipher suites allowed for TLSv1.3; use TLS_SM4_GCM_SM3 for
 Chinese national cryptography
    tls-ciphersuites: TLS_SM4_GCM_SM3
```

2. Start kvdb `./kvdb-server kvdb.yaml`

6.1.4.3 Client SSL Connection

- Connect using kvdb-cli: `./kvdb-cli --tls -p 6369 --cert ./kvdb.crt --key ./kvdb.key --cacert ./ca.crt --tls-ciphersuites TLS_SM4_GCM_SM3`

6.1.5 MPTCP Multipath Transmission Protocol

Apusic KVDB V2.0 supports the multipath transmission protocol MPTCP. MPTCP was introduced starting from Linux kernel version 5.6. This protocol allows multiple network paths to transmit data simultaneously, thereby improving transmission efficiency.

6.1.5.1 MPTCP Configuration Items

```
# MPTCP multipath transmission control protocol; supports concurrent
multi-link transmission, improving link aggregation bandwidth and
transmission throughput; requires Linux kernel version 5.6 or above
mptcp-enabled: yes
# Whether to enable MPTCP for master-slave replication
repl-mptcp-enabled: yes
```

6.1.5.2 Configuring the MPTCP Environment

Prerequisite: the operating system kernel version must be ≥ Linux 5.6. Linux kernels 5.6 and above natively include the core MPTCP functionality, with no need to manually compile kernel modules; kernels below 5.6 do not have native MPTCP capability and do not support this multipath transmission configuration scheme.

MPTCP multipath transmission relies on link aggregation of multiple physical network adapters. The server needs to be equipped with multiple physical network adapters connected to optical modules to achieve parallel multi-link transmission. The following uses two dual-NIC servers as an example to complete the MPTCP link aggregation configuration on the server side and the client side.

Server device: aggregated NIC IPs 192.168.10.25, 192.168.11.25

```
root@user:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp94s0f0np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
state UP group default qlen 1000
    link/ether 6c:fe:54:26:7a:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.25/24 brd 192.168.10.255 scope global enp94s0f0np0
       valid_lft forever preferred_lft forever
    inet6 fe80::6efe:54ff:fe26:7a70/64 scope link
       valid_lft forever preferred_lft forever
3: enp94s0f1np1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
state UP group default qlen 1000
    link/ether 6c:fe:54:26:7a:71 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.25/24 brd 192.168.11.255 scope global enp94s0f1np1
       valid_lft forever preferred_lft forever
    inet6 fe80::6efe:54ff:fe26:7a71/64 scope link
       valid_lft forever preferred_lft forever
```

Client device: aggregated NIC IPs 192.168.10.26, 192.168.11.26

```
root@user:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp94s0f0np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
```

```
state UP group default qlen 1000
    link/ether 6c:fe:54:22:a9:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.26/24 brd 192.168.10.255 scope global enp94s0f0np0
       valid_lft forever preferred_lft forever
    inet6 fe80::6efe:54ff:fe22:a930/64 scope link
       valid_lft forever preferred_lft forever
3: enp94s0f1np1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
state UP group default qlen 1000
    link/ether 6c:fe:54:22:a9:31 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.26/24 brd 192.168.11.255 scope global enp94s0f1np1
       valid_lft forever preferred_lft forever
    inet6 fe80::6efe:54ff:fe22:a931/64 scope link
       valid_lft forever preferred_lft forever
```

1. Preliminary check commands

```
# Check the kernel version
uname -r

# Confirm MPTCP kernel compilation is enabled (enabled by default in
5.6+)
sysctl net.mptcp.enabled
```

If it returns net.mptcp.enabled = 1, the kernel MPTCP feature is ready; if it is 0, run sysctl -w net.mptcp.enabled=1 to enable it temporarily, and write it into /etc/sysctl.conf for permanent enablement.

2. MPTCP information viewing commands

```
# View the configured MPTCP multi-link endpoint information
ip mptcp endpoint show

# View MPTCP subflow and address access limit parameters
ip mptcp limits show
```

3. Server-side configuration

Configure the signal flag on server NICs: advertises the machine's multiple NIC IP addresses to the outside, so that clients can actively establish multiple subflow links

```
# Bind the two NICs as MPTCP advertised addresses
ip mptcp endpoint add 192.168.10.25 id 1 dev enp94s0f0np0 signal
ip mptcp endpoint add 192.168.11.25 id 2 dev enp94s0f1np1 signal

# Global MPTCP resource limit optimization: maximum 8 concurrent
subflows, accept up to 8 new remote addresses
ip mptcp limits set subflows 8
ip mptcp limits set add_addr_accepted 8
```

4. Client-side configuration

Client subflow: actively creates MPTCP subflows through the local NIC; fullmesh: fully meshed mode, automatically establishes subflows with all IPs advertised by the server to aggregate bandwidth

```
# Full-link aggregation with fullmesh on both NICs; other NICs use
regular subflow mode
ip mptcp endpoint add 192.168.10.26 id 1 dev enp94s0f0np0 subflow
fullmesh
ip mptcp endpoint add 192.168.11.26 id 2 dev enp94s0f1np1 subflow
fullmesh

# Same limit parameters as the server side
ip mptcp limits set subflows 8
ip mptcp limits set add_addr_accepted 8
```

5. MPTCP connectivity & bandwidth test (iperf3 version >= 3.20)

Server side

```
iperf3 -s --mptcp
```

Client side (connecting to any one of the server IPs will automatically aggregate all links)

```
iperf3 -c 192.168.11.25 --mptcp
```

Bandwidth test comparison

- MPTCP not enabled:

```
root@user:~# iperf3 -c 192.168.11.25
Connecting to host 192.168.11.25, port 5201
[  5] local 192.168.11.26 port 51502 connected to 192.168.11.25 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.10 GBytes  9.45 Gbits/sec    0   1.33 MBytes
[  5]   1.00-2.00   sec  1.09 GBytes  9.39 Gbits/sec    0   1.78 MBytes
[  5]   2.00-3.00   sec  1.10 GBytes  9.43 Gbits/sec    0   1.95 MBytes
[  5]   3.00-4.00   sec  1.10 GBytes  9.40 Gbits/sec    0   2.09 MBytes
[  5]   4.00-5.00   sec  1.10 GBytes  9.41 Gbits/sec    0   2.21 MBytes
[  5]   5.00-6.00   sec  1.09 GBytes  9.40 Gbits/sec    0   3.04 MBytes
[  5]   6.00-7.00   sec  1.10 GBytes  9.43 Gbits/sec    0   3.20 MBytes
[  5]   7.00-8.00   sec  1.09 GBytes  9.39 Gbits/sec    0   3.20 MBytes
[  5]   8.00-9.00   sec  1.09 GBytes  9.38 Gbits/sec    0   3.20 MBytes
[  5]   9.00-10.00  sec  1.09 GBytes  9.38 Gbits/sec    0   3.20 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  11.0 GBytes  9.42 Gbits/sec    0             sender
[  5]   0.00-10.00  sec  11.0 GBytes  9.41 Gbits/sec                  receiver

iperf Done.
```

- MPTCP enabled:

```
root@user:~# iperf3 -c 192.168.11.25 --mptcp
Connecting to host 192.168.11.25, port 5201
[  5] local 192.168.11.26 port 24404 connected to 192.168.11.25 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  2.16 GBytes  18.6 Gbits/sec    2   3.70 MBytes
[  5]   1.00-2.00   sec  2.12 GBytes  18.2 Gbits/sec    2   5.86 MBytes
[  5]   2.00-3.00   sec  2.01 GBytes  17.3 Gbits/sec  142   4.55 MBytes
[  5]   3.00-4.00   sec  2.07 GBytes  17.8 Gbits/sec    1   4.81 MBytes
[  5]   4.00-5.00   sec  2.33 GBytes  20.0 Gbits/sec    2   4.82 MBytes
[  5]   5.00-6.00   sec  2.42 GBytes  20.8 Gbits/sec    0   5.21 MBytes
[  5]   6.00-7.00   sec  2.39 GBytes  20.6 Gbits/sec    0   5.21 MBytes
[  5]   7.00-8.00   sec  2.34 GBytes  20.1 Gbits/sec    0   5.21 MBytes
[  5]   8.00-9.00   sec  2.34 GBytes  20.1 Gbits/sec   68   4.04 MBytes
[  5]   9.00-10.00  sec  1.96 GBytes  16.8 Gbits/sec    2   2.94 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  22.6 GBytes  19.4 Gbits/sec  219             sender
[  5]   0.00-10.02  sec  22.6 GBytes  19.3 Gbits/sec                  receiver

iperf Done.
```

6. Parameter mode description

Parameter flag	Applicable side	Description

signal	Server side	Advertises the machine's multiple IP information to the outside, passively waiting for clients to initiate subflows
subflow	Client side	Allows establishing MPTCP subflows from the current NIC egress
fullmesh	Client side	Full-topology networking, automatically establishes subflows with all addresses of the peer side, maximizing link bandwidth aggregation

6.1.5.3 Starting the Server with MPTCP

1. Modify kvdb.yaml (key section)

```
 # MPTCP multipath transmission control protocol; supports concurrent multi-link transmission, improving link aggregation bandwidth and transmission throughput; requires Linux kernel version 5.6 or above
 mptcp-enabled: yes
```

2. Start kvdb `./kvdb-server kvdb.yaml`

6.1.5.4 Client MPTCP Connection

- Connect using kvdb-cli: `./kvdb-cli -p 6359 --mptcp`

6.1.6 Data Persistence

KVDB provides users with two persistence methods: RDB and AOF. The Operation Commands section also contains related operations; here we will elaborate on the use of the two persistence methods and the related tools.

6.1.6.1 RDB

The RDB persistence method stores the data in kvdb to the hard disk in RDB format.

6.1.6.1.1 WAYS TO GENERATE RDB FILES

There are three ways to generate RDB files:

1. Configure the RDB generation conditions with save in the configuration file; when the conditions are triggered, the system automatically uses bgsave to generate an RDB file;
2. Use the save command; after using this command, all requests will be blocked, and request processing resumes after the RDB file is generated;
3. Use the bgsave command; this command does not block requests, but generating the RDB file is slower than save.

6.1.6.1.2 WAYS TO RESTORE DATA FROM RDB

1. Place one or more RDB files in the directory specified by dir in the key-value configuration file; starting kvdb will automatically load the RDB data.

6.1.6.1.3 REPAIRING CORRUPTED RDB FILES

kvdb provides an RDB file repair tool.

Usage: `./kvdb-check-rdb [rdb-filename.rdb]`

6.1.6.2 AOF

The AOF persistence method writes the data in kvdb to a text file in aof format, stored under the appendonlydir folder, and kvdb will append new requests to the text in append mode.

6.1.6.2.1 WAYS TO GENERATE AOF FILES

1. Change AppendOnly to "yes" in the configuration file.

6.1.6.2.2 WAYS TO RESTORE AOF DATA

1. Place one or more AOF files, or the entire appendonlydir folder, in the directory specified by dir in the key-value configuration file; starting kvdb will automatically load the AOF data.

6.1.6.2.3 REPAIRING CORRUPTED AOF FILES

kvdb provides an AOF file repair tool, but the premise of using this tool is that only the last appended file is corrupted before it can be repaired. The reason for this design is that if the base file is already corrupted, it means most of the data has been lost and there is no need to repair it.

Usage: `./kvdb-check-aof --fix [filename.aof|filename.mainfest]`

6.1.7 Command Auditing

KVDB provides the monitor command, which can monitor all request information. In combination with the shell client, the monitored content is output to a specific file to achieve this.

The record format of request information: `timestamp [database_number client_address:client_port] command`

Usage: `kvdb-cli -h [ip] -p [port] [-a password] monitor >kvdb_commands.log`

6.2 kvdb-cli Client Usage Introduction

To make KVDB easy to use, the shell client provides users with a convenient command-line usage method, and offers practical features such as helping to create clusters, manage cluster slots, and perform LRU tests.

6.2.1 Client Parameters

Usage: `kvdb-cli [options] [command [command arguments …]]`

Option	Description
-h [hostname]	IP of the key-value core (default: 127.0.0.1)
-p [port]	Port of the key-value core (default: 6359)
-a [password]	The requirepass password of the key-value core. You can use the KVDBCLI_AUTH environment variable to set and enter the password, which is more secure
--user [username]	Username when logging in with an ACL user
--pass [password]	Password corresponding to the ACL user login
--askpass	Ignores the KVDBCLI_AUTH environment variable and forces the use of the directly entered password
-u [uri]	URI address of the key-value core (compatible with the Redis protocol)
-r [repeat]	Repeats the specified command [repeat] times
-i [interval]	When using -r, sets the interval in seconds between each repetition
-n [db]	Database number (16 databases by default, numbered 0-15)
-3	Switches to the RESP3 protocol
-x	Reads input from stdin as the last argument of kvdb-cli[br/]Example: kvdb-cli -x set key [/opt/file
-d [delimiter]	Delimiter between response blocks in raw format (e.g. \n)
-D [delimiter]	Delimiter between multiple responses in raw format (e.g. \n)
-c	Connects in cluster mode
-e	Returns an error code when command execution fails
--raw	Uses raw format output (when tty is not the default output device)
--no-raw	Forces formatted output even if standard output is not a tty
--quoted-input	Forces input to be treated as a quoted string
--csv	Outputs in CSV format
--show-pushes [yn]	Whether to print resp3 push messages; enabled by default
--stat	Dynamically prints key-value core status information: memory/number of client connections, etc.

--lru-test	Simulates a cache workload with an 80-20 distribution (key usage)
--replica	Simulates a backup node showing the commands received from the master node
--slave [host:ip]	Specifies a node as the master node of the currently connected node
--rdb [filename]	Obtains an RDB file from the key-value core as a local file
--pipe	Transfers raw kvdb protocol (Redis-compatible) from stdin to the key-value core
--pipe-timeout [n]	Timeout in pipe mode
--bigkeys	Finds keys with large value memory usage
--memkeys	Finds keys with large key-value memory usage
--memkeys-samples	Finds keys with large key-value memory usage, with more concise output
--hotkeys	Finds frequently used keys, but requires the cache policy to be lfu
--scan	Lists all keys with the scan command
--pattern [pat]	Performs regular expression matching of keys when using --scan/--bigkeys/--hotkeys; defaults to *
--qouted-pattern [pat]	Similar to --pattern, but the specified string type needs to be enclosed in quotes, otherwise it will be considered a non-binary-safe string
--intrinsic-latency [sec]	System intrinsic latency test; the test lasts for the specified number of seconds
--eval [file]	Sends and executes a LUA script file
--ldb	Used together with --eval, enables the kvdb lua debugger
--ldb-sync-mode	Same as --ldb, but synchronizes with the debugger; the key-value core will be blocked
--cluster [command] [args...] [opts...]	Cluster management commands and parameters; detailed commands and parameters can be used
--cluster help	Views help information
--verbose	Verbose mode
--no-auth-warning	Does not display warning messages when entering the password directly with -a
--help	Help information
--version	kvdb-cli version information

Usage: kvdb-cli --cluster [command [command arguments ...]]

Primary option	Secondary option	Description
create host1:port1 ... hostN:portN		Create a cluster
	--cluster-replicas [arg]	Number of slave nodes
check host:port		Check the cluster
	--cluster-search-multiple-owners	Check whether any slot is assigned to multiple nodes at the same time
info host:port		View cluster status
fix host:port		Fix the cluster
	--cluster-search-multiple-owners	Fix duplicate slot assignment problems
reshard host:port		Specifies any node of the cluster to migrate slots and re-shard slots
	--cluster-from [arg]	From which source nodes to migrate slots; migration can be done from multiple source nodes, separated by commas, passing node ids. You can also directly pass --from all, in which case the source nodes are all nodes of the cluster. If this parameter is not passed, the user will be prompted to enter it during migration
	--cluster-to [arg]	The node id of the destination node to which slots need to be migrated; only one destination node can be specified. If this parameter is not passed, the user will be prompted to enter it during migration
	--cluster-slots [arg]	The number of slots to migrate. If this parameter is not passed, the user will be prompted to enter it during migration.
	--cluster-yes	Specifies the confirmation input during migration
	--cluster-timeout [arg]	Sets the timeout of the migrate command

	--cluster-pipeline [arg]	Defines the number of keys fetched at once by the cluster getkeysinslot command; if not passed, the default value 10 is used
	--cluster-replace	Whether to replace directly to the target node
rebalance host:port		Specifies any node of the cluster to balance the number of slots across cluster nodes
	--cluster-weight [node1=w1...nodeN=wN]	Specifies the weights of cluster nodes
	--cluster-use-empty-masters	Allows master nodes without assigned slots to participate; not allowed by default
	--cluster-timeout [arg]	Sets the timeout of the migrate command
	--cluster-simulate	Simulates the rebalance operation without actually performing the migration
	--cluster-pipeline [arg]	Defines the number of keys fetched at once by the cluster getkeysinslot command; default value is 10
	--cluster-threshold [arg]	Executes the rebalance operation when the migrated slot threshold exceeds threshold
	--cluster-replace	Whether to replace directly to the target node
add-node new_host:new_port existing_host:existing_port		Adds a node, joining the new node into the specified cluster; adds a master node by default
	--cluster-slave	The new node acts as a slave node; a random master node by default
	--cluster-master-id [arg]	Specifies the master node for the new node
del-node host:port node_id		Deletes a given node; shuts down the node service after success
call host:port command arg arg .. arg		Executes related commands on all nodes of the cluster
set-timeout host:port		Sets cluster-node-timeout

milliseconds		
import host:port		Imports external kvdb data into the cluster (nodes not in cluster mode)
	--cluster-from [arg]	Imports the data of the specified instance into the cluster
	--cluster-copy	Specifies copy during migrate
	--cluster-replace	Specifies replace during migrate

6.2.2 Using the Client

KVDB client languages cover mainstream programming languages, such as Java, PHP, Python, C, C++, Node.js, etc. Connections can be made through kvdb-cli, and Redis clients are also supported to connect to kvdb for operations; both have the same effect.

6.2.2.1 General Operations

- Connect to kvdb-server via URL

  ```
  ./kvdb-cli -u kvdb://user:password@127.0.0.1:6359/db
  ```

- Display kvdb-cli command help information --help

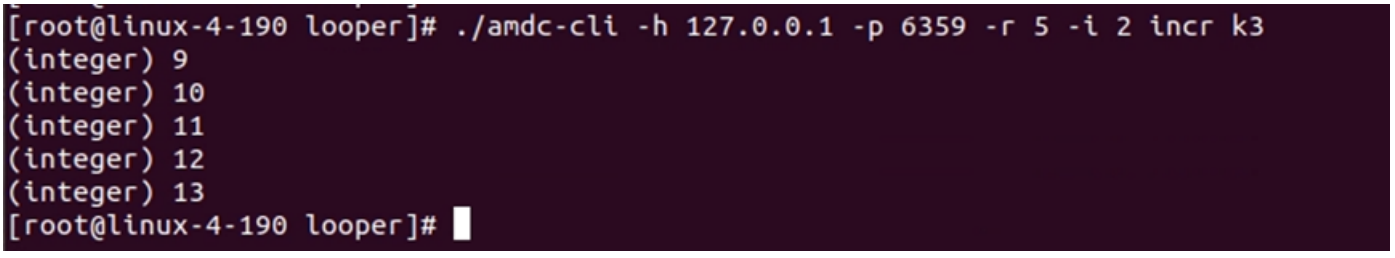
  Omitted.

- Connect to kvdb using the kvdb-cli client

  > ```
  > ./kvdb-cli -h host -p port -a password
  > ```
  >
  > > *-h is used to specify the ip*
  > > *-p is used to specify the port*
  > > *-a is used to specify the authentication password*

- Output returned data to the current command line --raw (hides data types) and --no-raw

- Run the same command n times consecutively: -r; set the run interval with -i (seconds)



```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 -r 5 -i 2 incr k3
(integer) 9
(integer) 10
(integer) 11
(integer) 12
(integer) 13
[root@linux-4-190 looper]#
```

- Connect to the specified db: -n

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 -n 2
127.0.0.1:6359[2]>
127.0.0.1:6359[2]>
127.0.0.1:6359[2]> set k5 v
OK
127.0.0.1:6359[2]> get k5
"v"
```

- Output command results in comma-separated format (csv format) --csv

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --csv lrange k4 0 -1
"6","5","4","3","2","1","0"
[root@linux-4-190 looper]#
```

- Read data from standard input (stdin) as the last argument of kvdb-cli: -x

```
[root@linux-4-190 looper]# echo "amdc-cli v2.0" | ./amdc-cli -h 127.0.0.1 -p 6359 -x set k7
OK
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359
127.0.0.1:6359> get k7
"amdc-cli v2.0\n"
127.0.0.1:6359>
```

- rdb backup --rdb

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --rdb 1.rdb
SYNC sent to master, writing 249 bytes to '1.rdb'
Transfer finished with success.
[root@linux-4-190 looper]# cat 1.rdb
REDIS0009◆ctime¶◆2cused-mem°◆repl-stream-db◆◆repl-id(dd399ab99571c81dce95f06f6f9167d9f87d944c◆
                                                                                                  repl-offset◆◆◆◆6◆

amdc v2.0
◆◆◆◆◆◆◆◆◆k1◆◆1◆k2◆◆2◆k3◆Qh◆k4◆◆◆◆◆◆◆◆◆
```

```
[root@linux-4-190 looper]# cat a.txt
set k50 v50
rpush k60 1 2 3 4 5 6 7 8 9
hset k70 key70 val70
[root@linux-4-190 looper]# cat a.txt | ./amdc-cli -h 127.0.0.1 -p 6359 --pipe
All data transferred. Waiting for the last reply...
Last reply received from server.
errors: 0, replies: 3
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359
127.0.0.1:6359> get k50
"v50"
127.0.0.1:6359> lrange k60 0 -1
1) "1"
2) "2"
3) "3"
4) "4"
5) "5"
6) "6"
7) "7"
8) "8"
9) "9"
127.0.0.1:6359> HGET k70 key70
"val70"
127.0.0.1:6359>
```

6.2.2.2 Statistics Operations

- Continuous statistics mode: view the number of kvdb keys, memory usage, and other data in real time --stat

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --stat
------- data ------ -------------------- load -------------------- - child -
keys       mem      clients blocked requests            connections
6          1.04M    2       0       26778 (+0)          37
6          1.13M    2       0       26779 (+1)          37
6          1.18M    2       0       26780 (+1)          37
6          1.22M    2       0       26781 (+1)          37
6          1.26M    2       0       26782 (+1)          37
6          1.30M    2       0       26783 (+1)          37
6          1.34M    2       0       26784 (+1)          37
```

- Find big keys --bigkeys

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --bigkeys

# Scanning the entire keyspace to find biggest keys as well as
# average sizes per key type.  You can use -i 0.1 to sleep 0.1 sec
# per 100 SCAN commands (not usually needed).

[00.00%] Biggest string found so far 'k6' with 1024 bytes
[00.00%] Biggest list   found so far 'k4' with 4 items

-------- summary -------

Sampled 6 keys in the keyspace!
Total key length in bytes is 12 (avg len 2.00)

Biggest string found 'k6' has 1024 bytes
Biggest   list found 'k4' has 4 items

0 hashs with 0 fields (00.00% of keys, avg size 0.00)
0 sets with 0 members (00.00% of keys, avg size 0.00)
5 strings with 1047 bytes (83.33% of keys, avg size 209.40)
0 zsets with 0 members (00.00% of keys, avg size 0.00)
1 lists with 4 items (16.67% of keys, avg size 4.00)
0 streams with 0 entries (00.00% of keys, avg size 0.00)
[root@linux-4-190 looper]#
```

- Find hot keys --hotkeys

```
# Scanning the entire keyspace to find biggest keys as well as
# average sizes per key type.  You can use -i 0.1 to sleep 0.1 sec
# per 100 SCAN commands (not usually needed).

[00.00%] Biggest string found so far '"k7"' with 2 bytes
[00.00%] Biggest zset   found so far '"k30"' with 3 members
[00.00%] Biggest string found so far '"k18"' with 3 bytes
[00.00%] Biggest list   found so far '"k4"' with 8 items
[50.00%] Biggest set    found so far '"k40"' with 3 members
[50.00%] Biggest hash   found so far '"k5"' with 2 fields

-------- summary -------

Sampled 20 keys in the keyspace!
Total key length in bytes is 51 (avg len 2.55)

Biggest   list found '"k4"' has 8 items
Biggest   hash found '"k5"' has 2 fields
Biggest string found '"k18"' has 3 bytes
Biggest    set found '"k40"' has 3 members
Biggest   zset found '"k30"' has 3 members

1 lists with 8 items (05.00% of keys, avg size 8.00)
1 hashs with 2 fields (05.00% of keys, avg size 2.00)
16 strings with 41 bytes (80.00% of keys, avg size 2.56)
0 streams with 0 entries (00.00% of keys, avg size 0.00)
1 sets with 3 members (05.00% of keys, avg size 3.00)
1 zsets with 3 members (05.00% of keys, avg size 3.00)
[root@linux-4-190 looper]#
```

6.2.2.3 Query Operations

- Get the list of all keys --scan

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --scan
k6
k7
k1
k4
k2
k3
[root@linux-4-190 looper]#
```

- Scan specified keys --pattern (regular expression)

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --scan --pattern k*
k6
k7
k1
k4
k2
k3
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --scan --pattern *6
k6
[root@linux-4-190 looper]#
```

- View the minimum, maximum, and average access latency of kvdb over a period of time --latency or --latency-history

```
[root@linux-4-190 looper]#
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --latency
min: 0, max: 2, avg: 0.19 (2561 samples)
```

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --latency-history
min: 0, max: 4, avg: 0.24 (1146 samples)
```

6.2.2.4 Test Operations

- Measure the latency of kvdb within n seconds --intrinsic-latency

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --intrinsic-latency 5
Max latency so far: 6 microseconds.
Max latency so far: 10 microseconds.
Max latency so far: 12 microseconds.
Max latency so far: 13 microseconds.
Max latency so far: 18 microseconds.
Max latency so far: 20 microseconds.
Max latency so far: 28 microseconds.
Max latency so far: 49 microseconds.
Max latency so far: 382 microseconds.
Max latency so far: 400 microseconds.
Max latency so far: 1170 microseconds.

1236720 total runs (avg latency: 4.0430 microseconds / 4042.95 nanoseconds per run).
Worst run took 289x longer than the average latency.
[root@linux-4-190 looper]#
```

- LRU simulation: simulates the cache eviction policy --lru-test

```
[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --lru-test 100000
28000 Gets/sec | Hits: 27884 (99.59%) | Misses: 116 (0.41%)
28250 Gets/sec | Hits: 28250 (100.00%) | Misses: 0 (0.00%)
28000 Gets/sec | Hits: 28000 (100.00%) | Misses: 0 (0.00%)
27000 Gets/sec | Hits: 26969 (99.89%) | Misses: 31 (0.11%)
29000 Gets/sec | Hits: 28929 (99.76%) | Misses: 71 (0.24%)
28750 Gets/sec | Hits: 28738 (99.96%) | Misses: 12 (0.04%)
29500 Gets/sec | Hits: 29500 (100.00%) | Misses: 0 (0.00%)
28500 Gets/sec | Hits: 28500 (100.00%) | Misses: 0 (0.00%)
28500 Gets/sec | Hits: 28382 (99.59%) | Misses: 118 (0.41%)
28250 Gets/sec | Hits: 28250 (100.00%) | Misses: 0 (0.00%)
29500 Gets/sec | Hits: 29500 (100.00%) | Misses: 0 (0.00%)
29250 Gets/sec | Hits: 29250 (100.00%) | Misses: 0 (0.00%)
28750 Gets/sec | Hits: 28750 (100.00%) | Misses: 0 (0.00%)
28500 Gets/sec | Hits: 28251 (99.13%) | Misses: 249 (0.87%)
28750 Gets/sec | Hits: 28750 (100.00%) | Misses: 0 (0.00%)
28750 Gets/sec | Hits: 28500 (99.13%) | Misses: 250 (0.87%)
29500 Gets/sec | Hits: 29500 (100.00%) | Misses: 0 (0.00%)
26250 Gets/sec | Hits: 26022 (99.13%) | Misses: 228 (0.87%)
```

- Simulates and displays a replica of the commands received from the master server --replica

```
[root@linux-4-190 looper]# redis-cli -h 127.0.0.1 -p 6359 --replica
sending REPLCONF capa eof
SYNC with master, discarding 478 bytes of bulk transfer...
SYNC done. Logging commands from master.
"PING"
"PING"
"PING"
"PING"
"PING"
"PING"
"PING"
"PING"
"PING"
"PING"
"PING"
```

6.2.2.5 LUA Operations

- Execute a lua script

```
[root@linux-4-190 looper]# cat script.lua
return redis.call('GET',"k1")

[root@linux-4-190 looper]# ./amdc-cli -h 127.0.0.1 -p 6359 --eval ./script.lua
"v1"
[root@linux-4-190 looper]#
```

- Start lua debug mode: --ldb or --ldb-sync-mode

```
[root@linux-4-190 looper]# redis-cli -h 127.0.0.1 -p 6359 --eval ./script.lua --ldb
Lua debugging session started, please use:
quit    -- End the session.
restart -- Restart the script in debug mode again.
help    -- Show Lua script debugging commands.

v1
lua debugger>
lua debugger>
[root@linux-4-190 looper]# redis-cli -h 127.0.0.1 -p 6359 --eval ./script.lua --ldb-sync-mode
Lua debugging session started, please use:
quit    -- End the session.
restart -- Restart the script in debug mode again.
help    -- Show Lua script debugging commands.

v1
lua debugger>
lua debugger>
```

6.2.2.6 Cluster Operations

The current multi-tenant version does not support cluster operations

6.3 RDB Cluster Data Migration Tool Usage Introduction

The current multi-tenant version does not support cluster operations

6.4 Performance Test Tool Usage Introduction

kvdb-benchmark is a dedicated performance test tool provided to users for testing KVDB, so as to quickly understand KVDB's performance and provide a reliable basis for tuning.

6.4.1 KVDB benchmark Parameters

kvdb-benchmark

Example: `kvdb-benchmark [-h <host>] [-p <port>] [-c <clients>] [-n <requests>]`

Command parameters

```
-h <hostname>      Instance host name (default 127.0.0.1)
-p <port>          Instance port (default 6359)
-a <password>      Instance password
-c <clients>       Number of concurrent connections (default 50)
-n <requests>      Total number of requests (default 100000)
-d <size>          Data size of SET/GET, in bytes (default 2 )
-dbnum <db>        Specifies the db (KVDB has databases 0-15 by
default; this parameter defaults to 0)
-k <boolean>       1=keep connection 0=reconnect (default 1)
-r <keyspacelen>   SET/GET/INCR use random keys, SADD uses random
values; keyspacelen refers to the maximum length of random data
(maximum length range is 0-12)
-P <numreq>        Uses a pipeline to combine the specified number of
requests; default 1 (no pipeline)
-e                 If the server returns an error, displays it in
standard output (no more than 1 error displayed per second)
-q                 Only displays the query/sec value
--csv              Outputs in CSV format
-l                 Loop; keeps testing continuously
-t <tests>         Specifies the test commands
-I                 Idle mode; opens N idle connections and waits
--threads <num>   Enables the specified number of threads; default 1
```

memtier-benchmark parameters

Example: `memtier-benchmark [-h <host>] [-p <port>] [-c <clients>] [-n <requests>] [-t <threads>]`

Command parameters

```
-h <hostname>      Instance host name (default 127.0.0.1)
-p <port>          Instance port (default 6359)
-s <socket>        Uses a Unix domain socket
-a <password>      Authentication password
--tls              Enables TLS encryption
```

```
-c <clients>      Number of clients
-t <threads>      Number of threads
--ratio           Read-write ratio
--pipeline <num>  Number of pipeline commands
--randomdata      Uses random data
--hide-histogram  Hides detailed histogram information
--csv <file>      Outputs a csv format file
--test-time <sec> Duration of the test
```

7 All Product Configuration Descriptions

7.1 All Key-Value Core Configurations

The KVDB key-value core has three actively configurable configuration files: key-value configuration (kvdb.yaml), sentinel configuration (sentinel.yaml), and authorization center configuration (acls.properties).

7.1.1 Cache Configuration File

The KVDB key-value core configuration file is stored at: /installation_root_directory/kvdb/kvdb.yaml. Some configurations can be modified through the console. The following are the detailed configurations of the KVDB key-value core:

Parameter	Default	Remarks
bind	127.0.0.1,-::1	The network addresses to listen on; by default, only the local loopback addresses are listened on. The - prefix means that startup is not affected when the address is unavailable
bind-source-addr	""	Forces all outbound connections to be sent from the specified local IP; not set by default, and automatically selected by the operating system routing table
port	6359	The TCP port to listen on; set to 0 to disable listening on TCP sockets
protected-mode	yes	Protected mode. When no bind is specified and no password is set, only local loopback addresses and Unix domain socket connections are allowed
mptcp-enabled	no	MPTCP multipath transmission control protocol; supports concurrent multi-link transmission, improving link aggregation bandwidth and transmission throughput; requires Linux kernel version 5.6 or above
tcp-backlog	511	The backlog size of TCP listening; the actual value may be limited by the system somaxconn and tcp_max_syn_backlog
unixsocket	""	Unix domain socket path; if not specified, Unix sockets are not listened on
unixsocketperm	0	Permissions of the Unix domain socket

timeout	0	Client idle timeout (seconds); 0 means disabled
tcp-keepalive	300	TCP keepalive time (seconds), used to detect dead connections and maintain network connections
socket-mark-id	0	Specifies a mark ID for the listening socket, used to implement advanced routing and traffic filtering capabilities; default 0 means disabled
daemonize	no	Whether to run in daemon mode
supervised	no	Interaction mode with the monitoring system (upstart/systemd); no interaction by default
pidfile	"/var/run/kvdb_6359.pid"	pid file path
loglevel	notice	Log level: debug, verbose, notice, warning
logfile	""	Log file path; an empty string means output to standard output
syslog-enabled	no	Whether to enable system logging
syslog-ident	"KVDB"	System log identifier
syslog-facility	local0	System log facility; must be USER or LOCAL0-LOCAL7
crash-log-enabled	yes	Whether to enable crash logs
crash-memcheck-enabled	yes	Whether to enable fast memory check in crash logs
worker-threads	1	Sets the number of worker threads; default is 1
databases	16	Number of databases; DB 0 is used by default
always-show-logo	no	Whether to display the ASCII art logo in the startup log
set-proc-title	yes	Whether to modify the process title to display runtime information
proc-title-template	"{title} {listen-addr} {server-mode}"	Process title template; supports multiple variables
locale-collate	""	Locale rules for string comparison / sorting
loadmodule	""	Loads the specified module at startup; if loading fails, the server will terminate startup. Can be used multiple times to load multiple modules

tls-port	0	TLS listening port; use together with port 0 to enable TLS only
tls-cert-file	""	Server TLS certificate file path (PEM format)
tls-key-file	""	Server TLS private key file path (PEM format)
tls-key-file-pass	""	Password of the TLS private key file (if encrypted)
tls-client-cert-file	""	Client TLS certificate file path (for client authentication)
tls-client-key-file	""	Client TLS private key file path
tls-client-key-file-pass	""	Password of the client TLS private key file (if encrypted)
tls-dh-params-file	""	DH parameter file path
tls-ca-cert-file	""	CA certificate file path used to verify clients and nodes
tls-ca-cert-dir	""	CA certificate directory path used to verify clients and nodes
tls-auth-clients	yes	Client certificate verification policy; default yes. All clients on the TLS port must use valid client certificates for authentication
tls-auth-clients-user	off	Enables TLS client certificate auto-login. CN: the server reads the CN field from the client TLS certificate; off: disables certificate auto-authentication
tls-replication	no	Whether to enable TLS in master-slave replication
tls-cluster	no	Whether to enable TLS in the cluster
tls-protocols	""	Allowed TLS versions; case-insensitive
tls-ciphers	""	Cipher suites allowed for TLSv1.2 and below. For Chinese national cryptography, use ECC-SM2-SM4-GCM-SM3:ECC-SM2-SM4-CBC-SM3
tls-ciphersuites	""	Cipher suites allowed for TLSv1.3. For Chinese national cryptography, use TLS_SM4_GCM_SM3
tls-prefer-server-ciphers	no	Whether to prefer the server's cipher suite preference
tls-session-	yes	Whether to enable TLS session caching to speed up

caching		reconnection
tls-session-cache-size	20480	Maximum number of entries in the TLS session cache; 0 means unlimited
tls-session-cache-timeout	300	Timeout of the TLS session cache (seconds)
save	3600 1,300 100,60 10000	RDB persistence trigger conditions, in the format ; save "" disables automatic RDB saving
stop-writes-on-bgsave-error	no	Whether to stop accepting writes when RDB persistence fails
rdbcompression	yes	Whether to use LZF to compress string objects in RDB files
rdbchecksum	yes	Whether to add a CRC64 checksum at the end of the RDB file; disabling it can improve performance
sanitize-dump-payload	no	Whether to perform comprehensive verification when loading RDB or RESTORE data; currently set to no by default due to cluster migration issues
dbfilename	"dump.rdb"	RDB file name
rdb-del-sync-files	no	Whether to delete the RDB files used for replication when no persistence is configured; only valid when both AOF and RDB are disabled
dir	"./"	Directory where data files (RDB, AOF) are stored
replicaof	""	Master-slave replication configuration, e.g.: replicaof: " "
masterauth	""	Authentication password of the master server, e.g.: masterauth: ""
masteruser	""	Username of the master server used for replication, e.g.: masteruser: ""
repl-mptcp-enabled	no	Whether to enable MPTCP for master-slave replication
replica-serve-stale-data	yes	Whether to respond to client requests during replication synchronization (may return stale data)
replica-read-only	yes	Whether the replica server is in read-only mode

repl-diskless-sync	no	Whether to enable diskless replication (transferring RDB directly through the socket)
repl-diskless-sync-delay	5	Delay time (seconds) waiting for new replica nodes during diskless replication
repl-diskless-sync-max-replicas	0	During diskless replication, the threshold of the number of replicas that triggers synchronization in advance
repl-diskless-load	disabled	How the replica node loads RDB: disabled (save to disk first), on-empty-db (diskless only when the database is empty), swapdb (keep old data in memory)
repl-ping-replica-period	10	Interval (seconds) at which the replica node sends PING to the master node
repl-timeout	60	Replication timeout (seconds), covering SYNC transfer, master-slave heartbeat, etc.
repl-disable-tcp-nodelay	no	Whether to disable TCP_NODELAY on the replication link; yes reduces bandwidth but increases latency
repl-backlog-size	1mb	Size of the replication backlog buffer, affecting partial resynchronization capability
repl-backlog-ttl	3600	Delay time (seconds) before the backlog buffer is released after the last replica node disconnects; 0 means never released
replica-priority	100	Replica node priority, used by Sentinel for master election; 0 means it cannot be elected as master
propagation-error-behavior	ignore	Server behavior when execution failures are encountered during master-slave replication / AOF recovery. ignore (default): skips erroneous commands and continues running; panic: crashes and exits immediately upon error; panic-on-replicas: only replica nodes crash when they encounter replication stream errors
replica-ignore-disk-write-errors	no	Server behavior when a replica node fails to write to disk. no (default): crashes directly; yes: ignores the error and continues running
replica-announced	yes	Whether the replica node is recognized and reported by Sentinel, so that the console and upper-layer clients can perceive this replica node

min-replicas-to-write	0	Minimum number of replica nodes required for the master node to accept writes; 0 disables this feature
min-replicas-max-lag	10	Maximum lag time (seconds) of replica nodes; used together with min-replicas-to-write
replica-announce-ip	""	IP address announced by the replica node to the master node (for NAT/Docker and other scenarios)
replica-announce-port	0	Port announced by the replica node to the master node (for NAT/Docker and other scenarios)
tracking-table-max-keys	1000000	Maximum number of keys in the client caching tracking table; 0 means unlimited
user	""	user: " "
acllog-max-len	128	Maximum number of entries in the ACL log
aclfile	""	Path of the external ACL user file; cannot be mixed with user configuration inside the configuration file
requirepass	""	Password of the default user (equivalent to the password of the ACL default user)
acl-pubsub-default	allchannels	Default Pub/Sub channel permission for new users: allchannels (allow all), resetchannels (deny all)
rename-command	""	Command renaming, e.g.: rename-command: "flushall ''", rename-command: "flushall no-flushall"
maxclients	10000	Maximum number of concurrent client connections, limited by system file descriptors
maxmemory	0	Memory usage limit; when reached, handled according to maxmemory-policy
maxmemory-policy	volatile-lru	Eviction policy when the memory limit is reached, such as volatile-lru, allkeys-lfu, etc.
maxmemory-samples	5	Number of samples for LRU/LFU/TTL algorithms; the larger the more accurate, but the higher the CPU consumption
maxmemory-eviction-tenacity	10	Aggressiveness of eviction processing, from 0 (lowest latency) to 100 (prioritize eviction)

replica-ignore-maxmemory	yes	Whether replica nodes ignore their own maxmemory configuration (eviction is handled by the master node by default)
active-expire-effort	1	Scanning effort for actively expiring keys, 1-10; the larger the more frequent (may increase CPU and latency)
lazyfree-lazy-eviction	no	Whether to use lazy deletion (releasing memory in background threads) during memory eviction
lazyfree-lazy-expire	no	Whether to use lazy deletion when keys expire
lazyfree-lazy-server-del	no	Whether to use lazy deletion when the server internally deletes keys (e.g. RENAME overwrite)
replica-lazy-flush	no	Whether to lazily flush the database during full synchronization of replica nodes
lazyfree-lazy-user-del	no	Whether to use lazy deletion when the user executes the DEL command
lazyfree-lazy-user-flush	no	Default behavior when the user executes the FLUSH command without specifying sync / async
oom-score-adj	no	Whether to control the process score of the kernel OOM killer; yes/relative for relative adjustment, absolute for absolute setting
oom-score-adj-values	0 200 800	OOM score adjustment values, corresponding to the master node, replica nodes, and background child processes respectively
disable-thp	yes	Whether to disable kernel Transparent Huge Pages (THP) to avoid performance problems with fork and CoW
appendonly	no	Whether to enable AOF persistence
appendfilename	"appendonly.aof"	AOF file name
appenddirname	"appendonlydir"	Dedicated storage directory for all appendonly files
appendfsync	everysec	AOF fsync policy: no (relies on the system), always (on every write), everysec (once per second, default)
no-appendfsync-on-rewrite	no	Whether to disable fsync of the main process during AOF rewrite; may increase the risk of data loss but improves

		performance
auto-aof-rewrite-percentage	100	Trigger percentage for automatic AOF rewrite (growth ratio of current size compared to the last rewrite)
auto-aof-rewrite-min-size	64mb	Minimum file size for automatic AOF rewrite
aof-load-truncated	yes	Whether to load a truncated AOF file at startup; no will report an error and exit
aof-use-rdb-preamble	yes	Whether to use the RDB preamble during AOF rewrite (improves rewrite and recovery speed)
aof-timestamp-enabled	no	Whether to support recording timestamps in AOF to enable restoring data from a specific point in time
shutdown-timeout	10	When the master node shuts down / stops the service, the maximum timeout (in seconds) waiting for replica synchronization. Set to 0 to disable this waiting mechanism and stop immediately
shutdown-on-sigint	save	Server shutdown behavior when receiving SIGINT (Ctrl+C). default: if automatic persistence save rules are configured, blocks and generates an RDB snapshot; save: ignores the automatic save configuration and forcibly persists to RDB; nosave: even if save rules are configured, no RDB is generated on shutdown; now: skips synchronization waiting directly and shuts down quickly; force: ignores all kinds of obstacles and forcibly exits the process
shutdown-on-sigterm	save	Server shutdown behavior when receiving SIGTERM; same as above
busy-reply-threshold	5000	Maximum execution time of a Lua script (milliseconds); after timeout, only SCRIPT KILL and SHUTDOWN NOSAVE are allowed
cluster-enabled	no	Whether to enable cluster mode
cluster-config-file	"node-6359.conf"	Cluster configuration file path, automatically maintained by the node
cluster-node-timeout	15000	Cluster node timeout (milliseconds), used to determine node failure

cluster-port	0	Dedicated port for internal cluster communication; when default 0, the business port + 10000 is automatically used
cluster-replica-validity-factor	10	Validity factor of replica node failover, affecting failover conditions
cluster-migration-barrier	1	Barrier for replica node migration (number of replica nodes that the original master node must retain)
cluster-allow-replica-migration	yes	Whether to allow replica nodes to automatically migrate to master nodes without replica nodes
cluster-require-full-coverage	yes	Whether the cluster requires all hash slots to be covered before accepting requests
cluster-replica-no-failover	no	Whether replica nodes are prohibited from automatic failover (can still be triggered manually)
cluster-allow-reads-when-down	no	When the cluster is unavailable, whether nodes still respond to read requests for the slots they are responsible for
cluster-allow-pubsubshard-when-down	yes	When the cluster is globally abnormally offline, whether nodes are allowed to continuously provide sharded publish/subscribe services based on local slot ownership, ensuring high availability of the message link
cluster-link-sendbuf-limit	0	Memory limit of the send buffer of the communication link between cluster nodes, preventing slow nodes from causing the buffer to expand indefinitely and fill up memory. Default 0 means unlimited
cluster-announce-hostname	""	Hostname / domain name announced by the cluster node to the outside, used for TLS SNI, DNS routing, containerized / cloud environments; empty by default means disabled
cluster-announce-human-nodename	""	Sets a custom friendly name for the cluster node, as a supplementary identifier to the node ID
cluster-preferred-endpoint-type	ip	Specifies the type of connection address the cluster exposes to clients: ip (default), hostname, or unknown-endpoint, determining the address format returned by MOVED redirection and CLUSTER SLOTS

cluster-compatibility-sample-ratio	0	Defines the sampling ratio (0-100) of command compatibility checks in cluster mode, used to detect incompatible commands such as cross-slot operations. Default 0 means disabled; the higher the ratio, the stronger the security but the greater the performance overhead
cluster-slot-stats-enabled	no	Whether to enable cluster slot resource statistics, supporting monitoring of CPU / network / memory usage, for hotspot identification, load balancing, and slot migration optimization
cluster-slot-migration-write-pause-timeout	10000	During cluster slot migration, the maximum timeout (milliseconds) for the source node to pause writes
cluster-slot-migration-handoff-max-lag-bytes	1mb	Sets the maximum allowed replication lag in bytes for triggering the handoff phase of slot migration. When the amount of data to be synchronized between source and target nodes is below this value, the slot migration handoff process begins and the source node pauses writes
cluster-announce-ip	""	IP address announced by the cluster node to the outside (for NAT/Docker scenarios)
cluster-announce-tls-port	0	TLS client port announced by the cluster node to the outside (for NAT/Docker scenarios)
cluster-announce-port	0	Client port announced by the cluster node to the outside (for NAT/Docker scenarios)
cluster-announce-bus-port	0	Bus port announced by the cluster node to the outside (for NAT/Docker scenarios)
slowlog-log-slower-than	10000	Threshold (microseconds) for slow log recording; negative values disable it, 0 records all commands
slowlog-max-len	128	Maximum number of entries in the slow log; old entries are evicted after exceeding it
latency-monitor-threshold	0	Threshold (milliseconds) for latency monitoring; 0 disables monitoring
latency-tracking	yes	Whether to enable command latency tracking

latency-tracking-info-percentiles	"50 99 99.9"	Defines the latency percentiles displayed by INFO; default 50/99/99.9
notify-keyspace-events	""	Keyspace event notification configuration; see the configuration comments for details
hash-max-listpack-entries	512	Maximum number of entries for a hash table to use listpack encoding
hash-max-listpack-value	64	Maximum value size (bytes) for a hash table to use listpack encoding
list-max-listpack-size	-2	Node size limit for a list to use listpack encoding; -1 to -5 correspond to 4Kb to 64Kb; positive numbers are the number of entries
list-compress-depth	0	Compression depth of a list; 0 disables compression; a positive number indicates the number of nodes not compressed at the head and tail
set-max-intset-entries	512	Maximum number of entries for a set to use intset encoding (elements are 64-bit signed integers)
set-max-listpack-entries	128	Maximum number of entries for a set to use listpack encoding
set-max-listpack-value	64	Maximum value size (bytes) for a set to use listpack encoding
zset-max-listpack-entries	128	Maximum number of entries for a sorted set to use listpack encoding
zset-max-listpack-value	64	Maximum value size (bytes) for a sorted set to use listpack encoding
hll-sparse-max-bytes	3000	Maximum number of bytes for HyperLogLog to use sparse representation (including 16-byte header)
stream-node-max-bytes	4kb	Maximum number of bytes of a single node in the stream data structure
stream-node-max-entries	100	Maximum number of entries of a single node in the stream data structure
stream-idmp-duration	100	Retention duration (seconds) of the IDMP message unique identifier; default 100 seconds, automatically cleaned up after timeout

stream-idmp-maxsize	100	Maximum number of IDs recorded per producer per stream; default 100; the oldest is automatically evicted when full
activerehashing	yes	Whether to enable active rehashing to periodically release hash table memory
client-output-buffer-limit normal	0 0 0	Output buffer limits for normal clients (hard limit, soft limit, soft limit duration)
client-output-buffer-limit replica	0 0 0	Output buffer limits for replica node clients
client-output-buffer-limit pubsub	32mb 8mb 60	Output buffer limits for Pub/Sub clients
client-query-buffer-limit	1gb	Maximum size of the client query buffer
lookahead	16	Sets the number of client pipeline commands to decode and prefetch in advance
maxmemory-clients	5%	Client memory eviction mechanism: limits the total memory of all client connections; when the threshold is exceeded, the connections occupying the most memory are disconnected first to prevent OOM
proto-max-bulk-len	512mb	Maximum size (bytes) of bulk requests in the protocol; must be ≥1mb
hz	10	Execution frequency (hertz) of background tasks, 1-500; the higher the better responsiveness but the more CPU consumption
dynamic-hz	yes	Whether to enable dynamic Hz, automatically increasing the frequency when there are many clients
aof-rewrite-incremental-fsync	yes	Whether to perform an fsync for every 32MB of data generated during AOF rewrite, reducing latency peaks
rdb-save-incremental-fsync	yes	Whether to perform an fsync for every 32MB of data generated during RDB save, reducing latency peaks
lfu-log-factor	10	Counter logarithmic factor of the LFU eviction algorithm, affecting the counter growth speed

lfu-decay-time	1	Decay time (minutes) of the LFU counter; the counter is halved after this duration
max-new-connections-per-cycle	10	Limits how many client connections can be accepted per event loop cycle, preventing a sudden large number of new connections from freezing the server
max-new-tls-connections-per-cycle	1	Limits how many TLS-encrypted client connections can be accepted per event loop cycle, preventing a sudden large number of new connections from freezing the server
activedefrag	no	Whether to enable active memory defragmentation (requires Jemalloc support)
active-defrag-ignore-bytes	100mb	Minimum number of wasted fragmentation bytes to trigger active defragmentation
active-defrag-threshold-lower	10	Minimum fragmentation percentage to trigger active defragmentation
active-defrag-threshold-upper	100	Fragmentation percentage at which maximum-effort defragmentation is triggered
active-defrag-cycle-min	1	Minimum CPU percentage for defragmentation, used when the lower threshold is reached
active-defrag-cycle-max	25	Maximum CPU percentage for defragmentation, used when the upper threshold is reached
active-defrag-max-scan-fields	1000	Maximum number of fields processed per main dictionary scan (applicable to structures such as sets and hashes)
jemalloc-bg-thread	yes	Whether to enable Jemalloc background threads for memory release
server-cpulist	""	CPU affinity list for server threads (e.g. "0-7:2" means cores 0, 2, 4, 6)
bio-cpulist	""	CPU affinity list for background I/O threads
aof-rewrite-cpulist	""	CPU affinity list for the AOF rewrite child process
bgsave-cpulist	""	CPU affinity list for the BGSAVE child process
ignore-warnings	""	List of warnings to ignore (space-separated), e.g.

		"ARM64-COW-BUG"
key-memory-histograms	no	Whether to enable key memory allocation histogram statistics by data type

7.1.2 Sentinel Configuration File

The sentinel configuration file is: `sentinel.yaml`

Parameter	Default	Remarks
bind	127.0.0.1, -::1	The network addresses to listen on; by default, only the local loopback addresses are listened on
port	26359	The TCP port to listen on; set to 0 to disable listening on TCP sockets
protected-mode	no	Protected mode; when no bind is specified and no password is set, only local loopback connections are allowed
unixsocket	empty	Unix domain socket path; if not specified, it is not listened on
unixsocketperm	0	Unix domain socket permissions
daemonize	no	Whether to run in daemon mode
pidfile	/var/run/kvdb-sentinel.pid	PID file path
loglevel	notice	Log level: debug, verbose, notice, warning
logfile	empty	Log file path; empty means output to standard output
syslog-enabled	no	Whether to enable system logging
syslog-ident	KVDB	System log identifier
syslog-facility	local0	System log facility; supports USER / LOCAL0~LOCAL7
dir	./	Directory where data files are stored
maxclients	10000	Maximum number of concurrent client connections
user	empty	ACL user configuration, format:
acllog-max-len	128	Maximum number of ACL log entries

aclfile	empty	External ACL user file path
requirepass	empty	Default user password
tls-port	0	TLS listening port; use together with port 0
tls-cert-file	empty	Server TLS certificate file (PEM)
tls-key-file	empty	Server TLS private key file (PEM)
tls-key-file-pass	empty	TLS private key password
tls-client-cert-file	empty	Client TLS certificate file
tls-client-key-file	empty	Client TLS private key file
tls-client-key-file-pass	empty	Client TLS private key password
tls-dh-params-file	empty	DH parameter file
tls-ca-cert-file	empty	CA certificate for client/node verification
tls-ca-cert-dir	empty	CA certificate directory
tls-auth-clients	yes	Client certificate verification policy: no=not accepted, optional=optional but must be valid
tls-auth-clients-user	off	TLS certificate auto-login: CN=reads the certificate user, off=disabled
tls-replication	no	Whether to enable TLS for master-slave replication
tls-cluster	no	Whether to enable TLS for the cluster
tls-protocols	empty	Allowed TLS versions
tls-ciphers	empty	Cipher suites for TLSv1.2 and below
tls-ciphersuites	empty	Cipher suites for TLSv1.3
tls-prefer-server-ciphers	no	Whether to prefer server cipher suites
tls-session-caching	yes	Enables TLS session caching
tls-session-cache-size	20480	Maximum number of TLS session cache entries
tls-session-cache-timeout	300	TLS session cache timeout (seconds)
sentinel monitor	mymaster 127.0.0.1 6359 2	Monitor the master node: name, IP, port, quorum

sentinel auth-user	mymaster empty	Authentication username of master-slave nodes
sentinel auth-pass	mymaster empty	Authentication password of master-slave nodes
sentinel down-after-milliseconds	mymaster 30000	Subjective down timeout (milliseconds)
sentinel sentinel-user	empty	Sentinel cluster communication username
sentinel sentinel-pass	empty	Sentinel cluster communication password
sentinel parallel-syncs	mymaster 1	Number of replica nodes synchronized concurrently during failover
sentinel failover-timeout	mymaster 180000	Failover timeout (milliseconds)
sentinel notification-script	mymaster empty	Warning event notification script
sentinel client-reconfig-script	mymaster empty	Client reconfiguration script after failover
sentinel deny-scripts-reconfig	yes	Prohibits modifying script configuration at runtime
sentinel rename-command	mymaster empty empty	Command renaming
sentinel resolve-hostnames	no	Whether to support hostname resolution (depends on DNS)
sentinel announce-hostnames	no	Whether to announce hostnames to the outside
sentinel announce-ip	empty	IP announced to the outside in NAT/Docker environments
sentinel announce-port	0	Port announced to the outside in NAT/Docker environments
sentinel master-reboot-down-after-period	mymaster 0	Does not trigger failover while the master node is loading data during a restart

7.1.3 Authorization and Authentication Center Configuration

When using the authentication center, fill in acls.properties; you need to provide the address of the apusic license authentication center. The authentication center address must be in ip:port format, and multiple addresses are separated by commas.

Note: This configuration existed in the old version of kvdb.yaml and has been removed in the new version. However, it remains compatible with the old version.

Configuration item	Default	Description
apusic_acls_enable	false	Whether to enable the apusic license authentication center; use true to enable, otherwise false
apusic_acls_authUrls	""	Authentication center address; must be in ip:port format, multiple addresses separated by commas
apusic_acls_ns	"public"	Namespace
apusic_acls_tenant	""	Tenant name

8 Product Tools Usage Guide

8.1 kvdb-check-rdb Tool User Manual

kvdb-check-rdb is KVDB's RDB file verification and repair tool. It is used to check the integrity and validity of RDB snapshot files, troubleshoot file corruption problems, and can repair some corrupted RDB files, avoiding KVDB startup failures or data loss caused by corrupted RDB files.

Core functions:

- Verifies whether an RDB file is complete and undamaged, and detects whether the file format conforms to KVDB specifications;
- Repairs RDB files corrupted by unexpected interruptions (such as server downtime or power failure);
- Views the basic information of RDB files (such as version, data volume, expiration time, etc.).

8.1.1 Tool Location and Prerequisites

8.1.1.1 Tool Location

After KVDB is installed, the kvdb-check-rdb tool is located in the KVDB installation directory folder:

```
Linux: /kvdb/kvdb-check-rdb (default path; adjust accordingly if a
custom installation is used)
```

8.1.1.2 Prerequisites

- No need to start the KVDB service; can be run directly on the command line;
- Read permission for the RDB file is required (if file permissions are insufficient, use chmod to modify permissions on Linux);
- The RDB file path must be correct (either absolute or relative paths are acceptable; relative paths are based on the directory where the current terminal is located).

8.1.2 Basic Syntax

The syntax of kvdb-check-rdb is simple; the core is only to specify the path of the RDB file to be processed. Optional parameters are used for repair or viewing detailed information:

```
# Basic syntax (verify RDB file)
kvdb-check-rdb [options] /path/to/dump.rdb

# Common shorthand (without options, only file integrity is verified by default)
kvdb-check-rdb dump.rdb
```

8.1.3 Core Options and Parameter Descriptions

kvdb-check-rdb has few options; focus on the repair and debugging related options, as follows:

Option	Description	Use Case
--fix	Repairs a corrupted RDB file, retaining recoverable data and discarding the parts that cannot be repaired	When the RDB file is corrupted and KVDB fails to start
--help (abbreviated -h)	Displays the tool's help information, including all option descriptions and basic syntax	Quick lookup when you forget the command syntax

8.1.4 Common Operation Examples

8.1.4.1 Verifying RDB File Integrity

No options are needed; directly specify the RDB file path, and the tool will automatically verify whether the file is complete and the format is correct:

```
# Verify the RDB file
kvdb-check-rdb/path/to/dump.rdb
```

Normal output (file is intact):

```
[offset 0] Checking RDB file dump.rdb
[offset 29] AUX FIELD kvdb-ver = '2.0'
[offset 46] AUX FIELD kvdb-bits = '64'
[offset 58] AUX FIELD ctime = '1769518212'
[offset 73] AUX FIELD used-mem = '2442688'
[offset 91] AUX FIELD repl-stream-db = '0'
[offset 141] AUX FIELD repl-id =
```

```
'069cb616c524d6ec885da6218b669f1d9c970125'
[offset 156] AUX FIELD repl-offset ='0'
[offset 172] AUX FIELD aof-preamblle ='0'
[offset 181] Checksum OK
[offset 181] \o/ RDB looks OK! \o/
[info]0 keys read
[info]0 expires
[info] 0 already expired
```

Abnormal output (file is corrupted):

```
[offset 0] Checking RDB file dump.rdb
[offset 21] AUX FIELD ctime = '1768265496'
[offet 42] AUX FIELD used-mem = '4275241848'
[offset 60] AUX FIELD repl-stream-db = '0'
[offset 11 ] AUX FIELD repl-id =
'f0a6b7b58ed1ece135f1ba84bcc8add612e8b347'
[offset 135] AUX FIELD repl-offset = '67232446953'
[offset 137] Selecting DB ID 0
--- RDB ERROR DETECTED ---
[offset 73027] Internal error in RDB reading offset 0, function at
rdb.c:1936 -> Intset integrity check failed.
[additional info] While doing: read-object-value
[additional info] Reading key
'example.tenant.wf.isallownextparticipant.demo.audit'
[additional info] Reading type 11 (set-intset)
[info] 159 keys read
[info] 158 expires
[info] 157 already expired
692:C 02 Feb 2026 10:49:23.182 # Terminating server after rdb file
reading failure.
```

8.1.4.2 Repairing a Corrupted RDB File

When an RDB file is corrupted, use the --fix option to repair it. After repair, the dump file is named dump-.rdb:

```
# Repair a corrupted RDB file
kvdb-check-rdb --fix /path/to/dump.rdb
```

Successful repair output:

```
[offset 0] Checking RDB file --fix
[offset 21] AUX FIELD ctime = '1768265496'
[offset 42] AUX FIELD used-mem = '4275241848'
[offset 60] AUX FIELD repl-stream-db = '0'
[offset 10] AUX FIELD repl-id =
'f0a6b7b58ed1ece135f1ba84bcc8add612e8b347'
[offset 135] AUX FIELD repl-offset = '67232446953'
[offset 137] Selecting DB ID 0
--- RDB ERROR DETECTED---
[offset 73027] Internal error in RDB reading offset 0, function at
rdb.c:1936 -> Intset integrity check failed.
[additional info] While doing: read-object-value
[additional info] Reading key
'example.tenant.wf.isallownextparticipant.demo.audit'
[additional info] Reading type 11 (set-intset)
[info] 159 keys read
[info] 158 expires
[info] 157 already expired
--- RDB ERROR DETECTED---
[offset 178543] Internal error in RDB reading offset 0, function at
rdb.c:1936 -> Intset integrity check failed.
[additional info] While doing: read-object-value
[additional info] Reading key
'example.tenant.wf.isallownextparticipant.demo.submit'
[additional info] Reading type 11 (set-intset)
[info] 392 keys read
[info] 390 expires
[info] 388 already expired
--- RDB ERROR DETECTED---
[offset 260382] Internal error in RDB reading offset 0, function at
```

```
rdb.c:1936 -> Intset integrity check failed.
[additional info] While doing: read-object-value
[additional info] Reading key
'example.tenant.wf.isallownextparticipant.demo.save'
[additional info] Reading type 11(set-intset)
[info] 571 keys read
[info] 568 expires
[info] 564 already expired
--- RDB ERROR DETECTED---
[offset 136547728] Invalid object type: 111
[additional info] While doing: read-type
[info] 114982 keys read
[info] 114589 expires
[info] 113699 already expired
[offset 136547728] \o/ RDB partially fixed: dump-790.rdb! \o/
[info] 114982 / 406455 keys fixed
```

8.1.5 Notes

- Whether to back up the original RDB file before repair: the --fix option does not directly overwrite the original file; after repair, the dump file is named dump-.rdb. To be safe, you can back up the RDB file before the operation;
- Only partially corrupted files can be repaired: if the header information of the RDB file is corrupted or the checksum is seriously wrong, the tool may not be able to repair it; in this case, you need to rely on backup files;
- Some data may be lost after repair: for key-value pairs with abnormal formats in the RDB file, the tool will try to skip and discard the data; if an unrepairable data segment is detected, it will stop scanning the remaining contents of the file, so the tool may only complete the repair of part of the file's data;
- The tool version must match the KVDB version: RDB file formats may differ between versions. Use the kvdb-check-rdb tool of the same version as the KVDB service to avoid verification or repair failures.

8.2 kvdb-check-aof Tool User Manual

8.2.1 Tool Introduction

kvdb-check-aof is an AOF (Append Only File) verification and repair tool. It is used to check the integrity and syntax correctness of AOF files, repair AOF file corruption caused by unexpected interruptions, and ensure that KVDB can properly load AOF files to restore data at startup.

Core functions:

- Verifies the syntax correctness and integrity of AOF files, and detects whether files are corrupted;
- Repairs AOF file truncation and syntax errors caused by unexpected events such as server downtime or power failure;
- Removes invalid commands from AOF files to ensure KVDB can load them properly.

8.2.2 Tool Location and Prerequisites

8.2.2.1 Tool Location

Located in the same directory as kvdb-check-rdb, i.e. the KVDB installation directory folder:

```
Linux: /kvdb/kvdb-check-aof
```

8.2.2.2 Prerequisites

No need to start the KVDB service; can be run directly on the command line;

- Have read and write permissions for the AOF file (writing to the file is required during repair);
- The AOF file path is correct; if a relative path is used, ensure that the current terminal directory matches the file path.

8.2.3 Basic Syntax

The syntax of kvdb-check-aof is similar to kvdb-check-rdb; the core is to specify the AOF file path, and optional parameters are used for repair and debugging:

```
# Basic syntax (verify AOF file)
kvdb-check-aof [options]
/path/to/appendonlydir/appendonly.aof.manifest

# Common shorthand (without options, file integrity is verified by
default)
kvdb-check-aof appendonlydir/appendonlydir/appendonly.aof.manifest
```

8.2.4 Core Options and Parameter Descriptions

The options of kvdb-check-aof focus on repair, forced repair, and detailed output, as follows:

Option	Description	Use Case
`--fix`	Repairs a corrupted AOF file, removes invalid commands, truncates erroneous parts, and overwrites the original file after repair (backing up first is recommended)	AOF file is corrupted and KVDB fails to start
`--truncate-to-timestamp`	Precisely trims to the specified timestamp, retaining all data before that point in time	When the abnormal time point of the service is known and data before the specified time needs to be precisely retained
`--help` (abbreviated `-h`)	Displays the tool's help information, including all options and syntax	Quick lookup when you forget the command

8.2.5 Common Operation Examples

8.2.5.1 Verifying AOF File Integrity

Directly specify the AOF file path, and the tool automatically verifies the file syntax and integrity:

```
# Verify the AOF file
kvdb-check-aof /path/to/appendonlydir/appendonly.aof.manifest
```

Normal output with aof-use-rdb-preamble no (file is intact):

```
Start checking Multi Part AOF
Start to check BASE AOF (RESP format).
AOF analyzed:
filename=appendonly.aof.1.base.aof,size=127,ok_up_to=127,ok_up_to_line=
BASE AOF appendonly.aof.1.base.aof is valid
Start to check INCR files.
INCR AOF appendonly.aof.1.incr.aof is empty
All AOF files and manifest are valid
```

Normal output with aof-use-rdb-preamble yes (file is intact):

```
Start checking Multi Part AOF
Start to check BASE AOF (RDB format).
[offset 0] Checking RDB file appendonlydir/appendonly.aof.1.base.rdb
[offset 25] AUX FIELD kvdb-ver = '2.0'
[offset 38] AUX FIELD kvdb-bits = '64'
[offset 50] AUX FIELD ctime = '1770690624'
[offset 65] AUX FIELD used-mem = '836208'
[offset 81] AUX FIELD aof-preamble = '1'
[offset 83] Selecting DB ID 0
[offset 151] Checksum OK
[offset 151] \o/ RDB looks OK! \o/
[info] 2 keys read
[info] 0 expires
[info] 0 already expired
RDB preamble is OK, proceeding with AOF tail...
AOF analyzed: filename=appendonly.aof.1.base.rdb, size=151,
ok_up_to=151, ok_up_to_line=1, diff=0
BASE AOF appendonly.aof.1.base.rdb is valid
Start to check INCR files.
INCR AOF appendonly.aof.1.incr.aof is empty
All AOF files and manifest are valid
```

Abnormal output with aof-use-rdb-preamble no (file is corrupted):

```
Start checking Multi Part AOF
Start to check BASE AOF (RESP format).
AOF appendonly.aof.1.base.aof format error
AOF analyzed:
filename=appendonly.aof.1.base.aof,size=194,ok_up_to=127,ok_up_to_line=
AOF appendonly.aof.1.base.aof is not valid. Use the --fix option to
try fixing it.
```

Abnormal output with aof-use-rdb-preamble yes (file is corrupted):

```
Start checking Multi Part AOF
Start to check BASE AOF (RDB format).
[offset 0] Checking RDB file appendonlydir/appendonly.aof.1.base.rdb
[offset 25] AUX FIELD kvdb-ver = '2.0'
[offset 38] AUX FIELD kvdb-bits = '64'
[offset 50] AUX FIELD ctime = '1770690624'
[offset 65] AUX FIELD used-mem = '836208'
[offset 81] AUX FIELD aof-preamble = '1'
[offset 83] Selecting DB ID 0
[offset 151] Checksum OK
[offset 151] \o/ RDB looks OK! \o/
[info] 2 keys read
[info] 0 expires
[info] 0 already expired
RDB preamble is OK, proceeding with AOF tail...
AOF appendonly.aof.1.base.rdb format error
AOF analyzed:
filename=appendonly.aof.1.base.rdb,size=167,ok_up_to=151,ok_up_to_line=
AOF appendonly.aof.1.base.rdb is not valid. Use the --fix option to
try fixing it.
```

8.2.5.2 Repairing a Corrupted AOF File

When an AOF file is corrupted, preferentially use the --fix option to repair it; be sure to back up the original file before repair:

```
# 1. Back up the original AOF file (Linux)
cp /path/to/appendonlydir /path/to/appendonlydir.bak

# 2. Regular repair
kvdb-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.aof
```

Successful repair output with aof-use-rdb-preamble no (file is corrupted):

```
Start checking Old-Style AOF
AOF appendonlydir/appendonly.aof.1.base.aof format error
AOF analyzed:
filename=appendonlydir/appendonly.aof.1.base.aof,size=146,ok_up_to=127,
This will shrink the AOF appendonlydir/appendonly.aof.1.base.aof
from 146 bytes, with 19 bytes, to 127 bytes
Continue? [y/N]: y
Successfully truncated AOF appendonlydir/appendonly.aof.1.base.aof
```

Successful repair output with aof-use-rdb-preamble yes (file is corrupted):

```
Start checking Old-Style AOF
[offset 0] Checking RDB file appendonlydir/appendonly.aof.1.base.rdb
[offset 25] AUX FIELD kvdb-ver = '2.0'
[offset 38] AUX FIELD kvdb-bits = '64'
[offset 50] AUX FIELD ctime = '1770690624'
[offset 65] AUX FIELD used-mem = '836208'
[offset 81] AUX FIELD aof-preamble = '1'
[offset 83] Selecting DB ID 0
[offset 151] Checksum OK
[offset 151] \o/ RDB looks OK! \o/
[info] 2 keys read
[info] 0 expires
[info] 0 already expired
RDB preamble is OK, proceeding with AOF tail...
AOF appendonlydir/appendonly.aof.1.base.rdb format error
AOF analyzed:
filename=appendonlydir/appendonly.aof.1.base.rdb,size=167,ok_up_to=151,
This will shrink the AOF appendonlydir/appendonly.aof.1.base.rdb
from 167 bytes, with 16 bytes, to 151 bytes
Continue? [y/N]: y
Successfully truncated AOF appendonlydir/appendonly.aof.1.base.rdb
```

8.2.6 Notes

- Backup first: before repairing an AOF file, you must back up the original file to avoid complete data loss if the repair fails;
- Repair principle: the tool repairs files by truncating invalid commands and correcting syntax errors; unrecognized commands will be discarded, and data integrity needs to be checked after repair;
- Difference from AOF rewrite: kvdb-check-aof is used to repair corrupted files, while KVDB's AOF rewrite (BGREWRITEAOF) is used to compress AOF files; the two have different functions;
- Under the appendonlydir directory, specified files can be repaired with the --fix parameter; supported files include: appendonly.aof.x.base.rdb, appendonly.aof.x.base.aof, and appendonly.aof.x.incr.aof as optional specifications.

8.3 kvdb-benchmark Tool User Manual

8.3.1 Tool Introduction

kvdb-benchmark is KVDB's built-in performance test tool. It is used to simulate multiple clients concurrently accessing the KVDB service and test KVDB's performance indicators in different scenarios, including requests per second (QPS), response time (latency), number of concurrent connections, etc., helping operations personnel and developers evaluate the carrying capacity of the KVDB service, optimize configuration parameters, and ensure that the KVDB service runs stably in the production environment.

Core functions:

- Tests the throughput (QPS) of the KVDB service under different numbers of concurrent connections and different command types;
- Evaluates the response latency of the KVDB service (minimum latency, maximum latency, average latency) and troubleshoots performance bottlenecks;
- Simulates access scenarios with different data volumes and different command combinations to verify the rationality of KVDB configurations (such as memory and number of threads);
- Compares KVDB performance differences in different environments (such as standalone and cluster) and different versions, providing a basis for deployment solutions.

Tip: kvdb-benchmark is a stress test tool. It will occupy the CPU, memory, and network resources of the KVDB service during testing. It should be used with caution in production environments. It is recommended to execute it in a test environment or during off-peak business hours to avoid affecting online business.

8.3.2 Tool Location and Prerequisites

8.3.2.1 Tool Location

Located in the same directory as the first three tools, i.e. the KVDB installation folder. No additional installation is required; it is deployed together with KVDB:

```
Linux: /kvdb/kvdb-benchmark
```

Tip: If the KVDB installation directory is added to the system environment variables, you can directly enter kvdb-benchmark in the terminal to run it without entering the full path.

8.3.2.2 Prerequisites

- The KVDB service must be started: kvdb-benchmark needs to connect to the KVDB service to execute tests; ensure that kvdb-server is running properly;
- Network connectivity: for local testing, ensure that 127.0.0.1:6359 (default port) is accessible; for remote testing, ensure that the KVDB service port (default 6359) is open and the configuration allows remote connections;
- Sufficient permissions: the user running kvdb-benchmark must have terminal execution permissions; if the KVDB service has a password set, the password must be specified during testing;
- Resource reservation: reserve sufficient CPU and memory resources before testing to avoid insufficient resources in the test environment itself, which would affect the accuracy of test results.

8.3.3 Basic Syntax

The syntax core of kvdb-benchmark is to specify connection parameters and test parameters, controlling test scenarios through options. The basic syntax is as follows:

```
# Basic syntax (connect to the KVDB service + execute tests)
kvdb-benchmark [connection options] [test options]

# Common shorthand (local default connection, no additional options,
executes default tests)
kvdb-benchmark
```

Description:

- Connection options: used to specify the host, port, password, etc. of the KVDB service; basically consistent with the connection options of kvdb-cli;

- Test options: used to control the number of concurrent connections, test commands, test duration, data volume, etc.; these are the core options of kvdb-benchmark;
- Default test: if no options are specified, the tool will use 10 concurrent connections, test 16 common KVDB commands, execute each command 10,000 times, and finally output a performance report.

8.3.4 Core Options and Parameter Descriptions

The options of kvdb-benchmark are divided into two categories: "connection options" and "test options", of which test options are the focus, as follows:

8.3.4.1 Connection Options

Option	Value Type	Description	Default
`-h <host>`	String (IP/domain name)	Specifies the host address (IP) of the KVDB service; can be omitted for local testing, must be specified for remote testing	`127.0.0.1`
`-p <port>`	Integer (port number)	Specifies the port number of the KVDB service	`6359`
`-s <socket>`	String (socket path)	Specifies the Unix socket path of the KVDB service (used when connecting via Unix socket)	None (TCP connection by default)
`-a <password>`	String	Specifies the password of the KVDB service (if the service has the `requirepass` configuration set)	None (no password by default)

8.3.4.2 Core Test Options

Option	Value Type	Description	Default
`-c <clients>`	Integer	Specifies the number of concurrently connected clients, simulating multiple users accessing at the same time	`50`
`-n <requests>`	Integer	Specifies the total number of requests for each test command, completed jointly by all clients	`100000`
`-d <size>`	Integer (bytes)	Specifies the value data size of SET/GET commands, simulating the value length of real business	`3`
`--dbnum <db>`	Integer (database	Specifies the KVDB database used for testing (SELECT command), corresponding to KVDB's	`0`

	number)	db0~db15	
`-k <boolean>`	Boolean (1/0)	Connection keepalive policy: 1 = keep connection (persistent connection), 0 = reconnect on each request (short connection), simulating real business connection modes	`1`
`-r <keyspacelen>`	Integer	Uses random key names for commands such as SET/GET/INCR, random values for SADD, and random members and scores for ZADD; automatically replaces randint in parameters with a 12-digit random number from 0 to keyspacelen-1, avoiding hot keys and fitting real scenarios	None
`-P <numreq>`	Integer	Enables pipeline requests; each pipeline packs requests for sending, simulating batch business operations (improving throughput)	None
`--help`	None	Displays the tool's help information, including all options and basic syntax	Quick lookup when you forget the command syntax
`--version`	None	Displays the kvdb-benchmark tool version, consistent with the KVDB service version	Confirm version compatibility

8.3.4.3 Advanced Extension Parameters (Multi-threading / Cluster / Special Test Modes)

Option	Value Type	Description	Default
`--threads <num>`	Integer	Enables multi-threaded mode to run the stress test tool, utilizing multi-core CPUs to improve stress test side performance (avoiding the stress test side becoming a bottleneck)	`4`
`--cluster`	None (switch parameter)	Enables cluster mode, adapts to KVDB Cluster stress testing, automatically identifies slots and distributes requests	Disabled
`-l`	None (switch parameter)	Loop test mode: runs tests indefinitely until manually stopped by pressing Ctrl+C; used for long-term stability stress testing	Disabled
`-I`	None (switch parameter)	Idle mode: only opens the N idle connections specified by -c, without sending any requests; used to test KVDB's idle connection carrying capacity	Disabled

8.3.5 Common Operation Examples (Core Focus)

The following examples cover the most commonly used test scenarios in daily work. Combined with actual business needs, they are adapted to Apusic KVDB V2.0.3, V2.0.4, and V2.0 versions and can be used directly for reference:

8.3.5.1 Local Default Test

Without specifying any options, execute the default test: 50 concurrent clients, testing 16 default commands, each command executed 100,000 times, outputting a complete performance report:

```
kvdb-benchmark
```

The default test commands include common commands such as SET, GET, INCR, LPUSH, LPOP, SADD, SPOP, etc., suitable for quickly understanding the basic performance of local KVDB.

8.3.5.2 Testing with Specified Concurrency and Request Counts

Simulate real concurrency scenarios by specifying the number of concurrent clients and the number of requests for each command; this is the most commonly used test method:

```
# Example: 50 concurrent clients, each command executed 100,000
times, testing local KVDB performance
kvdb-benchmark -c 50 -n 100000

# Example: 100 concurrent clients, each command executed 50,000
times, testing remote KVDB (with password)
kvdb-benchmark -h 192.168.1.100 -p 6359 -a your_password -c 100 -n
50000
```

8.3.5.3 Targeted Testing of Core Commands

In actual business, usually only a few KVDB commands are used. Specify commands through the -t option to avoid useless testing and improve efficiency:

```
# Example: test SET and GET commands, 50 concurrency, each command
executed 100,000 times
kvdb-benchmark -t SET,GET -c 50 -n 100000

# Example: test hash commands (HSET, HGET), 30 concurrency, value
size 1KB
kvdb-benchmark -t HSET,HGET -c 30 -n 50000 -d 1024
```

8.3.5.4 Time-Based Stress Testing for Targeted Testing of Core Commands

Simulate long-term stable access by specifying the test duration; suitable for testing the stability of Redis running for a long time:

```
# Example: loop test (manually stopped), test the SET command, 60 concurrency, value size 512 bytes
kvdb-benchmark -t SET -c 60 -1 -d 512
```

8.3.5.5 Simplified Output for Quick QPS Viewing

No detailed latency information is needed; only view the QPS of each command, suitable for quickly comparing performance differences:

```
# Example: default test, simplified QPS output
kvdb-benchmark -q

# Example: test the SET and GET commands of remote KVDB, simplified output
kvdb-benchmark -h 192.168.1.100 -a your_password -t SET,GET -c 50 -n 100000 -q
```

8.3.5.6 Test Result Interpretation (Key Indicators)

After executing the test, the tool will output a detailed report. Focus on the following 4 indicators, taking the SET command as an example:

```
SET: 100000 requests completed in 0.89 seconds
50 parallel clients
3 bytes payload
keep alive:1
host configuration "save": 3600 1 300 100 60 10000
host configuration "appendonly": no
multi-thread:no
Latencyby percentile distribution:
0.000% <= 0.103 ms (cumulative count 123)
50.000% <= 0.215 ms (cumulative count 50123)
90.000% <= 0.327 ms (cumulative count 90123)
```

```
99.000% <= 0.439 ms (cumulative count 99012)
99.900% <= 0.551 ms (cumulative count 99901)
99.990% <= 0.663 ms (cumulative count 99990)
99.999% <= 0.775 ms (cumulative count 99999)
100.000% <= 0.887 ms (cumulative count 100000)
Cumulative distribution of latencies:
0.103 ms: 123 requests (0.12%)
0.215 ms: 50123 requests (50.12%)
0.327 ms: 90123 requests (90.12%)
0.439 ms: 99012 requests (99.01%)
0.551 ms: 99901 requests (99.90%)
0.663 ms: 99990 requests (99.99%)
0.775 ms: 99999 requests (99.999%)
0.887 ms: 100000 requests (100.00%)

Summary:
throughput summary: 112359.55 requests per second
latency summary (msec):
| avg   | min   | p50   | p95   | p99   | max   |
|-------|-------|-------|-------|-------|-------|
| 0.220 | 0.080 | 0.215 | 0.383 | 0.439 | 0.887 |
```

Core indicator interpretation:

- throughput summary: QPS (requests per second), the core performance indicator; the higher the better (112359.55 QPS in the example);
- avg / min / max: average latency, minimum latency, maximum latency (milliseconds); the lower the better;
- p50 / p95 / p99: latency percentiles, representing that 50%/95%/99% of request latencies do not exceed this value (in the example, 99% of request latencies ≤ 0.439ms); percentiles better reflect real performance;
- requests completed: the total number of completed requests, consistent with the number of requests specified by -n, indicating that the test is normal.

8.3.6 Notes

- Use with caution in production environments: kvdb-benchmark will occupy a large amount of KVDB CPU, memory, and network resources, which may cause online business lag and timeouts. It is recommended to

execute it only in a test environment or during off-peak business hours, and stop it promptly after testing;

- Keep the test environment consistent with the production environment: when testing, try to ensure that the KVDB configuration (memory, number of threads, persistence method) and server hardware (CPU, memory) are consistent with the production environment, otherwise the test results have no reference value;
- Avoid a single test scenario: real business scenarios are complex; it is recommended to test multiple scenarios combining concurrency, data volume, and command types to comprehensively evaluate performance;
- Eliminate interference factors: disable KVDB persistence (appendonly no, save "") during testing to avoid persistence operations affecting test results; at the same time, close other resource-consuming processes on the server;
- Password security: when using the -a option to enter a password, the password will be displayed in plain text in the terminal history. It is recommended to clear the terminal history after testing, or hide the password by other means;
- Set test parameters reasonably: the concurrency should not be too high (beyond the server's carrying capacity), and the number of requests / test duration should not be too short (to avoid excessive fluctuation of test results). It is recommended to run the test multiple times and take the average value.

8.4 kvdb-conf-conv Tool User Manual

8.4.1 Tool Introduction

kvdb-conf-conv is KVDB's dedicated configuration file format conversion tool. Its core purpose is to convert the configuration files of KVDB V2.0.2 and earlier KVDB versions into the configuration files of Apusic KVDB V2.0, solving the problem of incompatible configuration file formats between different KVDB versions. It achieves smooth migration from KVDB V2.0.2 and earlier KVDB versions to Apusic KVDB V2.0 without manually modifying configuration items, improving the efficiency and accuracy of version migration.

Core functions:

- One-click conversion of the YAML configuration files of KVDB V2.0.2 and earlier KVDB versions into YAML configuration files compatible with Apusic KVDB V2.0, automatically adapting to configuration item formats and naming conventions;
- Supports precise conversion of two configuration types: server and sentinel, matching the feature configurations of the corresponding versions respectively;

- Retains all valid configuration items in the original configuration file, ensuring that the converted configuration can be directly loaded and used by Apusic KVDB V2.0;
- Avoids syntax errors and missing items caused by manual configuration conversion, reducing the configuration risk of version migration.

The execution logic of the configuration conversion tool is "first read the original structure and basic configuration of the template file pointed to by --c-yaml -> extract the valid configuration items of the KVDB V2.0.2 and earlier configuration file pointed to by --go-yaml -> replace the corresponding configuration items in the template file with the configuration values of KVDB V2.0.2 and earlier versions -> finally overwrite and write to the file pointed to by --c-yaml". It does not directly generate a new Apusic KVDB V2.0 configuration file from the KVDB V2.0.2 and earlier configuration. The configuration file pointed to by --c-yaml must not be an empty file.

8.4.2 Tool Location and Prerequisites

8.4.2.1 Tool Location

Located in the same directory as other KVDB supporting tools (such as kvdb-check-aof, kvdb-check-rdb), deployed together with the KVDB installation package, no additional installation is required:

```
Linux: /kvdb/kvdb-conf-conv
```

8.4.2.2 Prerequisites

- No need to start the KVDB service; can be run directly on the command line;
- Have read permission for the KVDB V2.0.2 and earlier configuration files, and read-write permissions for the target Apusic KVDB V2.0 configuration file;
- Ensure that the configuration files (YAML format) of KVDB V2.0.2 and earlier / Apusic KVDB V2.0 KVDB have correct paths, are complete and undamaged, and can be read normally.

8.4.3 Basic Syntax

The syntax rules of kvdb-conf-conv are fixed; the configuration type and the source or target configuration file path must be specified. There is no default execution logic; complete and valid parameters must be passed in to run. The basic syntax is as follows:

```
# Basic syntax (specify configuration type + source KVDB V2.0.2 and
earlier configuration + target C version configuration)
```

```
kvdb-conf-conv [--server|--sentinel] --go-yaml <kvdb_go.yaml> --c-
yaml <kvdb.yaml>
```

8.4.4 Core Options and Parameter Descriptions

The options of kvdb-check-aof focus on repair, forced repair, and detailed output, as follows:

Option	Description	Use Case
`--server`	Specifies that the configuration file to be converted is of the KVDB server type; `[--server	--sentinel]` is a mandatory either-or parameter, which cannot be omitted and cannot be used at the same time
`--sentinel`	Specifies that the configuration file to be converted is of the KVDB sentinel type; `[--server	--sentinel]` is a mandatory either-or parameter, which cannot be omitted and cannot be used at the same time
`--go-yaml`	The full path (absolute or relative) of the KVDB V2.0.2 and earlier KVDB configuration file, specifying the conversion source file	All conversion scenarios; the source file path is mandatory
`--c-yaml`	The full path (absolute or relative) of the converted C version KVDB configuration file	All conversion scenarios; the target file path is mandatory
`--help`	Displays the tool's help information, including all options and syntax	Quick lookup when you forget the command

8.4.5 Common Operation Examples

The following examples cover the conversion scenarios of the two core configuration types, server and sentinel, adapted to the Linux environment, and can be used directly for reference. It is recommended to back up the relevant configuration files before execution.

8.4.5.1 Converting KVDB V2.0.2 and Earlier KVDB Server Configuration to Apusic KVDB V2.0

Specify the --server parameter to convert the KVDB V2.0.2 and earlier server configuration file conf.yaml into the Apusic KVDB V2.0 server configuration file kvdb.yaml:

```
# Using absolute paths (recommended, no path compatibility issues)
kvdb-conf-conv --server --go-yaml /kvdb/conf.yaml --c-yaml
/kvdb/kvdb.yaml

# Using relative paths (ensure the current terminal directory is
```

```
/kvdb)
cd /kvdb
kvdb-conf-conv --server --go-yaml conf.yaml --c-yaml kvdb.yaml
```

Conversion successful:

```
Skip unsupported configuration item: ReplSlaveIgnoreMaxMemory
Skip unsupported configuration item: PoolSliceSize
Skip unsupported configuration item: PoolSliceNum
Skip unsupported configuration item: PoolNewWorkerNum
Skip unsupported configuration item: AofNoFsyncOnRewrite
Skip unsupported configuration item: GoroutineMaxCpuNum
Skip unsupported configuration item: SlowLogSlowerThan
Skip unsupported configuration item: LuaMaxLocalVarNum
Skip unsupported configuration item: LuaMaxStackSize
Skip unsupported configuration item: ReplServeStaleData
Skip unsupported configuration item: ReplPingSlavePeriod
Skip unsupported configuration item: Ip
Skip unsupported configuration item: Segment
\o/ Config file converted successfully! (conf.yaml -> kvdb.yaml)\o/
```

conf.yaml before conversion:

```yaml
onf.yaml
  Network:
    MaxClients: 10000
    # 连接在空闲超过设置的时间后，服务器会主动关闭连接，当设置为0时，不启用超时机制，timeout单位
    Timeout: 0
    # TCP keepalive，单位为秒，当为0时不设置tcp keepalive
    TcpKeepAlive: 300

  General:
    # db数量
    Databases: 16
    # 日志等级，用于过滤输出日志，包括debug, info, warn, error, fatal五个等级
    LogLevel: "info"
    # 日志文件输出目录，当设置为空字符串时，日志文件不会写入磁盘， eg: LogFile: "/tmp/server
    LogFile: "./server-6359.log"
    # license文件位置
    LicensePath: "./license.lic"

  MemoryManagement:
    # 最大内存限制，如果maxmemory值为0，表示不做限制，如果数字后没有单位，则默认单位为字节，单位
    # eg: "1gb", "1GB", "1000mb", "1000m", "1000000KB", "1000000kb", "1000000000B"
    MaxMemory: "6GB"
    # 缓存淘汰策略，noeviction, volatile-lru, allkeys-lru, volatile-random, allkeys-r
    MaxMemoryPolicy: "volatile-lru"
    # 每次缓存淘汰时的采样数量
    MaxMemorySamples: 5
    # lfu-log-factor可以调整计数器counter的增长概率，lfu-log-factor越大，counter增长概率越
    # 计算公式为： 1 / (old_value * lfu_log_factor + 1)
    LFULogFactor: 10.000000
    # lfu-decay-time是一个以分钟为单位的数值，可以调整counter的减少速度
    LFUDecayTime: 1.000000
    # 从节点是否忽略maxmemory检查
    ReplSlaveIgnoreMaxMemory: "yes"
    # PoolSliceSize
    PoolSliceSize: 40
    # PoolSliceNum
    PoolSliceNum: 40
    # PoolSliceNum
    PoolNewWorkerNum: 4

  SnapShotting:
    # save <seconds> <changes>，指定在多长时间内，有多少次更新操作后，server会将rdb文件写入
    # Save:
    #  - "3600 1"
    #  - "300 10"
    #  - "60 10000"
    # 当 Save: "" 为空时，不启用RDB自动保存
```

kvdb.yaml after conversion:

mdc.yaml

```yaml
REPLICATION:
    # 从节点向主节点公布的端口（用于 NAT/Docker 等场景）
    replica-announce-port: 0

KEYS TRACKING:
    # 客户端缓存跟踪表的最大键数，0 表示无限制
    tracking-table-max-keys: 1000000

SECURITY:
    # user: "<user> <acl rules...>"
    user: default on nopass ~* &* +@all
    # ACL 日志的最大条目数
    acllog-max-len: 128
    # 外部 ACL 用户文件路径，与配置文件内用户配置不可混用
    aclfile: ""
    # 默认用户的密码（等价于 ACL 的 default 用户密码）
    requirepass: "test123"
    # 新用户默认的 Pub/Sub 频道权限，allchannels（允许所有）、resetchannels（禁止所有）
    acl-pubsub-default: allchannels
    # 命令重命名，eg: rename-command: "flushall ''", rename-command: "flushall no-
    rename-command: ""

CLIENTS:
    # 最大并发客户端连接数，受系统文件描述符限制
    maxclients: 10000

MEMORY MANAGEMENT:
    # 内存使用上限，达到后按 maxmemory-policy 处理
    maxmemory: 6gb
    # 内存达到上限时的淘汰策略，如 volatile-lru、allkeys-lfu 等
    maxmemory-policy: volatile-lru
    # LRU/LFU/TTL 算法的采样数，越大越精确但 CPU 消耗越高
    maxmemory-samples: 5
    # 淘汰处理的激进程度，0（最低延迟）到 100（优先处理淘汰）
    maxmemory-eviction-tenacity: 10
    # 从节点是否忽略自身的 maxmemory 配置（默认由主节点处理淘汰）
    replica-ignore-maxmemory: yes
    #主动过期键的扫描力度，1-10，越大越频繁（可能增加 CPU 和延迟）
    active-expire-effort: 1

LAZY FREEING:
    # 内存淘汰时是否使用惰性删除（后台线程释放内存）
    lazyfree-lazy-eviction: no
    # 键过期时是否使用惰性删除
    lazyfree-lazy-expire: no
    # 服务器内部删除键（如 RENAME 覆盖）时是否使用惰性删除
```

8.4.5.2 Converting KVDB V2.0.2 and Earlier KVDB Sentinel Configuration to Apusic KVDB V2.0

Specify the --sentinel parameter to convert the KVDB V2.0.2 and earlier sentinel configuration file sentinel.yaml into the Apusic KVDB V2.0 sentinel configuration file sentinel.yaml:

```
# Absolute path example
kvdb-conf-conv --sentinel --go-yaml /kvdb/sentinelgo.yaml --c-yaml
/kvdb/sentinel.yaml

# Relative path example (current terminal directory is /kvdb)
kvdb-conf-conv --sentinel --go-yaml sentinelgo.yaml --c-yaml
sentinel.yaml
```

Successful repair output:

```
Config file converted successfully! (sentinelgo.yaml->sentinel.yaml)
```

sentinel_go.yaml before conversion:

```
Sentinel:
    # 绑定的ip地址，可以绑定多个ip地址，支持ipv4/ipv6地址，eg:
    # Bind:
    #  - "127.0.0.1"
    #  - "::1"
    Bind:
    - "0.0.0.0"
    # 端口号
    port: 26369
    # 日志等级，用于过滤输出日志，包括debug, info, warn, error, fatal五个等级
    loglevel: notice
    # 日志文件输出目录，当设置为空字符串时，日志文件不会写入磁盘
    logfile: ""
    # sentinel相关配置项
    SentinelItems:
    - "sentinel monitor mymaster 127.0.0.1 6369 1" # 设置监听的主节点信息，IP地址/端口
    - "sentinel down-after-milliseconds mymaster 30000" # 设置主节点主观下线的超时时间
    - "sentinel failover-timeout mymaster 180000"
    - "sentinel parallel-syncs mymaster 1"
    # - "sentinel auth-pass mymaster 123456"              # 设置哨兵访问主节点的密码,
    # - "sentinel config-epoch mymaster 0"                # 设置节点的选举纪元
    # - "sentinel leader-epoch mymaster 0"
    # - "sentinel known-replica mymaster 127.0.0.1 6380"  # 设置其他已知的从节点信息
    # - "sentinel current-epoch 0"                        # 设置当前哨兵选举纪元

SSL:
    # 是否开启ssl，如果开启则使用true，否则是false
    Enable: false
    # SSL监听端口，如果仅开启SSL监听，需要把NetWord: Port端口设置为0

    # 服务端SSL证书文件
    TlsCertFile: "./certs/ssl_tls_cert/server.crt"
    # 服务端SSL证书的密钥
    TlsKeyFile: "./certs/ssl_tls_cert/server.key"
    # 证书签发机构的证书文件，即可信从根证书
    TlsCaCertFile: "./certs/ssl_tls_cert/ca.crt"
    # 证书签发机构的证书文件夹，即多个可信的根证书，可放在一个文件夹中
    TlsCaCertDir: ""
    # 客户端SSL证书文件,为集群/主从模式使用
    TlsClientCertFile: "./certs/ssl_tls_cert/client.crt"
    # 客户端SSL证书的密钥，为集群/主从模式使用
    TlsClientKeyFile: "./certs/ssl_tls_cert/client.key"
    # 服务器是否验证客户端证书，设置为yes需要客户端提供证书，并且服务端作验证。 如果不需要验证
    # 设置为optional,即客户端可以提供证书也可以不提供证书，但是如果客户端提供证书，那么amdc服
    TlsAuthClients: "no"
    # 主从模式是否使用TLS通信，当master开启了tls模式，那么从节点必须开启，否则无法连接master
```

sentinel.yaml after conversion:

```
TLS/SSL:
    # TLSv1.3 允许的密码套件
    tls-ciphersuites: ""
    # 是否优先使用服务器的密码套件偏好
    tls-prefer-server-ciphers: no
    # 是否启用 TLS 会话缓存以加速重连
    tls-session-caching: yes
    # TLS 会话缓存的最大条目数，0 表示无限制
    tls-session-cache-size: 20480
    # TLS 会话缓存的超时时间（秒）
    tls-session-cache-timeout: 300

SENTINEL:
    # 定义要监控的主节点，格式为 <master-name> <ip> <amdc-port> <quorum>（如 mymaster
    sentinel monitor: mymaster 127.0.0.1 6369 1
    # 主从节点的认证用户名，格式为 <master-name> <username>
    sentinel auth-user: mymaster ""
    # 主从节点的认证密码，格式为 <master-name> <password>
    sentinel auth-pass: mymaster ""
    # 判定节点主观下线的超时时间（毫秒），默认 30000（30 秒）
    sentinel down-after-milliseconds: mymaster 30000
    # Sentinel 之间通信的认证用户名
    sentinel sentinel-user: ""
    # Sentinel 之间通信的认证密码其他设置
    sentinel sentinel-pass: ""
    # 故障转移时从节点同步新主节点的并发数量，默认 1
    sentinel parallel-syncs: mymaster 1
    # 故障转移超时时间（毫秒），默认 180000（3 分钟）
    sentinel failover-timeout: mymaster 180000
    # 发生警告级事件时执行的通知脚本（如邮件、短信通知），格式为 <master-name> <script-path
    sentinel notification-script: mymaster ""
    # 故障转移后通知客户端重新配置的脚本，格式为 <master-name> <script-path>
    sentinel client-reconfig-script: mymaster ""
    # 是否禁止运行时修改脚本配置，默认 yes
    sentinel deny-scripts-reconfig: yes
    # 重命名命令，格式为 <master-name> <old-command-name> <new-command-name>
    sentinel rename-command: mymaster "" ""
    # 是否支持主机名（默认 no，需依赖 DNS）
    sentinel resolve-hostnames: no
    # 是否对外公布主机名（默认 no）
    sentinel announce-hostnames: no
    # NAT/Docker 环境下对外公布的 IP
    sentinel announce-ip: ""
    # NAT/Docker 环境下对外公布的端口
    sentinel announce-port: 0
```

8.4.6 Notes

- Template file is directly overwritten: after the tool completes the configuration item replacement, it will directly overwrite the file pointed to by --c-yaml without any pop-up window or command-line prompt. Be sure to make a complete backup of the template file before conversion to avoid losing the original Apusic KVDB V2.0 configuration structure or basic items;
- Strictly follow the parameter rules: specify either --server or --sentinel configuration type; it cannot be omitted and cannot be used at the same time; --go-yaml (KVDB V2.0.2 and earlier source configuration) and --c-yaml (Apusic KVDB V2.0 replacement template) are mandatory parameters and must be immediately followed by the actual file path. Missing parameters, format errors, or invalid paths will directly trigger a tool error and exit;
- Conversion direction and type are unique: this tool only supports one-way replacement from KVDB V2.0.2 and earlier KVDB configuration -> Apusic KVDB V2.0 KVDB configuration. It does not support reverse conversion from Apusic KVDB V2.0 to KVDB V2.0.2 and earlier versions, nor is it compatible with configuration replacement of other KVDB versions; and -server or --sentinel must be consistent with the configuration type of the source file and the template file (server configuration can only replace the server template, sentinel configuration can only replace the sentinel template);
- Validity must be verified after replacement: after the replacement is complete (no error output), first check whether the --c-yaml file is properly overwritten and whether the configuration items have been replaced, then start the corresponding type of Apusic KVDB V2.0 KVDB (server or sentinel) to load the file, and confirm that the service can start normally without configuration-related errors, avoiding service startup failure caused by abnormal configuration item matching.

8.5 Troubleshooting Common Problems

8.5.1 Tool Cannot Run

Problem 1: entering kvdb-check-rdb in the terminal prompts "command not found" (Linux);

- Cause: the tool path has not been added to the system environment variables;
- Solution: run with the full path (e.g. /usr/local/kvdb/kvdb-check-rdb), or add the environment variable (echo "export PATH=/usr/local/kvdb/:$PATH" >> /etc/profile, source /etc/profile).

8.5.2 Verification / Repair File Failure

Problem 1: kvdb-check-rdb or kvdb-check-aof prompts "Permission denied";

- Cause: no file read/write permission;

- Solution: use chmod to modify file permissions (chmod 644 dump.rdb), or run the tool with sudo (sudo kvdb-check-rdb --fix dump.rdb).

Problem 2: KVDB still fails to start after repair;

- Cause: the file is too severely corrupted, or the tool version does not match the KVDB version;
- Solution: restore from the backup file, or upgrade or downgrade the tool version to be consistent with the KVDB service version.

8.5.3 kvdb-cli Cannot Connect to the KVDB Service

Problem 1: connection prompts "Could not connect to KVDB at 127.0.0.1:6359: Connection refused";

- Cause: the KVDB service is not started;
- Solution: start the KVDB service (kvdb-server /path/to/kvdb.yaml, Linux).

Problem 2: remote connection prompts "Could not connect to KVDB at 192.168.1.100:6359:Connection timed out";

- Cause: the firewall does not open port 6359, or the KVDB configuration does not allow remote connections;
- Solution: open the firewall port (firewall-cmd --add-port=6359/tcp --permanent, Linux), modify the KVDB configuration (bind 0.0.0.0, protected-mode no), and restart the KVDB service.

Problem 3: after connecting, executing a command prompts "NOAUTH Authentication required";

- Cause: the KVDB service has a password set, but it was not entered when connecting;
- Solution: enter the password with the -a option when connecting, or enter the AUTH password in interactive mode.

8.6 Summary

This document describes in detail how to use KVDB's four core tools. The key summary is as follows:

- kvdb-check-rdb: focuses on the verification and repair of RDB files; the core option is --fix; be sure to back up before repair;
- kvdb-check-aof: focuses on the verification and repair of AOF files; supports regular repair (--fix); risks need to be handled with caution;

- kvdb-benchmark: KVDB's built-in performance test tool, used to test the throughput, response time, and other indicators of the KVDB service. The core is to control test scenarios through options to adapt to different performance evaluation needs.
- kvdb-conf-conv: kvdb-conf-conv is KVDB's dedicated configuration file format conversion tool, used to convert the configuration files of KVDB V2.0.2 and earlier KVDB versions into the configuration files of Apusic KVDB V2.0.

The above four tools are all built into KVDB and require no additional installation. They need to be mastered proficiently in daily operations, especially kvdb-cli and kvdb-benchmark, which can efficiently complete various KVDB operations, troubleshooting, and performance evaluation. During use, be sure to pay attention to data security and test specifications to avoid data loss or service abnormalities caused by misoperations.

9 Solutions to Common Problems

9.1 Problem 1: Port Occupation and Solutions

Installing KVDB will occupy some ports, and at the same time, client links need to rely on the internal DNS service. If it cannot be guaranteed that the ports are not in use, or the installation machine environment lacks a valid DNS, startup will fail. Ports occupied by the product by default:

- KVDB key-value core default port: 6359
- KVDB sentinel service default port: 26359

If the above ports are occupied before startup, startup will fail. You can switch ports or stop the program using the port.

9.2 Modifying the KVDB Key-Value Core Port

Modify the configuration file: vim kvdb.yaml, the port field

> *For more configuration, please refer to: [Cache Configuration (kvdb.yaml)]*

9.3 ACL File Loading

Store the ACL file in the directory specified by the "ACLFile" configuration item of the KVDB cache, and execute ACL load on the KVDB cache command line to reload the user configuration in the ACL file.

Notes:

1. Reloading the ACL file will replace the existing ACL rules in the cache.
2. If one or more items in the ACL file are invalid data, the entire file will fail to load, and the existing rules in the cache system will continue to be used.

9.4 Resolving KVDB Master-Slave Failover

1. If a slave fails in the key-value core cluster, the system will automatically reconnect If the Slave fails and goes down, it will automatically reconnect after returning to normal. After the Master receives the connection from the Slave, it sends its complete data file to the Slave. If the Master receives synchronization requests from multiple Slaves at the same time, the Master will only start one process in the background to save the data file, and then send it to all Slaves to ensure the Slaves are normal.
2. If the master fails in the key-value core cluster, manually restarting the master service enables the client to read and write the system When the key-value core master fails, client reads from the slave services are still

supported. The master service needs to be manually restarted. After the master service is restarted and the system returns to normal, client reads and writes to the master service are supported again.

9.5 Request Latency Issues

If KVDB requests experience obvious latency, you can identify the specific problem in the following ways:

1. slowlog command: this command can query commands with long execution times (the time length can be configured through the slowlog parameter in the kvdb.yaml file), so as to better track the code execution process.
2. –bigkeys: use the kvdb-cli client, connect and add the –bigkeys parameter to query the largest values, so as to determine whether the request latency is caused by overly large values.
3. –memkeys: same as above; this parameter queries the overall memory usage of key-value pairs, so as to determine whether the request latency is caused by overly large key-value pairs.

9.6 How to Detect Which Commands Were Executed

Use kvdb-cli to connect, and then use the monitor command to monitor which commands KVDB has continuously received. You can also use the following command to record to monitor.log in the background:

```
 kvdb-cli -h [ip] -p [port] [-a password] monitor >monitor.log 2>&1
&
```

When you want to stop recording monitor.log, use the following command:

```
 ps -ef | grep kvdb-cli | grep -v grep | awk '{print $2}' | xargs
kill -9
```

9.7 Process Is Running but Cannot Connect

When the number of clients exceeds the Linux system limit or the memory used exceeds the license memory limit, the process may still be running, but connections time out / cannot be established.

1. Clients exceed the Linux system limit: Some Linux systems default the client connection limit to 1024. When there are too many business applications and more than 1024 client connections are generated, the excess part will be restricted by the Linux system and cannot continue to connect.

   Solution: permanently set the user-level file descriptor limit by adding the following to /etc/security/limits.conf:

```
* soft nofile 65535
* hard nofile 65535
```

Note: different Linux systems may have different modification methods. If the above does not take effect, please search for the TCP connection limit of the corresponding system yourself.

2. Memory used exceeds the license memory limit: the default memory limit of the license is generally 6G (in most cases). When the business system writes more than 6G of data and continues to write, KVDB will frequently trigger the memory eviction algorithm, causing KVDB performance degradation. Under continuous heavy concurrency, some requests will time out, resulting in errors on the business side.

Solution: scale-out can solve the corresponding problem. There are two methods of scale-out: one is to apply for a license file with a larger memory, directly expanding the memory available to the node (it is not recommended to exceed 16G; too large will lead to lower query efficiency); the other is to use a distributed cluster mode (corresponding to Redis's Cluster mode).

9.8 dir Path Issue

The path configured by the "dir" configuration item in the configuration file will, after the program finishes reading the configuration file, become the program's current execution file path, not the file path of the kvdb-server executable file. Please note the following: Method 1: fill in certificate paths, log file paths, etc. uniformly with absolute paths, which is stable and reliable. Method 2: if you still need to use relative paths, please check the paths in the configuration, and modify the paths of the corresponding certificates, logs, etc. based on the path configured by the "dir" configuration item.

9.9 Password Configuration Issues

When using kvdb master-slave mode, we know that requirepass needs to be configured to set a password for the nodes.

Note here that the masterauth configuration item of the slave nodes must also be filled in with the requirepass of the master node, so that the slave nodes and the master node can interact normally. Otherwise, if the master node has requirepass set but the slave nodes have not set masterauth, or the password is wrong, the replication connection will be immediately rejected by the master node, and (error) NOAUTH Authentication required. or (error) ERR invalid password errors will appear in the log.

--It is recommended that whether it is a cluster service or a master-slave service, and whether it is a master node or a slave node, requirepass and masterauth should be fully configured to prevent the authentication of the old master from becoming invalid after a master-slave switch.

Similarly, sentinel monitoring also requires configuring the node password.

Sentinels need to continuously monitor all master-slave nodes and perform automatic failover when the master node fails. Therefore, sentinels must also be able to "enter" these password-protected nodes. This is achieved through the sentinel auth-pass directive in the sentinel's own configuration file: sentinel auth-pass

9.10 Log Configuration Issues

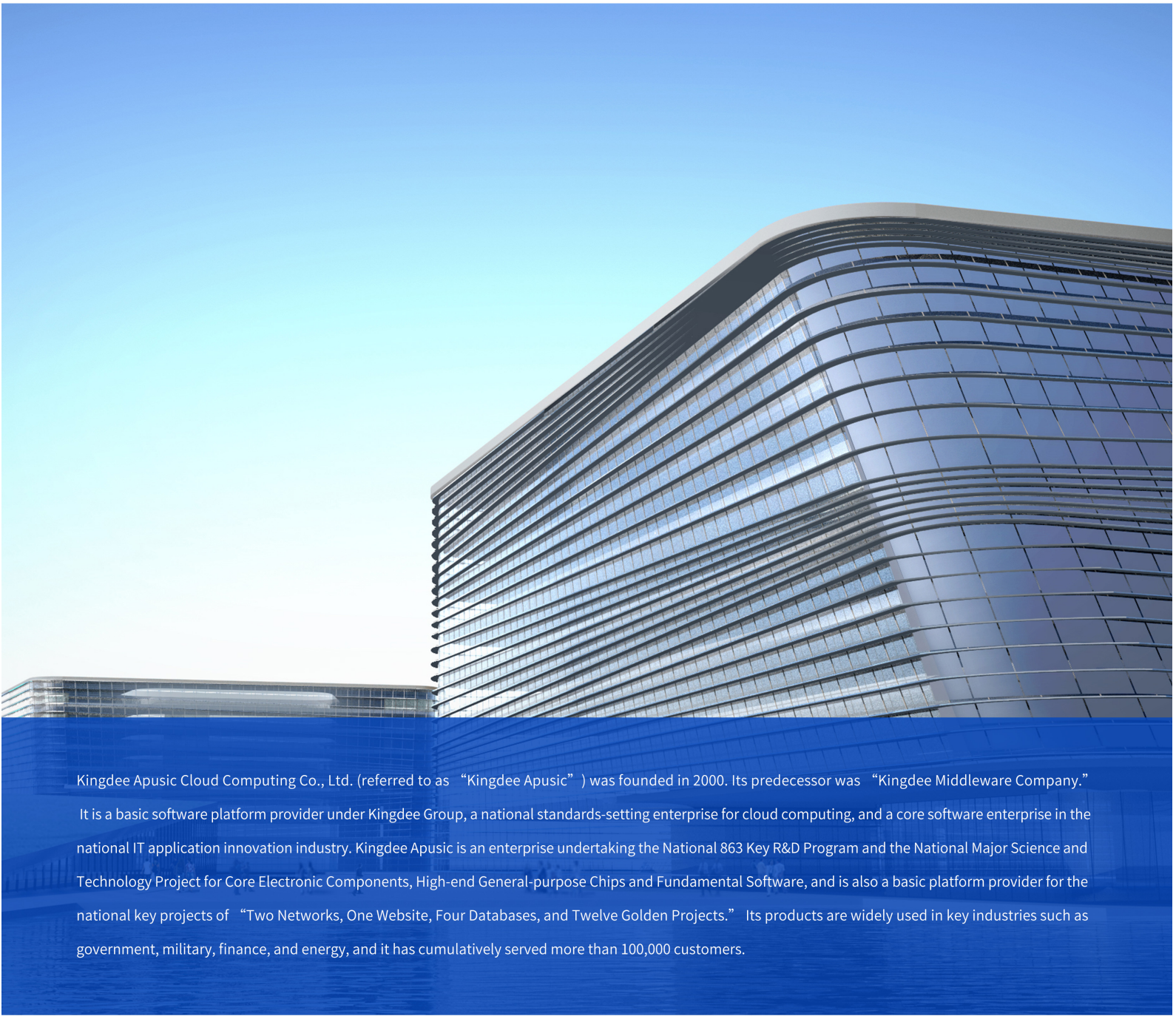
The system can perform log configuration management of kvdb's server.log on the machine, and do simple processing of the logs written to server.log by default in combination with its own business. This can make the logs more personalized and easier to use.

9.11 Master-Slave Buffer Configuration Issues

repl-backlog-size is the size of the master-slave buffer. The default configuration is 1mb, which cannot provide buffering for master-slave services with large amounts of data interaction. It is recommended to increase it to 128mb.

UNIFIED SERVICE HOTLINE

4008-555-800



Kingdee Apusic Cloud Computing Co., Ltd. (referred to as "Kingdee Apusic") was founded in 2000. Its predecessor was "Kingdee Middleware Company." It is a basic software platform provider under Kingdee Group, a national standards-setting enterprise for cloud computing, and a core software enterprise in the national IT application innovation industry. Kingdee Apusic is an enterprise undertaking the National 863 Key R&D Program and the National Major Science and Technology Project for Core Electronic Components, High-end General-purpose Chips and Fundamental Software, and is also a basic platform provider for the national key projects of "Two Networks, One Website, Four Databases, and Twelve Golden Projects." Its products are widely used in key industries such as government, military, finance, and energy, and it has cumulatively served more than 100,000 customers.

Apusic 金蝶天燕

National standard setter for cloud computing

Basic software enterprise under Kingdee Group

Core enterprise for IT application innovation

Website:www.apusic.com