



APUSIC
固若长城
睿比世界

Operations Manual

金蝶Apusic分布式缓存 V2.0.4

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 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 Check List
 - 3.1 Basic Connection Check
 - 3.2 Master-Slave Replication Check
 - 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 Response
 - 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 Check
 - 6.1 Health Check Script
 - 6.2 Scheduled Tasks
 - 6.3 Monitoring Alert Threshold Recommendations
 - 6.4 Regular Maintenance Plan

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

1 Preface

This document is the operations manual for Apusic In-Memory Data Cache (AMDC) V2.0.4, designed to provide middleware operations personnel with standardized operational guidance for AMDC v2.0.4, covering core scenarios such as installation and deployment, configuration optimization, daily operations, troubleshooting, and security hardening, ensuring stable, efficient, and secure operation of AMDC product services.

1.1 Target Audience

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

1.2 Related Documentation

For more information about AMDC V2.0.4, please refer to the following AMDC V2.0.4 product manual documentation set:

No.	Document	Description
1	Apusic In-Memory Data Cache V2.0.4 Quick Start Guide	A brief introduction on how to quickly get started with AMDC.
2	Apusic In-Memory Data Cache V2.0.4 Installation Guide	Detailed introduction on how to install AMDC on various operating systems, AMDC service start/stop operations, and product registration process.
3	Apusic In-Memory Data Cache V2.0.4 Core User Manual	Detailed introduction on how to use, configure, and manage AMDC features and supporting tools.
4	Apusic In-Memory Data Cache V2.0.4 Console User Manual	Detailed instructions on using and operating AMDC console features.
5	Apusic In-Memory Data Cache V2.0.4 Development Guide	Detailed instructions for developing AMDC client applications using various programming languages.
6	Apusic In-Memory Data Cache V2.0.4 Migration Guide	Detailed instructions for migrating AMDC from historical versions to V2.0.4, as well as migrating from Redis to AMDC.
7	Apusic In-Memory Data Cache V2.0.4 Operations Guide	Detailed instructions for AMDC monitoring, operations, security hardening, and other operational procedures.
8	Apusic In-Memory Data Cache V2.0.4 Performance Tuning	Detailed instructions for AMDC performance tuning.

1.3 Technical Support

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

- Website: 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
./amdc-cli -h [IP] -p [port] -a [password] shutdown

# Interactive client
127.0.0.1:6359> shutdown
```

2.1.2 Forced Shutdown

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

2.2 License Management

- Hot Update: Directly replace `license.lic` , `license.xml` or `acls.properties` , system will automatically load.
- Priority Order:
 1. `acls.properties` (Unified Authorization Center)
 2. `license.lic` (KBC Authorization)
 3. `license.xml` (Local Authorization)

3 Health Check List

Perform the following checks regularly, recommended to be included in automated scripts.

3.1 Basic Connection Check

Mode	Command	Expected Result
Standalone	<code>./amdc-cli -h [ip] -p 6359 -a [pwd] info</code>	Contains <code>amdc_version:2.0.4</code>
Sentinel	<code>./amdc-cli -h [sentinel_ip] -p 26359 -a [pwd] info sentinel</code>	<code>sentinel_masters:1</code> , master-slave status normal
Cluster	<code>./amdc-cli -h [ip] -p 6359 -a [pwd] cluster info</code>	<code>cluster_state:ok</code> , slots fully allocated

3.2 Master-Slave Replication Check

```
./amdc-cli -h [master_ip] -p 6359 -a [pwd] info replication
```

- Key indicators:
 - `role:master`
 - `connected_slaves` = expected count
 - All `slaveX.state=online`
 - `lag ≤ 5s` (>30s triggers alert)

3.3 Memory Usage

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

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

3.4 Client Connections

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

- Monitor:
 - `connected_clients` (>80% of `maxclients` triggers alert)
 - `blocked_clients` $\approx$ 0

3.5 Persistence Status

```
./amdc-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

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

Output displayed as follows:

4.2 Slow Query Analysis

```
./amdc-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

```
./amdc-cli -h [IP] -p 6359 -a [password] --bigkeys # Big Value
./amdc-cli -h [IP] -p 6359 -a [password] --memkeys # Large Memory
Key
./amdc-cli -h [IP] -p 6359 -a [password] --hotkeys # Hot Key
(requires LFU policy)
```

4.4 Real-time Monitoring

```
./amdc-cli --stat # Dynamic metrics
./amdc-cli monitor > log # Command audit (use with caution, high
performance overhead)
```

4.5 Monitoring Platform Integration (Recommended)

- Prometheus + Grafana: Collect metrics via `redis_exporter`
- ELK: Centralized log analysis

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

5 Troubleshooting and Emergency Response

5.1 Common Problem Diagnosis

Problem	Symptom	Troubleshooting Steps
Master-slave sync delay	Slave node <code>lag</code> value keeps increasing	1. Check network connectivity and bandwidth between master and slave nodes 2. Check if <code>repl_backlog_size</code> on master node is too small 3. Check if slave node disk I/O performance is bottlenecked 4. Check AMDC logs for replication-related errors (e.g. "MASTER timeout")
Cluster status abnormal	<code>cluster_state:fail</code>	1. Check if all nodes can communicate with each other 2. Confirm all master nodes are online 3. Check if slot allocation is complete (<code>cluster_slots_ok == 16384</code>) 4. Check <code>cluster info</code> and cluster message statistics in logs
Insufficient memory	Slow response, large number of keys evicted	1. Check if <code>maxmemory</code> configuration is reasonable 2. Analyze memory usage pattern (<code>info memory</code>) 3. Confirm if eviction policy (e.g. <code>allkeys-lru</code>) is appropriate 4. Use <code>--bigkeys</code> OR <code>--memkeys</code> to identify big keys

5.2 Key Error Log Keywords

- "No valid license file found" → Missing license
- "MASTER timeout" → Master-slave disconnection
- "Node timeout" → Cluster node lost contact

5.3 Emergency Recovery Procedures

5.3.1 Service Unavailable

1. Confirm scope of impact (single point or cluster)
2. Check process: `ps -ef | grep amdc`
3. Check resources: CPU/memory/disk

4. Attempt graceful restart → Force restart if necessary

5.3.2 Data File Corruption

- RDB Repair: `./amdc-check-rdb file.rdb`
- AOF Repair: `./aof-check-aof --fix file.aof`
- Backup Recovery:
 1. Stop service
 2. Replace files
 3. Start and verify data

6 Automated Health Check

6.1 Health Check Script

You can write a shell script to check the health status of AMDC service, as shown in the following example:

`amdc_health_check.sh`

```
#!/bin/bash
HOST="127.0.0.1"; PORT="6359"; PWD="your_password"

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

echo "Memory Usage: $(./amdc-cli -h $HOST -p $PORT -a $PWD info
memory | grep used_memory_human) "
echo "Client Connections: $(./amdc-cli -h $HOST -p $PORT -a $PWD
info clients | grep connected_clients) "
echo "Slow Query Count: $(./amdc-cli -h $HOST -p $PORT -a $PWD
slowlog len) "
```

6.2 Scheduled Tasks

You can use `crontab` to implement scheduled tasks for AMDC service health checks, log rotation, big key scanning, etc.

```
# Hourly health check
0 * * * * /opt/amdc/scripts/amdc_health_check.sh

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

6.3 Monitoring Alert Threshold Recommendations

Indicator	Warning Threshold	Critical Threshold
Memory Usage	70%	85%
Client Connections	80% of maxclients	90%
Replication Lag	5 seconds	30 seconds
Slow Query Count	10 per hour	50 per hour
CPU Usage	70%	90%

6.4 Regular Maintenance Plan

- **Daily:** Health check, log rotation
- **Weekly:** Stress test, big key analysis
- **Monthly:** Configuration review, security audit
- **Quarterly:** Disaster recovery drill, capacity planning

7 Software Management

7.1 Configure System Service for Auto-start on Boot

1. Create service file:

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

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

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

[Service]
Type=simple
User=amdc
Group=amdc
WorkingDirectory=/opt/amdc
ExecStart=/opt/amdc/bin/amdc-server /opt/amdc/amdc.yaml
ExecStop=/opt/amdc/bin/amdc-cli -h 127.0.0.1 -p 6359 -a
your_password shutdown
Restart=on-failure
RestartSec=10
LimitNOFILE=65536

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

3. Enable service:

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

7.2 Build Docker Image

1. Prepare installation package: `AMDC-V2.0.4-xxxx-amd64.tar.gz`
2. Write `Dockerfile` :

```
FROM ubuntu:20.04  
ADD AMDC-V2.0.4-xxxx-amd64.tar.gz /opt/  
WORKDIR /opt/amdc  
EXPOSE 6359  
CMD ["/opt/amdc/bin/amdc-server", "/opt/amdc/amdc.yaml"]
```

3. Build and run:

```
docker build -t apusic/amdc:v2.0.4 .  
docker run -d \  
  --name amdc \  
  -p 6359:6359 \  
  -v ./license.lic:/opt/amdc/license.lic \  
  -v ./amdc.yaml:/opt/amdc/amdc.yaml \  
  -v ./data:/opt/amdc/data \  
  apusic/amdc:v2.0.4
```

Note: In production environments, always mount license, configuration, and persistence directories.

8 Appendix: Key Configuration Optimization Recommendations

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

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网: www.apusic.com

