Generated Date: 2026-10-23

Operations Guide

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 Target Audience
  - 1.2 Related Documentation
  - 1.3 Technical Support
- 2 Daily Operations
  - 2.1 Service Start/Stop Management
    - 2.1.1 Normal Shutdown
    - 2.1.2 Forced Shutdown
  - 2.2 License Management
- 3 Health Inspection Checklist
  - 3.1 Basic Connectivity Check
  - 3.2 Master-Slave Replication Deployment
  - 3.3 Memory Usage
  - 3.4 Client Connections
  - 3.5 Persistence Status
- 4 Performance Monitoring and Optimization
  - 4.1 Benchmark Testing
  - 4.2 Slow Query Analysis
  - 4.3 Big Key and Hot Key Detection
  - 4.4 Real-Time Monitoring
  - 4.5 Monitoring Platform Integration (Recommended)
- 5 Troubleshooting and Emergency Handling
  - 5.1 Common Problem Diagnosis
  - 5.2 Key Error Log Keywords
  - 5.3 Emergency Recovery Procedures
    - 5.3.1 Service Unavailable
    - 5.3.2 Data File Corruption
- 6 Automated Health Checks
  - 6.1 Health Check Script
  - 6.2 Scheduled Tasks
  - 6.3 Recommended Monitoring Alert Thresholds
  - 6.4 Regular Maintenance Plan

- 7 Software Management
  - 7.1 Configure a System Service for Auto-Start on Boot
  - 7.2 Build a Docker Image
- 8 Appendix: Key Configuration Optimization Recommendations

1 Preface

This document is the operations guide for Apusic Key-Value Database (KVDB) V2.0, aiming to provide middleware operations personnel with a standardized operations guide for Apusic KVDB V2.0, covering core scenarios such as installation and deployment, configuration optimization, daily operations, troubleshooting, and security hardening, to ensure stable, efficient, and secure operation of KVDB product services.

1.1 Target Audience

This document is intended for KVDB product operations engineers, IT system operations engineers, development engineers, software architects, R&D managers, and other personnel.

1.2 Related Documentation

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

No.	Document	Description
1	Apusic Key-Value Database V2.0 Quick Start Guide	A brief introduction on how to quickly get started with KVDB.
2	Apusic Key-Value Database V2.0 Installation Guide	Detailed introduction on how to install KVDB on various operating systems, KVDB service start/stop operations, and the product registration process.
3	Apusic Key-Value Database V2.0 Key-Value Core User Manual	Detailed introduction on how to use, configure, and manage KVDB features and supporting tools.
4	Apusic Key-Value Database V2.0 Console User Manual	Detailed instructions on using and operating KVDB console features.
5	Apusic Key-Value Database V2.0 Development Guide	Detailed instructions for developing KVDB client applications using various programming languages.
6	Apusic Key-Value Database V2.0 Migration Guide	Detailed instructions for migrating KVDB from historical versions to V2.0, as well as migrating from Redis to KVDB.
7	Apusic Key-Value Database V2.0 Operations Guide	Detailed instructions for KVDB monitoring, operations, security hardening, and other operational procedures.

8	Apusic Key-Value Database V2.0 Performance Tuning Guide	Detailed instructions for KVDB performance tuning.

1.3 Technical Support

KVDB products provide comprehensive technical support services. You can obtain technical support through the following channels:

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

When requesting technical support, please provide the following information:

1. Your name
2. Company information and contact details
3. Operating system and version
4. Product version number
5. Detailed information including logs, screenshots, etc. of any exceptions or errors

2 Daily Operations

2.1 Service Start/Stop Management

2.1.1 Normal Shutdown

```
# Command line method
./kvdb-cli -h [IP] -p 6359 -a [password] shutdown

# Interactive client
127.0.0.1:6359> shutdown
```

2.1.2 Forced Shutdown

```
ps -ef | grep kvdb        # Find the PID
kill -9 [PID]             # Force termination (may cause data loss)
```

2.2 License Management

- Hot update: Directly replace `license.lic`, `license.xml`, or `acls.properties`; the system loads them automatically.
- Priority order:
  1. `acls.properties` (unified authorization center)
  2. `license.lic` (KBC authorization)
  3. `license.xml` (local authorization)

3 Health Inspection Checklist

Perform the following checks regularly; it is recommended to incorporate them into automation scripts.

3.1 Basic Connectivity Check

Mode	Command	Expected Result
Standalone	`./kvdb-cli -h [IP] -p 6359 -a [password] info`	Contains `kvdb_version:2.0`
Sentinel	`./kvdb-cli -h [sentinel_ip] -p 26359 -a [password] info sentinel`	`sentinel_masters:1`, master-slave status normal
Cluster	`./kvdb-cli -h [IP] -p 6359 -a [password] cluster info`	`cluster_state:ok`, slots fully assigned

3.2 Master-Slave Replication Deployment

```
./kvdb-cli -h [master_ip] -p 6359 -a [password] info replication
```

- Key metrics:
  - `role:master`
  - `connected_slaves` = expected number
  - All `slaveX.state=online`
  - `lag ≤ 5s` (alert triggered if >30s)

3.3 Memory Usage

```
./kvdb-cli -h [IP] -p 6359 -a [password] info memory
```

- Watch:
  - `used_memory_human`
  - `mem_fragmentation_ratio` (ideal value: 1.0–1.5)

3.4 Client Connections

```
./kvdb-cli -h [IP] -p 6359 -a [password] info clients
```

- Watch:
  - `connected_clients` (alert if >80% of `maxclients`)
  - `blocked_clients ≈ 0`

3.5 Persistence Status

```
./kvdb-cli -h [IP] -p 6359 -a [password] info persistence
```

- RDB: `rdb_last_bgsave_status=ok`
- AOF (if enabled): `aof_enabled=1` and `aof_last_bgrewrite_status=ok`

4 Performance Monitoring and Optimization

4.1 Benchmark Testing

```
./kvdb-benchmark -h [IP] -p 6359 -a [password] -c 50 -n 10000 --csv
```

4.2 Slow Query Analysis

```
./kvdb-cli -h [IP] -p 6359 -a [password] slowlog get 10
```

- Recommended configuration:

  ```
  slowlog-log-slower-than 10000  # 10ms
  slowlog-max-len 128
  ```

4.3 Big Key and Hot Key Detection

```
./kvdb-cli -h [IP] -p 6359 -a [password] --bigkeys   # Big values
./kvdb-cli -h [IP] -p 6359 -a [password] --memkeys   # Big-memory
keys
./kvdb-cli -h [IP] -p 6359 -a [password] --hotkeys   # Hot keys
(requires LFU policy)
```

4.4 Real-Time Monitoring

```
./kvdb-cli --stat          # Dynamic metrics
./kvdb-cli monitor > log   # Command auditing (use with caution;
high performance overhead)
```

4.5 Monitoring Platform Integration (Recommended)

- Prometheus + Grafana: Collect metrics via `redis_exporter`
- ELK: Elasticsearch, Logstash, Kibana (ELK), centralized log analysis

```yaml
# prometheus.yml example
scrape_configs:
  - job_name: 'kvdb'
    static_configs:
      - targets: ['redis-exporter:9121']
    metrics_path: /scrape
    params:
      target: ['redis://kvdb-host:6359']
```

5 Troubleshooting and Emergency Handling

5.1 Common Problem Diagnosis

Problem	Symptom	Troubleshooting Steps
Master-slave synchronization delay	The `lag` value of the slave node keeps growing	1. Check network connectivity and bandwidth between master and slave nodes 2. Check whether `repl_backlog_size` on the master node is too small 3. Check whether disk I/O performance of the slave node is a bottleneck 4. Check the KVDB logs for replication-related errors (e.g., "MASTER timeout")
Abnormal cluster status	`cluster_state:fail`	1. Check network connectivity between all nodes 2. Confirm that all master nodes are online 3. Check whether slot assignment is complete (`cluster_slots_ok == 16384`) 4. Check `cluster info` and cluster message statistics in the logs
Insufficient memory	Slow response, large numbers of keys evicted	1. Check whether the `maxmemory` configuration is reasonable 2. Analyze memory usage patterns (`info memory`) 3. Confirm whether the eviction policy (e.g., `allkeys-lru`) is appropriate 4. Use `--bigkeys` or `--memkeys` to identify big keys

5.2 Key Error Log Keywords

- `"No valid license file found"` → License missing
- `"MASTER timeout"` → Master-slave disconnection
- `"Node timeout"` → Cluster node lost

5.3 Emergency Recovery Procedures

5.3.1 Service Unavailable

1. Confirm the scope of impact (single node or cluster)
2. Check the process: `ps -ef | grep kvdb`
3. Check resources: CPU/memory/disk
4. Try a graceful restart → force restart if necessary

5.3.2 Data File Corruption

- RDB repair: `./kvdb-check-rdb file.rdb`
- AOF repair: `./kvdb-check-aof --fix file.aof`
- Backup recovery:
    1. Stop the service
    2. Replace the files
    3. Start and verify the data

6 Automated Health Checks

6.1 Health Check Script

You can write a shell script to check the health status of the KVDB service, as shown in the following sample: `kvdb_health_check.sh`

```
#!/bin/bash
HOST="127.0.0.1"; PORT="6359"; PWD="your_password"  # Please replace
with the actual password

./kvdb-cli -h $HOST -p $PORT -a $PWD ping >/dev/null || { echo
"ERROR"; exit 1; }

echo "Memory usage: $(./kvdb-cli -h $HOST -p $PORT -a $PWD info
memory | grep used_memory_human)"
echo "Client connections: $(./kvdb-cli -h $HOST -p $PORT -a $PWD
info clients | grep connected_clients)"
echo "Slow query count: $(./kvdb-cli -h $HOST -p $PORT -a $PWD
slowlog len)"
```

6.2 Scheduled Tasks

You can use `crontab` to implement scheduled health checks, log rotation, big Key scans, and other tasks for the KVDB service.

```
# Hourly health check
0 * * * * /opt/kvdb/scripts/kvdb_health_check.sh

# Daily early-morning big Key scan
0 2 * * * /opt/kvdb/bin/kvdb-cli -h [IP] -p 6359 -a [password] --
bigkeys > /logs/bigkeys_$(date +\%Y\%m\%d).log
```

6.3 Recommended Monitoring Alert Thresholds

Metric	Warning Threshold	Critical Threshold
Memory usage	70%	85%
Client connections	80% of maxclients	90%
Replication lag	5 seconds	30 seconds
Slow query volume	10 entries/hour	50 entries/hour
CPU usage	70%	90%

6.4 Regular Maintenance Plan

- Daily: Health checks, log rotation
- Weekly: Stress testing, big Key analysis
- Monthly: Configuration review, security audit
- Quarterly: Disaster recovery drills, capacity planning

7 Software Management

7.1 Configure a System Service for Auto-Start on Boot

1. Create the service file:

```
sudo vim /etc/systemd/system/kvdb.service
```

2. Write the configuration (adjust according to actual paths, including client connection IP and password):

```
[Unit]
Description=KVDB Service
After=network.target

[Service]
Type=simple
User=kvdb
Group=kvdb
WorkingDirectory=/opt/kvdb
ExecStart=/opt/kvdb/bin/kvdb-server /opt/kvdb/kvdb.yaml
ExecStop=/opt/kvdb/bin/kvdb-cli -h 127.0.0.1 -p 6359 -a
your_password shutdown  # Please replace with the actual
password
Restart=on-failure
RestartSec=10
LimitNOFILE=65536

[Install]
WantedBy=multi-user.target
```

3. Enable the service:

```
sudo systemctl daemon-reload
sudo systemctl enable --now kvdb.service
```

7.2 Build a Docker Image

1. Prepare the installation package: `KVDB-V2.0-xxxx-amd64.tar.gz`
2. Write the `Dockerfile`:

```
FROM ubuntu:20.04
ADD KVDB-V2.0-xxxx-amd64.tar.gz /opt/
WORKDIR /opt/kvdb
EXPOSE 6359
CMD ["/opt/kvdb/bin/kvdb-server", "/opt/kvdb/kvdb.yaml"]
```

3. Build and run:

```
docker build -t apusic/kvdb:v2.0 .
docker run -d \
  --name kvdb \
  -p 6359:6359 \
  -v ./license.lic:/opt/kvdb/license.lic \
  -v ./kvdb.yaml:/opt/kvdb/kvdb.yaml \
  -v ./data:/opt/kvdb/data \
  apusic/kvdb:v2.0
```

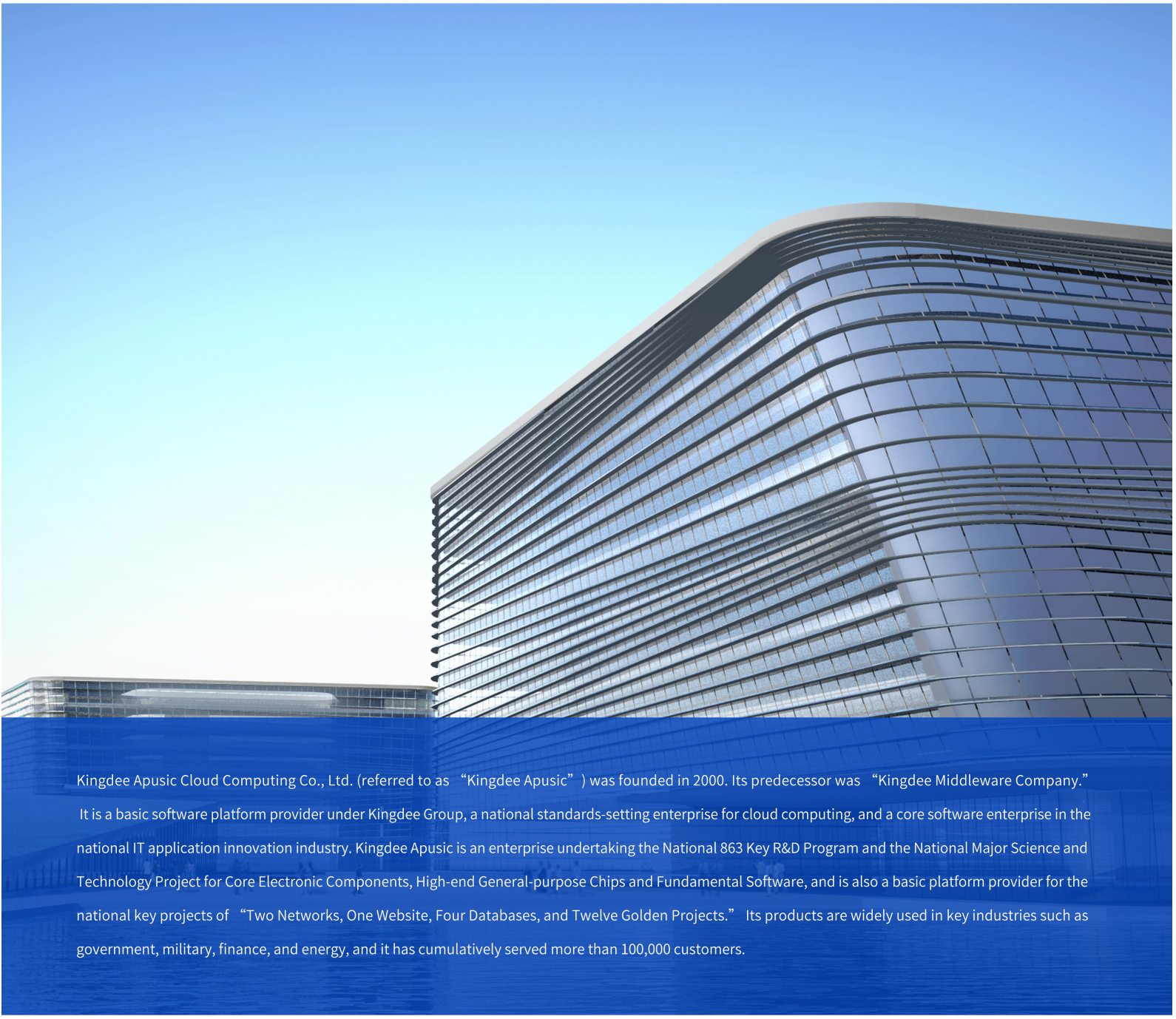
> *Note: In production environments, be sure to mount the license, configuration, and persistence directories as volumes.*

8 Appendix: Key Configuration Optimization Recommendations

Configuration Item	Recommended Value	Description
`maxmemory`	70% of physical memory	Avoid OOM
`maxmemory-policy`	`allkeys-lru` or `allkeys-lfu`	Reasonable eviction
`timeout`	300	Idle connection timeout (seconds)
`tcp-keepalive`	60	TCP keepalive detection (seconds)
`slowlog-log-slower-than`	10000	Slow query threshold (microseconds)

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