



APUSIC
固若长城
睿比世界

迁移手册

金蝶Apusic分布式缓存 V2.0.4

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

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 前言
 - 1.1 适用对象
 - 1.2 相关文档
 - 1.3 技术支持
- 2 概述
- 3 AMDC V2.0.2及以下版本迁移至V2.0.4操作说明
 - 3.1 准备工作
 - 3.1.1 环境检查
 - 3.1.2 工具与文件准备
 - 3.1.2.1 必备工具
 - 3.1.2.2 待迁移文件收集
 - 3.2 单机模式迁移
 - 3.2.1 迁移前提
 - 3.2.2 准备工作
 - 3.2.2.1 工具与配置前置
 - 3.2.2.2 备份数据并停止AMDC V2.0.2及以下版本单节点（全量停机）
 - 3.2.3 迁移步骤
 - 3.2.3.1 配置文件迁移
 - 3.2.3.2 缓存数据迁移
 - 3.2.3.3 启动AMDC V2.0.4版本AMDC服务
 - 3.2.3.4 验证启动
 - 3.2.4 注意事项
 - 3.3 主从模式迁移
 - 3.3.1 迁移前提
 - 3.3.2 准备工作
 - 3.3.2.1 工具与配置前置
 - 3.3.2.2 备份数据并停止AMDC V2.0.2及以下的本主从集群（全量停机）
 - 3.3.3 迁移步骤（按「主节点->所有从节点」顺序执行）
 - 3.3.3.1 主节点迁移
 - 3.3.3.2 从节点迁移（所有从节点同步执行）
 - 3.3.4 注意事项
 - 3.4 哨兵模式迁移

- 3.4.1 迁移前提
- 3.4.2 准备工作
 - 3.4.2.1 工具与配置前置
 - 3.4.2.2 备份数据并全量停止AMDC V2.0.2及以下版本的哨兵 + 主从集群
- 3.4.3 迁移步骤（按「AMDC V2.0.4版本主从集群->AMDC V2.0.4版本哨兵集群」顺序执行）
 - 3.4.3.1 重建AMDC V2.0.4版本主从集群（核心，同上一章完整步骤）
 - 3.4.3.2 哨兵节点迁移（所有哨兵节点同步执行）
- 3.4.4 注意事项
- 3.5 集群模式迁移
 - 3.5.1 迁移前提
 - 3.5.2 准备工作
 - 3.5.2.1 工具与配置前置
 - 3.5.2.2 统一环境与记录源拓扑
 - 3.5.2.3 备份数据并全量停止AMDC V2.0.2及以下版本的集群（禁止任何写入）
 - 3.5.2.4 数据文件预处理（所有源节点执行）
 - 3.5.3 迁移步骤（按「配置转换->AMDC V2.0.4版本集群初始化->分片数据导入->集群状态验证」执行）
 - 3.5.3.1 初始化AMDC V2.0.4版本空集群
 - 3.5.3.2 分片数据导入
 - 3.5.3.3 集群状态修复与数据校验
 - 3.5.4 注意事项
- 4 Redis 6.2及以下版本迁移至AMDC V2.0.4操作说明
 - 4.1 准备工作
 - 4.1.1 环境检查
 - 4.1.2 工具与文件准备
 - 4.2 单机模式迁移
 - 4.2.1 迁移前提
 - 4.2.2 准备工作
 - 4.2.2.1 数据与配置备份
 - 4.2.2.2 数据备份并停止Redis单节点
 - 4.2.3 迁移步骤
 - 4.2.3.1 缓存数据迁移
 - 4.2.3.2 修改redis.conf配置

- 4.2.3.3 启动AMDC服务
- 4.2.3.4 验证启动
- 4.2.4 注意事项
- 4.3 主从模式迁移
 - 4.3.1 迁移前提
 - 4.3.2 准备工作
 - 4.3.2.1 工具与文件准备
 - 4.3.2.2 数据备份并停止Redis主从集群（全量停机）
 - 4.3.3 迁移步骤（按「主节点->所有从节点」顺序执行）
 - 4.3.3.1 缓存数据迁移
 - 4.3.3.2 配置文件迁移（redis.conf）
 - 4.3.3.3 启动AMDC主从节点服务
 - 4.3.3.4 主从关系验证
 - 4.3.4 注意事项
- 4.4 哨兵模式迁移
 - 4.4.1 迁移前提
 - 4.4.2 准备工作
 - 4.4.2.1 工具与文件准备
 - 4.4.2.2 数据备份并全量停止Redis哨兵 + 主从集群
 - 4.4.3 迁移步骤
 - 4.4.3.1 缓存数据迁移
 - 4.4.3.2 配置文件迁移（redis.conf）
 - 4.4.3.3 启动AMDC主从节点服务
 - 4.4.3.4 启动AMDC哨兵节点服务
 - 4.4.3.5 哨兵功能验证
 - 4.4.4 注意事项
- 4.5 集群模式迁移
 - 4.5.1 迁移前提
 - 4.5.2 准备工作
 - 4.5.2.1 工具与文件准备
 - 4.5.2.2 数据备份并全量停止Redis集群（禁止任何写入）
 - 4.5.3 迁移步骤
 - 4.5.3.1 数据文件预处理（所有源节点执行）
 - 4.5.3.2 配置文件迁移（redis.conf, node.conf）

- 4.5.3.3 启动AMDC集群节点
- 4.5.3.4 集群状态验证
- 4.5.4 注意事项
- 5 常见问题排查
 - 5.0.1 工具无法运行
 - 5.0.2 校验或修复文件失败
 - 5.0.3 amdc-cli 无法连接AMDC服务
- 6 总结
 - 6.1 AMDC V2.0.2及以下版本向AMDC V2.0.4版本迁移流程
 - 6.2 Redis 6.2及以下版本向AMDC V2.0.4版本迁移流程

1 前言

本文档为金蝶Apusic分布式缓存（AMDC）V2.0.4的迁移手册，详细介绍了如何将Redis迁移到AMDC，以及AMDC老版本迁移到V2.0.4版本。

1.1 适用对象

本文档适用于AMDC产品运维工程师、IT系统运维工程师、开发工程师、软件架构师及研发经理等人员。

1.2 相关文档

了解更多AMDC V2.0.4产品相关的信息，请参阅以下AMDC V2.0.4产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic分布式缓存 V2.0.4 快速使用手册	简单介绍了如何快速上手使用AMDC。
2	金蝶Apusic分布式缓存 V2.0.4 安装手册	详细介绍如何在各操作系统上安装AMDC，以及AMDC服务启停操作，产品的注册过程。
3	金蝶Apusic分布式缓存 V2.0.4 缓存核心用户手册	详细介绍 AMDC 相关功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分布式缓存 V2.0.4 管控台用户手册	详细介绍AMDC管控台相关功能的使用和操作说明。
5	金蝶Apusic分布式缓存 V2.0.4 开发手册	详细介绍基于各开发语言进行AMDC客户端应用开发的说明。
6	金蝶Apusic分布式缓存 V2.0.4 迁移手册	详细介绍AMDC历史版本迁移升级到V2.0.4版本的说明，以及Redis迁移到AMDC的说明。
7	金蝶Apusic分布式缓存 V2.0.4 运维手册	详细介绍AMDC的监控、运维、安全加固等运维说明。
8	金蝶Apusic分布式缓存 V2.0.4 性能优化手册	详细介绍AMDC性能调优的说明。

1.3 技术支持

AMDC产品提供全面的技术支持服务，您可以通过以下方式获得技术支持：

- 网址: www.apusic.com
- 电话: 400-855-5800
- 邮箱: support@apusic.com
- 金蝶云社区: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技术支持时, 请提供如下信息:

1. 您的姓名
2. 公司与联系方式
3. 操作系统及其版本
4. 产品版本号
5. 出现异常及错误的日志、截图等详细信息

2 概述

本手册介绍两种迁移至AMDC V2.0.4的操作说明：

1. **AMDC V2.0.2及以下版本迁移至AMDC V2.0.4**：使用amdc-conf-conv转换AMDC V2.0.2及以下版本的配置文件为AMDC V2.0.4版本兼容格式，复用原有配置及数据文件，启动后验证功能即可；
2. **Redis 6.2及以下版本迁移至AMDC V2.0.4**：核心依托AMDC可直接加载Redis配置（redis.conf等）及数据文件（RDB/AOF等）的特性。

3 AMDC V2.0.2及以下版本迁移至V2.0.4操作说明

3.1 准备工作

3.1.1 环境检查

- 已安装AMDC V2.0.4版本（需与工具版本匹配），且安装目录权限正常（可读可执行）；
- 待迁移的AMDC V2.0.2及以下版本的服务可正常访问，或已获取其RDB缓存文件、配置文件（yaml格式）；
- 迁移目标服务器（部署的AMDC V2.0.4服务器）具备足够的磁盘空间，确保能存储迁移后的缓存数据及配置文件；

3.1.2 工具与文件准备

3.1.2.1 必备工具

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的RDB缓存文件（随AMDC V2.0.4安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AOF文件（同目录部署）；
- [amdc-conf-conv](#)：用于转换AMDC V2.0.2及以下版本的配置文件为AMDC V2.0.4版本兼容格式（同目录部署）；

3.1.2.2 待迁移文件收集

- AMDC V2.0.2及以下版本的配置文件：服务端配置（如conf.yaml）或哨兵配置（如sentinel_go.yaml）；
- AMDC V2.0.2及以下版本的RDB快照文件：默认名为dump.rdb，路径通常在安装目录下（需确认实际路径）；
- 若AMDC V2.0.2及以下版本启用了AOF持久化，则优先使用AOF文件作为迁移数据源，RDB文件作为备份迁移数据源；

3.2 单机模式迁移

单机模式迁移适用于仅部署单台AMDC服务的场景，核心流程为「文件准备 -> 配置转换 -> 数据校验修复 -> 部署启动 -> 验证」，操作简单且无依赖关系，具体步骤如下：

3.2.1 迁移前提

- 目标服务器已安装AMDC V2.0.4，安装目录权限正常（可读可执行）；

- 已收集AMDC V2.0.2及以下版本的配置文件（如conf.yaml）、RDB快照文件(dump.rdb)，若启用AOF持久化则优先获取AOF文件；
- 目标服务器磁盘空间充足，至少为源数据文件大小的2倍（预留备份与运行空间）。

3.2.2 准备工作

3.2.2.1 工具与配置前置

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的RDB缓存文件（随AMDC V2.0.4安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AOF文件（同目录部署）；
- [amdc-conf-conv](#)：用于转换AMDC V2.0.2及以下版本的配置文件为AMDC V2.0.4版本兼容格式（同目录部署）；

3.2.2.2 备份数据并停止AMDC V2.0.2及以下版本单节点（全量停机）

- AMDC V2.0.2及以下版本单节点的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端AMDC配置；
- 必须先停止AMDC V2.0.2及以下版本的单节点，禁止任何数据写入，确保源数据文件最终一致性。

#节点执行：停止AMDC V2.0.2及以下版本的AMDC进程

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

#验证进程已停止（无输出则表示成功）

```
ps -ef | grep amdc-server | grep -v grep
```

3.2.3 迁移步骤

3.2.3.1 配置文件迁移

使用 [amdc-conf-conv](#) 工具将AMDC V2.0.2及以下版本的服务端配置转换为AMDC V2.0.4版本兼容格式，操作如下：

1. 备份目标服务器的AMDC V2.0.4版本默认配置文件（避免覆盖丢失）：

```
cp /amdc/amdc.yaml /amdc/amdc.yaml.bak
```

2. 执行配置转换（指定--server参数，适配服务端配置）：

```
amdc-conf-conv --server --go-yaml /path/to/conf.yaml --c-yaml
/amdc/amdc.yaml
```

3. 验证转换结果：打开/amdc/amdc.yaml，确认核心配置项（如端口、密码、数据目录、内存限制等）已正确继承配置，无语法错误；

3.2.3.2 缓存数据迁移

1. 校验并修复源RDB或AOF文件（优先处理AOF文件，数据更完整）：

#若使用AOF文件

```
amdc-check-aof /path/to/appendonlydir/appendonly.aof.manifest
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件损坏，执行修复

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.base.rdb /
```

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.base.aof /
```

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.incr.aof
```

```
amdc-check-rdb --fix /path/to/dump.rdb
```

2. 复制修复后的文件到AMDC V2.0.4版本的安装目录：

#查看AMDC V2.0.4版本数据目录（在amdc.yaml中通过dir配置项确认，默认路径如下），复制RDB文件

```
cp /path/to/dump.rdb/ amdc/
```

#若使用AOF文件，需先启用AOF配置（在amdc.yaml中设置appendonly yes），再复制

```
cp /path/to/appendonlydir /amdc/
```

3.2.3.3 启动AMDC V2.0.4版本AMDC服务

1. 启动AMDC V2.0.4版本AMDC服务，加载转换后的配置文件：

```
amdc-server /amdc/amdc.yaml
```

2. 验证服务启动状态：

```
#查看服务进程
ps -ef | grep amdc-server

#查看日志（默认日志路径 / amdc/server.log）
tail -100f /amdc/server.log
```

3.2.3.4 验证启动

1. 连接AMDC V2.0.4版本AMDC服务，验证数据完整性：

```
#使用amdc-cli连接（默认端口6359，若配置密码需加-a选项）
amdc-cli -p 6359 -a test123

#查看键空间统计信息
info keyspace
```

2. 验证功能可用性：执行写入、修改、删除等操作，确认服务正常响应，无配置或数据相关报错：

3.2.4 注意事项

- 迁移过程中需停止AMDC V2.0.2及以下版本的服务，避免数据写入导致源文件损坏；
- 迁移后观察服务运行 10-20 分钟，确认无日志报错等异常。

3.3 主从模式迁移

主从模式迁移适配下，AMDC V2.0.2及以下版本与AMDC V2.0.4版本无法跨版本建立主从的场景，核心原则为全量停机 -> 先迁主节点 -> 后迁从节点 -> 数据一致验证，通过RDB或AOF全量数据导入完成迁移，全程需短暂停机（无跨版本同步环节），具体步骤如下：

3.3.1 迁移前提

- 主、从目标服务器均已安装AMDC V2.0.4版本，安装目录权限正常（可读、可写、可执行），且主从节点网络互通（无防火墙拦截6359端口）；
- 已收集AMDC V2.0.2及以下版本主、从节点的配置文件（conf.yaml）、AOF文件（优先，数据更完整）或RDB快照文件，并做好全量备份；
- 提前告知业务方停机窗口，建议选择业务低峰期操作，停机时长约为「数据文件拷贝 + 服务启动」耗时（根据数据量调整，一般5-30分钟）；
- 目标服务器磁盘空间充足，至少为源数据文件大小的2倍（预留备份与运行空间）。

3.3.2 准备工作

3.3.2.1 工具与配置前置

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的RDB缓存文件（随AMDC V2.0.4安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AOF文件（同目录部署）；
- [amdc-conf-conv](#)：用于转换AMDC V2.0.2及以下版本的配置文件为AMDC V2.0.4版本兼容格式（同目录部署）；

3.3.2.2 备份数据并停止AMDC V2.0.2及以下的本主从集群（全量停机）

- AMDC V2.0.2及以下版本单节点的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端AMDC配置；
- 必须先停止所有AMDC V2.0.2及以下版本的主、从节点，禁止任何数据写入，确保源数据文件最终一致性。

```
#主节点 + 所有从节点统一执行：停止AMDC V2.0.2及以下版本的AMDC进程
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs
kill -9

#验证进程已停止（无输出则表示成功）
ps -ef | grep amdc-server | grep -v grep
```

3.3.3 迁移步骤（按「主节点->所有从节点」顺序执行）

3.3.3.1 主节点迁移

主节点为数据核心，需优先完成配置转换、数据校验导入和服务启动，作为后续从节点的同步源。

步骤1: 配置文件转换与优化

1. 备份AMDC V2.0.4版本主节点默认配置（避免覆盖）：

```
cp /amdc/amdc.yaml /amdc/amdc-master.yaml.bak
```

2. 将AMDC V2.0.2及以下版本的主节点配置转换为AMDC V2.0.4版本主节点专用配置（命名为amdc-master.yaml，便于区分）：

```
amdc-conf-conv --server --go-yaml /path/to/conf.yaml --c-yaml  
/amdc/amdc-master.yaml
```

3. 检查AMDC V2.0.4版本主节点配置，启用主节点身份（无replicaof配置），确认核心配置项：

```
vi /amdc/amdc-master.yaml
```

关键配置确认 / 修改：

- port:6359（端口统一，若需修改则所有节点同步）；
- requirepass: 你的密码（若源主节点有密码，需保留，后续从节点需配置masterauth）；
- dir:/amdc/（数据目录，与实际规划一致）；
- appendonly:yes（建议启用AOF，与源端保持一致，若源端仅用RDB则设为no）；
- logfile:amdc-master.log（主节点独立日志，便于排查）。

步骤2: 数据文件校验、修复与导入

1. 校验并修复源AMDC V2.0.2及以下版本的主节点数据文件（主节点执行，若文件无损坏则跳过修复）：

#若使用AOF文件（推荐）

```
amdc-check-aof /path/to/appendonlydir/appendonly.aof.manifest
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件损坏，执行修复

```

amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.rdb /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.aof /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.incr.aof

amdc-check-rdb --fix /path/to/dump.rdb

```

2. 将修复后的干净数据文件拷贝到AMDC V2.0.4版本主节点数据目录，并修改权限：

```

#拷贝AOF文件（推荐，若用RDB则执行下一行）
cp /path/to/appendonlydir/ /amdc/

#拷贝RDB文件
cp /path/to/dump.rdb /amdc/

#赋权（避免服务无权限读取）
chmod 644 /amdc/{appendonlydir/, dump.rdb}

```

步骤3：启动AMDC V2.0.4版本主节点服务

```

#启动主节点，指定专用配置文件
amdc-server /amdc/amdc-master.yaml

#验证主节点启动状态
ps -ef | grep amdc-server | grep master # 查看进程
tail -100f /amdc/amdc-master.log # 查看日志

```

日志显示 Ready to accept connections 表示主节点启动成功；

连接主节点验证数据完整性（核心，确保数据导入成功）：

```

amdc-cli -p 6359 -a 密码 # 连接主节点

```

```
info keypace # 查看键空间统计 (确认数据量)
```

3.3.3.2 从节点迁移 (所有从节点同步执行)

从节点无需单独导入数据，启动后通过AMDC V2.0.4版本内部主从同步，从已启动的AMDC V2.0.4版本主节点拉取全量数据，确保与主节点数据完全一致。

步骤1: 配置文件转换与主从关联

1. 备份AMDC V2.0.4版本从节点默认配置：

```
cp /amdc/amdc.yaml /amdc/amdc-slave.yaml.bak
```

2. 将AMDC V2.0.2以下版本从节点配置转换为AMDC V2.0.4版本从节点专用配置 (命名为 amdc-slave.yaml)：

```
amdc-conf-conv --server --go-yaml /path/to/conf.yaml --c-yaml  
/amdc/amdc-slave.yaml
```

3. 检查AMDC V2.0.4版本从节点配置，指定主节点信息，启用从节点身份：

```
vi /amdc/amdc-slave.yaml
```

关键配置确认 / 修改：

- port:6360 (端口统一，若需修改则所有节点同步)；
- requirepass: 你的密码 (若源主节点有密码，需保留，后续从节点需配置 masterauth)；
- dir:/amdc/ (数据目录，与实际规划一致)；
- appendonly:yes (建议启用AOF，与源端保持一致，若源端仅用RDB则设为no)；
- replicaof: 主节点IP主节点Port# 指向AMDC V2.0.4版本主节点的IP和端口；
- logfile:amdc-slave.log (主节点独立日志，便于排查)。

步骤2: 启动AMDC V2.0.4版本从节点服务

无需向从节点数据目录拷贝任何源数据文件，AMDC V2.0.4版本从节点会自动从主节点同步：

```
#启动从节点，指定专用配置文件
amdc-server /amdc/amdc-slave.yaml

#验证从节点启动状态
ps -ef | grep amdc-server # 查看进程
tail -100f /amdc/amdc-slave.log # 查看日志
```

日志显示 MASTER <-> REPLICA sync: Finished with success 表示从节点已成功连接主节点并开始同步。

步骤3：验证从节点同步状态

连接从节点，确认同步完成且数据与主节点一致：

```
amdc-cli -p 6360 -a 密码 # 连接从节点

info replication # 查看同步状态
```

关键状态确认：

- role:slave：从节点身份正常；
- master_host：主节点IP、masterport:6359：主节点指向正确；
- master_link_status: up：主从连接正常。

3.3.4 注意事项

- 迁移全程必须先全量停止AMDC V2.0.2及以下版本的主从节点，禁止数据写入，否则会导致源数据文件不一致，导入后数据丢失；
- 主节点数据导入后必须先验证数据完整性，再启动从节点，避免从节点同步错误数据；
- 从节点无需拷贝任何源数据文件，由AMDC V2.0.4版本内部主从同步完成，手动拷贝会导致同步冲突；
- 主、从节点的核心配置必须一致（AOF开关、端口、密码等），否则会导致连接失败或同步异常；
- 迁移后观察集群运行10-20分钟，确认主从同步无延迟、日志无报错、业务访问正常。

3.4 哨兵模式迁移

AMDC V2.0.2及以下版本与AMDC V2.0.4版本无法跨版本主从同步、哨兵无法监控跨版本节点的场景，核心原则为全量停机->先迁主从节点（同第六章）->后迁哨兵节点->重新关联监控，通过「AMDC V2.0.4版本主从集群重建 + 哨兵重新监控」完成迁移，全程需短暂业务停机，具体步骤如下：

3.4.1 迁移前提

- 所有主从节点、哨兵节点均已安装AMDC V2.0.4版本，网络互通（开放6359服务端口、26369哨兵端口）；
- 已收集AMDC V2.0.2及以下版本的主从节点配置、数据文件、哨兵配置文件(sentinel_go.yaml)，并做好全量备份；
- 提前规划停机窗口（低峰期），停机时长为「AMDC V2.0.2及以下版本停机 + AMDC V2.0.4版本主从启动 + 哨兵启动」耗时（一般10-40分钟）；
- 所有目标节点时间同步，磁盘空间充足（主节点数据目录预留2倍源数据空间）。

3.4.2 准备工作

3.4.2.1 工具与配置前置

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的RDB缓存文件（随AMDC V2.0.4安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AOF文件（同目录部署）；
- [amdc-conf-conv](#)：用于转换AMDC V2.0.2及以下版本的配置文件为AMDC V2.0.4版本兼容格式（同目录部署）；

3.4.2.2 备份数据并全量停止AMDC V2.0.2及以下版本的哨兵 + 主从集群

- AMDC V2.0.2及以下版本单节点的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端AMDC配置；
- 必须先停止所有AMDC V2.0.2及以下版本的哨兵节点和主从节点，禁止数据写入和哨兵监控，确保源数据一致性。

```
#所有节点统一执行：停止AMDC V2.0.2及以下版本的AMDC主从 + 哨兵进程
```

```
ps -ef | grep -E 'amdc-server | amdc-sentinel' | grep -v grep | awk
'{print $2}' | xargs kill -9
```

```
#验证进程已停止（无输出则成功）
```

```
ps -ef | grep -E 'amdc-server | amdc-sentinel' | grep -v grep
```

3.4.3 迁移步骤（按「AMDC V2.0.4版本主从集群->AMDC V2.0.4版本哨兵集群」顺序执行）

3.4.3.1 重建AMDC V2.0.4版本主从集群（核心，同上一章完整步骤）

- 严格按照上一章主从模式迁移的所有步骤，完成AMDC V2.0.4版本主从集群的搭建：
- 迁移AMDC V2.0.4版本主节点：配置转换 -> 数据校验导入 -> 启动 -> 验证数据完整性；
- 迁移所有AMDC V2.0.4版本从节点：配置转换（关联主节点） -> 启动 -> 验证主从同步；
- 完成主从集群整体验证：数据一致性、主从功能、日志无异常。

此步骤为哨兵迁移的基础，必须确保AMDC V2.0.4版本主从集群完全正常运行，再进行后续哨兵迁移。

3.4.3.2 哨兵节点迁移（所有哨兵节点同步执行）

哨兵节点无需导入数据，仅需将AMDC V2.0.2及以下版本的哨兵配置转换为AMDC V2.0.4版本兼容格式，并修改监控目标为已搭建完成的AMDC V2.0.4版本主节点。

步骤1：配置文件转换与监控目标修改

1. 备份AMDC V2.0.4版本哨兵默认配置：

```
cp /amdc/sentinel.yaml /amdc/sentinel-c.yaml.bak
```

2. 将AMDC V2.0.2及以下版本的哨兵配置转换为AMDC V2.0.4版本哨兵专用配置（命名为sentinel-c.yaml）：

```
amdc-conf-conv --sentinel --go-yaml /path/to/sentinel_go.yaml --c-  
yaml /amdc/sentinel.yaml
```

3. 检查AMDC V2.0.4版本哨兵配置，确认哨兵关键配置：

```
vi /amdc/sentinel.yaml
```

关键配置修改或确认（核心为monitor配置项，指向AMDC V2.0.4版本主节点）：

```
#哨兵核心监控配置（替换为AMDC V2.0.4版本主节点信息，其余参数与源端保持一致）  
  
SENTINEL:  
  
sentinel monitor: mymaster 192.168.1.100 6359 2 # mymaster: 监控名称
```

(与源端一致), 后接C主节点IP、端口、quorum阈值

```
sentinel down-after-milliseconds: mymaster 30000 #主观下线超时时间 (与源端一致)
```

```
sentinel failover-timeout: mymaster 180000 #故障转移超时时间 (与源端一致)
```

```
sentinel auth-pass: mymaster 密码 #与C主节点的requirepass一致, 无密码则省略
```

其他配置确认:

- port:26359 (哨兵默认端口, 与源端一致);
- logfile:sentinel.log (哨兵独立日志);
- dir:/amdc/ (哨兵数据目录, 存储监控状态)。

#所有哨兵节点同步启动, 指定专用配置文件

```
amdc-sentinel /amdc/sentinel.yaml
```

#验证哨兵启动状态

```
ps -ef | grep amdc-sentinel #查看进程
```

```
tail -100f /amdc/sentinel.log #查看日志
```

步骤2: 启动AMDC V2.0.4版本哨兵服务

日志显示 Sentinel ID is xxxxxxxxxxxxxxxxxxxxxxxx 表示哨兵启动成功。

关键验证: 哨兵已成功发现AMDC V2.0.4版本主从集群的所有节点

#连接哨兵端口 (26369), 查询监控状态

```
amdc-cli -p 26359
```

```
info sentinel #查看哨兵监控信息
```

```
sentinel slaves mymaster # 查看监控的从节点列表
```

验证结果要求：

- 从节点列表中显示的IP和端口均为AMDC V2.0.4版本从节点信息，状态为online。

3.4.4 注意事项

- 迁移前必须全量停止AMDC V2.0.2及以下版本的哨兵 + 主从节点，避免哨兵持续监控已停机节点，或数据写入导致源文件不一致；
- 必须先完成AMDC V2.0.4版本主从集群的搭建和验证，再启动哨兵节点，否则哨兵会监控不到有效节点，导致迁移失败；
- 哨兵配置中监控名称（如mymaster）、quorum阈值、超时时间建议与源端保持一致，减少客户端配置修改；
- 所有哨兵节点的监控配置必须完全一致（监控目标、密码、阈值等），否则会导致哨兵集群分裂，故障转移异常；
- 迁移后观察10-20分钟，重点监控：哨兵日志无报错、主从切换无异常、同步延迟在可接受范围、业务访问无中断。

3.5 集群模式迁移

集群模式迁移适配下，AMDC V2.0.2及以下版本与AMDC V2.0.4版本无法跨版本建立集群、无法跨版本同步分片数据的场景，核心原则为全量停机->分片数据全量导出->AMDC V2.0.4版本集群初始化->分片数据批量导入->集群验证，因集群为多主多从分片架构，无跨版本同步可能，需通过「全量数据导出-导入」完成整集群迁移，全程需业务停机，具体步骤如下：

3.5.1 迁移前提

- 目标集群所有节点（与源集群节点数一致：M主N从）均已安装AMDC V2.0.4版本，网络互通（开放6359服务端口、16359集群gossip通信端口）；
- 已收集AMDC V2.0.2及以下版本的集群所有节点的配置文件、AOF、RDB数据文件，并记录源集群拓扑信息（执行cluster nodes 获取槽位分配、主从关系）；
- 提前规划长停机窗口（根据数据量调整，建议低峰期），停机时长为「数据导出 + 集群初始化 + 数据导入 + 验证」耗时（数据量10G内约30-60分钟）；
- 所有目标节点磁盘空间充足（每节点预留2倍源节点数据空间），时间同步，防火墙放行6359、16359端口。

3.5.2 准备工作

3.5.2.1 工具与配置前置

- [amdc-check-rdb](#): 用于校验和修复AMDC V2.0.2及以下版本的RDB缓存文件（随AMDC V2.0.4安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#): 用于校验和修复AMDC V2.0.2及以下版本的AOF文件（同目录部署）；
- [amdc-conf-conv](#): 用于转换AMDC V2.0.2及以下版本的配置文件为AMDC V2.0.4版本兼容格式（同目录部署）；

3.5.2.2 统一环境与记录源拓扑

1. 所有目标节点创建统一目录并赋权：
2. 观察AMDC V2.0.2及以下版本的集群核心拓扑：

连接AMDC V2.0.2及以下版本的集群任意节点，执行以下命令，保存输出结果（后续AMDC V2.0.4版本集群初始化完全复用）：

```
amdc-cli cluster nodes #记录：主从节点IP和端口、节点ID、槽位分配（如0-5460）
```

或者查看node.conf文件，其中记录有集群节点IP，端口，节点ID，槽位分配等信息。

```
29755a26cc60c056b3af39eb66dc5ccbfbdd20c1 127.0.0.1:6383@16383 slave
882b57434ef8c361fcea44c327379880181af1a6 0 1769505692837 2 connected

882b57434ef8c361fcea44c327379880181af1a6 127.0.0.1:6380@16380 master
- 0 1769505692432 2 connected 5461-10922

980c1df981902c2df617ca81d4df7219b1e86a8b 127.0.0.1:6384@16384
myself,slave 2793d6ad527eee582aa30890d7c8407e9c112b4f003 connected

43d25d48fac16f9b8f6b519f88a889179b8a6e87 127.0.0.1:6382@16382 slave
ac9f0c10e742b4570f6436db8af281c1d51ea570 0 1769505692331 1 connected

2793d6ad527eee582aa30890d7c8407e9c112b4f 127.0.0.1:6381@16381 master
- 0 1769505692332 3 connected 10923-16383
```

```
ac9f0c10e742b4570f6436db8af281c1d51ea570 127.0.0.1:6379@16379 master
- 0 1769505692228 1 connected 0-5460
```

记录每个AMDC V2.0.2及以下版本的每个主节点IP与端口号，备份对应机器上的AOF或RDB文件用于后续将数据迁移到AMDC V2.0.4版本集群：

3.5.2.3 备份数据并全量停止AMDC V2.0.2及以下版本的集群（禁止任何写入）

- AMDC V2.0.2及以下版本单节点的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端AMDC配置；

```
#所有AMDC V2.0.2及以下版本的集群节点统一执行：停止进程
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs
kill -9

#验证进程已停止（无输出则成功）
ps -ef | grep amdc-server | grep -v grep
```

3.5.2.4 数据文件预处理（所有源节点执行）

对每个AMDC V2.0.2及以下版本的主节点的AOF或RDB文件进行校验 + 修复，确保为干净数据文件，避免导入AMDC V2.0.4版本后出现数据损坏：

```
#优先使用AOF文件（每个节点单独处理，集群为分片存储，各节点数据独立）
amdc-check-aof /path/to/nodeX/appendonlydir/appendonly.aof.manifest

#若无AOF则使用RDB文件
amdc-check-rdb /path/to/nodeX/dump.rdb

#若文件损坏，执行修复
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.rdb /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.aof /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.incr.aof
```

```
amdc-check-rdb --fix /path/to/nodeX/dump.rdb
```

3.5.3 迁移步骤（按「配置转换->AMDC V2.0.4版本集群初始化->分片数据导入->集群状态验证」执行）

3.5.3.1 初始化AMDC V2.0.4版本空集群

步骤1: 所有AMDC V2.0.4版本节点配置转换与集群启用

1. 配置转换:

```
amdc-conf-conv --server --go-yaml /path/to/nodeX/conf.yaml --c-yaml
/amdc/amdc.yaml
```

2. 检查配置文件:

```
vi /amdc/amdc.yaml
```

关键配置:

- port:6359（哨兵默认端口，与源端一致）；
- logfile: amdc-cluster-XXX.log（节点独立日志）；
- dir:/amdc/（统一数据目录）；
- requirepass: 密码（与源端一致，所有节点相同）；
- cluster-enabled:yes（启用集群模式）；
- cluster-config-file:nodes-XXX.conf（每个节点单独的集群配置文件）；
- cluster-node-timeout:15000（集群节点超时时间，与源端一致）；
- cluster-require-full-coverage:yes（开启全槽位覆盖检查）。

3. 所有节点配置完成后，启动所有AMDC V2.0.4版本节点（仅启动服务，未形成集群）：

```
amdc-server /amdc/amdc-cluster-XXX.yaml
```

验证所有节点启动成功（日志显示Ready to accept connections）

步骤2: 执行集群初始化 (槽位 + 主从分配)

选择任意一个AMDC V2.0.4版本主节点, 执行amdc-cli --cluster create命令, 严格按照源拓扑指定节点IP、端口和槽位分配, 示例以3主3从 (源拓扑槽位 0-5460、5461-10922、10923-16383) 为例:

```
#命令格式: amdc-cli --cluster create 主1 主2 主3 从1 从2 从3 --cluster-replicas 1
#--cluster-replicas 1: 表示1主1从, 与源端一致

amdc-cli --cluster create 192.168.1.10:6359 192.168.1.11:6359
192.168.1.12:6359 192.168.1.13:6359 192.168.1.14:6359
192.168.1.15:6359 --cluster-replicas 1
```

执行成功后, 验证集群初始化状态:

```
amdc-cli -p 6359 -a 密码 cluster nodes
```

要求: 全部槽位均已合理分配至集群各节点, 所有节点状态为 myself, master 或 slave, 无 fail 状态。

3.5.3.2 分片数据导入

先启动AMDC V2.0.4版本单机节点用于加载RDB或AOF文件, 再使用AMDC V2.0.4版本 amdc-cli --cluster import 命令, 将单机节点中的数据自动重新映射导入到集群中, 命令为集群标准化导入方式, 无需手动操作数据文件, 避免文件权限、格式不兼容等问题。

```
#核心命令格式 (一行执行, 换行符 \ 仅为排版)

amdc-cli --cluster import 集群主节点IP: 集群主节点Port \

--cluster-from 单节点IP: 单节点Port \

--cluster-copy \

--cluster-replace \

-a 集群密码
```

参数详解:

- --cluster-copy: 必加参数, 复制数据时保留源单节点的原始数据, 避免源数据被删除;
- --cluster-replace: 可选参数, 若集群中存在与源节点同名的键, 直接覆盖 (建议迁移时添加, 确保数据与源端一致, 根据业务需求选择)。

因集群为分片存储, 每个AMDC V2.0.2及以下版本的源分片仅存储对应槽位的数据, 需按「源分片主节点->AMDC V2.0.4版本对应分片主节点」的顺序, 逐个分片执行导入命令, 全程在AMDC V2.0.4版本集群任意可操作节点执行即可, 无需在目标主节点本地操作。

示例: 以3主3从集群为例 (源主1 -> C主1、源主2 -> C主2、源主3 -> C主3)

```
#导入分片1: 源主1 (192.168.1.1:6359) -> AMDC V2.0.4版本主
1 (192.168.1.10:6359)

amdc-cli--cluster import 192.168.1.10:6359 \

--cluster-from 192.168.1.1:6359 \

--cluster-copy \

--cluster-replace \

-a 集群密码

#导入分片2: 源主2 (192.168.1.2:6359) -> AMDC V2.0.4版本主
2 (192.168.1.11:6359)

amdc-cli--cluster import 192.168.1.11:6359 \

--cluster-from 192.168.1.2:6359 \

--cluster-copy \

--cluster-replace \

-a 集群密码
```

```
#导入分片3: 源主3(192.168.1.3:6359) -> AMDC V2.0.4版本主
3(192.168.1.12:6359)

amdc-cli --cluster import 192.168.1.13:6359 \

--cluster-from 192.168.1.3:6359 \

--cluster-copy \

--cluster-replace \

-a 集群密码
```

3.5.3.3 集群状态修复与数据校验

因手动导入数据，需执行集群状态修复，确保槽位与数据匹配，无槽位丢失或数据错乱：

```
#验证槽位覆盖 (必须显示16384个槽位全部分配)
amdc-cli -p 6359 -a 密码 cluster info
cluster info 输出中 cluster_slots_assigned:16384 表示槽位全部分配，无遗
漏。
```

3.5.4 注意事项

- 集群迁移必须全量停机，且源集群所有节点禁止写入，因集群为分片存储，单个节点数据写入会导致分片数据不一致；
- 集群 gossip通信端口 16359(6359+10000) 必须放行，否则节点无法发现集群拓扑，导致集群分裂；
- 执行amdc-cli --cluster import命令时，--cluster-copy为必加参数，省略该参数会直接删除AMDC V2.0.2及以下版本的源节点的原始数据，造成不可逆的源数据丢失；--cluster-replace 建议迁移时添加，确保覆盖AMDC V2.0.4版本集群内的同名测试键，与源端数据完全一致；
- 若源集群存在数据倾斜（某分片数据量过大），导入前需提前评估AMDC V2.0.4版本集群对应节点的磁盘、内存资源；迁移后及时对倾斜分片进行扩容，避免集群出现性能瓶颈、影响业务访问速度；
- 迁移完成后建议观察集群运行10-20分钟，重点监控：各分片数据无丢失、主节点故障转移自动触发、跨分片操作正常执行、主从同步无持续延迟、业务访问延迟符合预期。

4 Redis 6.2及以下版本迁移至AMDC V2.0.4操作说明

4.1 准备工作

4.1.1 环境检查

- 目标服务器已安装AMDC V2.0.4版本，安装目录权限正常（可读可执行）；
- 已收集Redis的RDB快照文件(dump.rdb)，若启用AOF持久化则优先获取AOF文件；
- 目标服务器磁盘空间充足，至少为源数据文件大小的2倍（预留备份与运行空间）。

4.1.2 工具与文件准备

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的RDB缓存文件（随AMDC V2.0.4版本AMDC安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的AOF文件（同目录部署）；

4.2 单机模式迁移

单机模式迁移适用于仅部署单台AMDC服务的场景，核心流程为「文件准备 -> 数据校验修复 -> 部署启动 -> 验证」，具体步骤如下：

4.2.1 迁移前提

- 目标服务器已安装AMDC V2.0.4版本，安装目录权限正常（可读可执行）；
- 已收集Redis的RDB快照文件(dump.rdb)，若启用AOF持久化则优先获取AOF文件；
- 目标服务器磁盘空间充足，至少为源数据文件大小的2倍（预留备份与运行空间）。

4.2.2 准备工作

4.2.2.1 数据与配置备份

- 检查AMDC V2.0.4安装目录、数据存储目录权限（确保AMDC用户可读写）；
- RDB文件和Redis配置文件复制至目标端AMDC指定备份目录（如/amdc 644），确保AMDC可读取。

4.2.2.2 数据备份并停止Redis单节点

- 停止业务对Redis的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端Redis配置(redis.conf)；

- 必须先停止Redis单节点，禁止任何数据写入，确保源数据文件最终一致性。

```
#节点执行：停止Redis进程
```

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

```
#验证进程已停止（无输出则表示成功）
```

```
ps -ef | grep redis-server | grep -v grep
```

4.2.3 迁移步骤

4.2.3.1 缓存数据迁移

1. 校验并修复源RDB或AOF文件（优先处理AOF文件，数据更完整）：

```
#若使用AOF文件
```

```
amdc-check-aof /path/to/appendonly.aof
```

```
#若使用RDB文件
```

```
amdc-check-rdb /path/to/dump.rdb
```

```
#若文件损坏，执行修复
```

```
amdc-check-aof --fix /path/to/appendonly.aof
```

```
amdc-check-rdb --fix /path/to/dump.rdb
```

2. 复制修复后的文件到AMDC V2.0.4版本的安装目录：

```
#查看AMDC V2.0.4版本数据目录（在amdc.yaml中通过dir配置项确认，默认路径如下），复制RDB文件
```

```
cp /path/to/dump.rdb/ amdc/
```

```
#若使用AOF文件，需先启用AOF配置（在amdc.yaml中设置appendonly yes），再复制
```

```
cp /path/to/appendonly.aof /amdc/
```

4.2.3.2 修改redis.conf配置

1. 修改Redis服务配置文件(redis.conf), 追加以下配置项到配置文件最下方:

```
license /path/to/license.xml
apusic-acls /path/to/acls.properties
worker-threads 1
```

4.2.3.3 启动AMDC服务

1. 启动AMDC V2.0.4版本AMDC服务, 直接加载Redis的配置文件:

```
amdc-server /amdc/redis.conf
```

2. 验证服务启动状态:

```
#查看服务进程
ps -ef | grep redis-server

#查看日志 (默认日志路径 / amdc/server.log)
tail -100f /amdc/server.log
```

4.2.3.4 验证启动

1. 连接AMDC V2.0.4版本AMDC服务, 验证数据完整性:

```
#使用amdc-cli连接 (Redis默认端口6379, 若配置密码需加-a选项)
amdc-cli -p 6379 -a test123

#查看键空间统计信息
info keyspace
```

2. 验证功能可用性: 执行写入、修改、删除等操作, 确认服务正常响应, 无配置或数据相关报错:

4.2.4 注意事项

- 迁移过程中需停止Redis服务, 避免数据写入导致源文件损坏;
- 迁移后观察服务运行 10-20 分钟, 确认无日志报错等异常。

4.3 主从模式迁移

主从模式迁移适用于部署一主多从AMDC服务的场景，核心流程为「文件准备 -> 数据校验修复 -> 部署启动 -> 验证」，具体步骤如下：

4.3.1 迁移前提

- 主、从目标服务器均已安装AMDC V2.0.4版本，安装目录权限正常（可读、可写、可执行），且主从节点网络互通（无防火墙拦截6359端口）；
- 已收集Redis主、从节点的配置文件（redis.conf）、AOF文件（优先，数据更完整）或RDB快照文件，并做好全量备份；
- 提前告知业务方停机窗口，建议选择业务低峰期操作，停机时长约为「数据文件拷贝 + 服务启动」耗时（根据数据量调整，一般5-30分钟）；
- 目标服务器磁盘空间充足，至少为源数据文件大小的2倍（预留备份与运行空间）。

4.3.2 准备工作

4.3.2.1 工具与文件准备

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的RDB缓存文件（随AMDC V2.0.4版本AMDC安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的AOF文件（同目录部署）；

4.3.2.2 数据备份并停止Redis主从集群（全量停机）

- 停止业务对Redis的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端Redis配置（redis.conf）；
- 必须先停止所有Redis主、从节点，禁止任何数据写入，确保源数据文件最终一致性。

```
#主节点 + 所有从节点统一执行：停止Redis进程
```

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

```
#验证进程已停止（无输出则表示成功）
```

```
ps -ef | grep redis-server | grep -v grep
```

4.3.3 迁移步骤（按「主节点->所有从节点」顺序执行）

4.3.3.1 缓存数据迁移

1. 校验并修复源Redis主节点数据文件（主节点执行，若文件无损坏则跳过修复）：

```
#若使用AOF文件（推荐）
amdc-check-aof /path/to/appendonly.aof

#若使用RDB文件
amdc-check-rdb /path/to/dump.rdb

#若文件损坏，执行修复
amdc-check-aof --fix /path/to/appendonly.aof

amdc-check-rdb --fix /path/to/dump.rdb
```

2. 将修复后的干净数据文件拷贝到AMDC V2.0.4版本主节点数据目录，并修改权限：

```
#拷贝AOF文件（推荐，若用RDB则执行下一行）
cp /path/to/appendonly.aof /amdc/

#拷贝RDB文件
cp /path/to/dump.rdb /amdc/

#赋权（避免服务无权限读取）
chmod 644 /amdc/{appendonly.aof, dump.rdb}
```

4.3.3.2 配置文件迁移（redis.conf）

1. 将Redis主/从节点的redis.conf拷贝至AMDC主/从节点的根目录。
2. 修改Redis服务配置文件(redis.conf)，追加以下配置项到配置文件最下方：

```
license /path/to/license.xml
apusic-acls /path/to/acls.properties
worker-threads 1
```

4.3.3.3 启动AMDC主从节点服务

1. 先启动AMDC主节点：切换至主节点AMDC安装目录，执行启动命令，确认主节点服务启动成功；

```
#启动从节点，指定专用配置文件
amdc-server /amdc/redis.conf

#验证从节点启动状态
ps -ef | grep amdc-server # 查看进程
tail -100f /amdc/server.log # 查看日志
```

2. 再启动所有AMDC从节点：逐个启动从节点AMDC服务，查看每个从节点的启动日志，确认无报错。

```
#启动从节点，指定专用配置文件
amdc-server /amdc/redis.conf

#验证从节点启动状态
ps -ef | grep amdc-server # 查看进程
tail -100f /amdc/server.log # 查看日志
```

4.3.3.4 主从关系验证

- 在AMDC主节点执行命令：info replication命令，查看从节点列表，确认所有从节点已正常连接主节点（master_link_status为up）；
- 在每个AMDC从节点执行命令：info replication命令，确认与主节点同步正常（offset一致）；
- 执行数据读写测试，主节点set key，从节点get key，确认数据同步正常。

4.3.4 注意事项

- 迁移全程必须先全量停止Redis主从节点，禁止数据写入，否则会导致源数据文件不一致，导入后数据丢失；
- 主节点数据导入后必须先验证数据完整性，再启动从节点，避免从节点同步错误数据；
- 从节点无需拷贝任何源数据文件，由AMDC V2.0.4版本内部主从同步完成，手动拷贝会导致同步冲突；
- 主、从节点的核心配置必须一致（AOF开关、端口、密码等），否则会导致连接失败或同步异常；
- 迁移后观察集群运行10-20分钟，确认主从同步无延迟、日志无报错、业务访问正常。

4.4 哨兵模式迁移

AMDC V2.0.4版本支持加载Redis配置redis.conf和sentinel.conf配置文件，核心流程为全量停机->先迁主从节点->后迁哨兵节点->重新关联监控，通过「主从集群重建 + 哨兵重新监控」完成迁移，全程需短暂业务停机，具体步骤如下：

4.4.1 迁移前提

- 所有主从节点、哨兵节点均已安装AMDC V2.0.4版本，网络互通（开放6379服务端口、26379哨兵端口）；
- 已收集Redis主从节点配置(redis.conf)、数据文件、哨兵配置文件(sentinel.conf)，并做好全量备份；
- 提前规划停机窗口（低峰期），停机时长为「Redis停机 + AMDC V2.0.4版本主从启动 + 哨兵启动」耗时（一般10-40分钟）；
- 所有目标节点时间同步，磁盘空间充足（主节点数据目录预留2倍源数据空间）。

4.4.2 准备工作

4.4.2.1 工具与文件准备

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的RDB缓存文件（随AMDC V2.0.4版本AMDC安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的AOF文件（同目录部署）；

4.4.2.2 数据备份并全量停止Redis哨兵 + 主从集群

- 停止业务对Redis的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端Redis配置(redis.conf)；
- 必须先停止所有Redis哨兵节点和主从节点，禁止数据写入和哨兵监控，确保源数据一致性。

```
#所有节点统一执行：停止Redis主从 + 哨兵进程
```

```
ps -ef | grep -E 'redis-server | redis-sentinel' | grep -v grep |
awk '{print $2}' | xargs kill -9
```

```
#验证进程已停止（无输出则成功）
```

```
ps -ef | grep -E 'redis-server | redis-sentinel' | grep -v grep
```

4.4.3 迁移步骤

4.4.3.1 缓存数据迁移

1. 校验并修复源Redis主节点数据文件（主节点执行，若文件无损坏则跳过修复）：

```
#若使用AOF文件（推荐）
amdc-check-aof /path/to/appendonly.aof

#若使用RDB文件
amdc-check-rdb /path/to/dump.rdb

#若文件损坏，执行修复
amdc-check-aof --fix /path/to/appendonly.aof

amdc-check-rdb --fix /path/to/dump.rdb
```

2. 将修复后的干净数据文件拷贝到AMDC V2.0.4版本主节点数据目录，并修改权限：

```
#拷贝AOF文件（推荐，若用RDB则执行下一行）
cp /path/to/appendonly.aof /amdc/

#拷贝RDB文件
cp /path/to/dump.rdb /amdc/

#赋权（避免服务无权限读取）
chmod 644 /amdc/{appendonly.aof, dump.rdb}
```

4.4.3.2 配置文件迁移 (redis.conf)

1. 将Redis主/从节点的redis.conf拷贝至AMDC主/从节点的根目录。
2. 修改Redis服务配置文件(redis.conf)，追加以下配置项到配置文件最下方：

```
license /path/to/license.xml
apusic-acls /path/to/acls.properties
worker-threads 1
```

3. 哨兵配置文件(sentinel.conf)无需修改，可拷贝至AMDC哨兵节点的根目录。

4.4.3.3 启动AMDC主从节点服务

1. 先启动AMDC主节点：切换至主节点AMDC安装目录，执行启动命令，确认主节点服务启动成功；

```
#启动从节点，指定专用配置文件
amdc-server /amdc/redis.conf

#验证从节点启动状态
ps -ef | grep amdc-server # 查看进程
tail -100f /amdc/server.log # 查看日志
```

2. 再启动所有AMDC从节点：逐个启动从节点AMDC服务，查看每个从节点的启动日志，确认无报错。

```
#启动从节点，指定专用配置文件
amdc-server /amdc/redis.conf

#验证从节点启动状态
ps -ef | grep amdc-server # 查看进程
tail -100f /amdc/server.log # 查看日志
```

4.4.3.4 启动AMDC哨兵节点服务

1. 再启动所有AMDC哨兵节点，查看哨兵服务进程与日志状态，确保启动成功。

```
#启动从节点，指定专用配置文件
amdc-sentinel /amdc/sentinel.conf

#验证从节点启动状态
ps -ef | grep amdc-sentinel # 查看进程
tail -100f /amdc/sentinel.log # 查看日志
```

4.4.3.5 哨兵功能验证

- 在任意AMDC哨兵节点执行命令：info sentinel命令，查看哨兵监控的主从节点数量、状态，确认哨兵能正常识别AMDC主从节点；
- 测试业务是否能通过哨兵自动获取新主节点地址，读写业务正常；测试完成后，恢复原AMDC主节点服务

4.4.4 注意事项

- 迁移前必须全量停止Redis哨兵 + 主从节点，避免哨兵持续监控已停机节点，或数据写入导致源文件不一致；
- 拷贝sentinel.conf后，若源端与目标端AMDC主节点IP不同，必须修改sentinel monitor参数中的主节点IP，否则哨兵无法监控到目标主节点；
- 必须先启动主从节点，再启动哨兵节点，确保哨兵启动时能正常发现主从节点，避免监控失败。
- 迁移后观察10-20分钟，重点监控：哨兵日志无报错、主从切换无异常、同步延迟在可接受范围、业务访问无中断。

4.5 集群模式迁移

AMDC V2.0.4版本支持加载Redis配置redis.conf和node.conf配置文件，核心流程为「文件准备 -> 数据校验修复 -> 部署启动 -> 验证」，具体步骤如下：

4.5.1 迁移前提

- 目标集群所有节点（与源集群节点数一致：M主N从）均已安装AMDC V2.0.4版本，网络互通；
- 已收集Redis集群所有节点的配置文件、AOF、RDB数据文件，并记录源集群拓扑信息（执行cluster nodes获取槽位分配、主从关系）；
- 提前规划长停机窗口（根据数据量调整，建议低峰期），停机时长为「数据导出 + 集群初始化 + 数据导入 + 验证」耗时（数据量10G内约30-60分钟）；
- 所有目标节点磁盘空间充足（每节点预留2倍源节点数据空间）。

4.5.2 准备工作

4.5.2.1 工具与文件准备

- [amdc-check-rdb](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的RDB缓存文件（随AMDC V2.0.4版本AMDC安装包部署，位于/amdc/目录下）；
- [amdc-check-aof](#)：用于校验和修复AMDC V2.0.2及以下版本的AMDC的AOF文件（同目录部署）；

4.5.2.2 数据备份并全量停止Redis集群（禁止任何写入）

- 停止业务对Redis的写入操作，再次执行bgsave命令，生成最新RDB文件；备份源端Redis配置(redis.conf)；

```
#所有Redis集群节点统一执行：停止进程
```

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

```
#验证进程已停止（无输出则成功）
```

```
ps -ef | grep redis-server | grep -v grep
```

4.5.3 迁移步骤

4.5.3.1 数据文件预处理（所有源节点执行）

对每个Redis主节点的AOF或RDB文件进行校验 + 修复，确保为干净数据文件，避免导入AMDC后出现数据损坏：

```
#优先使用AOF文件（每个节点单独处理，集群为分片存储，各节点数据独立）
```

```
amdc-check-aof /path/to/nodeX/appendonly.aof
```

```
#若无AOF则使用RDB文件
```

```
amdc-check-rdb /path/to/nodeX/dump.rdb
```

```
#若文件损坏，执行修复
```

```
amdc-check-aof --fix /path/to/appendonly.aof
```

```
amdc-check-rdb --fix /path/to/nodeX/dump.rdb
```

4.5.3.2 配置文件迁移（redis.conf, node.conf）

1. 将Redis每个集群节点的redis.conf拷贝至AMDC对应集群节点的根目录。
2. 修改Redis服务配置文件(redis.conf)，追加以下配置项到配置文件最下方：

```
license /path/to/license.xml
apusic-acls /path/to/acls.properties
worker-threads 1
```

3. 将Redis每个集群节点的node.conf拷贝至AMDC对应集群节点的根目录。

4.5.3.3 启动AMDC集群节点

- 逐个启动所有AMDC集群节点，执行启动命令，查看每个节点的启动日志，确认无报错（重点关注集群连接及槽位加载情况）。

```
#启动从节点，指定专用配置文件
amdc-server /amdc/redis.conf

#验证从节点启动状态
ps -ef | grep amdc-server # 查看进程
tail -100f /amdc/server.log # 查看日志
```

4.5.3.4 集群状态验证

- 在任意AMDC节点执行命令：cluster info，确认集群状态为ok，所有槽位分配完整、无未分配槽位；
- 执行命令：cluster slots，查看槽位分配与源端一致，每个节点负责的槽位正确；
- 执行数据读写测试，跨分片查询测试，确认数据完整、业务正常；

4.5.4 注意事项

- 集群迁移必须全量停机，且源集群所有节点禁止写入，因集群为分片存储，单个节点数据写入会导致分片数据不一致；
- 若源集群存在数据倾斜（某分片数据量过大），导入前需提前评估AMDC V2.0.4版本集群对应节点的磁盘、内存资源；迁移后及时对倾斜分片进行扩容，避免集群出现性能瓶颈、影响业务访问速度；
- 迁移完成后建议观察集群运行10-20分钟，重点监控：各分片数据无丢失、主节点故障转移自动触发、跨分片操作正常执行、主从同步无持续延迟、业务访问延迟符合预期。

5 常见问题排查

5.0.1 工具无法运行

问题1: 终端输入amdc-check-rdb提示"command not found" (Linux);

- 原因: 工具路径未添加到系统环境变量;
- 解决: 输入完整路径运行 (如 /usr/local/amdc/amdc-check-rdb) , 或添加环境变量 (echo "export PATH=/usr/local/amdc:\$PATH" >> /etc/profile, source /etc/profile) 。

5.0.2 校验或修复文件失败

问题1:amdc-check-rdb或amdc-check-aof 提示"Permission denied";

- 原因: 无文件读取或写入权限;
- 解决: 使用chmod修改文件权限 (chmod 644 dump.rdb), 或使用sudo运行工具 (sudo amdc-check-rdb--fix dump.rdb)。

问题2: 修复后AMDC仍无法启动;

- 原因: 文件损坏过于严重, 或工具版本与AMDC版本不匹配;
- 解决: 使用备份文件恢复, 升级或降级工具版本, 与AMDC服务版本一致。

5.0.3 amdc-cli 无法连接AMDC服务

问题1: 连接提示 "Could not connect to AMDC at 127.0.0.1:6359: Connection refused";

- 原因: AMDC服务未启动;
- 解决: 启动AMDC服务 (amdc-server /path/to/amdc.yaml, Linux) 。

问题2: 远程连接提示 "Could not connect to AMDC at 192.168.1.100:6359: Connection timed out";

- 原因: 防火墙未开放6359端口, 或AMDC配置不允许远程连接;
- 解决: 开放防火墙端口 (firewall-cmd --add-port=6359/tcp --permanent, Linux), 修改AMDC配置 (bind 0.0.0.0, protected-mode no), 重启AMDC服务。

问题3: 连接后执行命令提示 "NOAUTH Authentication required";

- 原因: AMDC服务设置了密码, 连接时未输入;

- 解决：连接时用 -a 选项输入密码，或交互模式输入 AUTH 密码。

6 总结

本文档详细介绍了AMDC V2.0.2及以下版本与Redis 6.2及以下版本向AMDC V2.0.4版本的适配迁移流程，核心总结如下：

6.1 AMDC V2.0.2及以下版本向AMDC V2.0.4版本迁移流程

- 迁移前需完成环境检查、工具与文件准备，确保权限充足、文件完整；
- 配置文件迁移通过 `amdc-conf-conv` 工具实现，需指定配置类型、源或目标文件路径，转换前务必备份目标配置文件；
- 缓存数据迁移基于RDB文件，需先通过 `amdc-check-rdb` 校验文件完整性，修复损坏文件后部署到AMDC V2.0.4安装目录；
- 迁移后需验证服务启动状态、数据完整性和功能可用性，确保迁移成功；
- 迁移过程中若遇到工具运行、文件处理、服务启动等问题，可参考常见问题排查章节快速解决。

6.2 Redis 6.2及以下版本向AMDC V2.0.4版本迁移流程

- 安装AMDC，解压安装包至目标目录，完成基础安装；
- 停止源端Redis服务，收集对应模式核心文件（配置文件+数据文件）并备份；
- 使用Redis的配置文件按对应模式顺序启动AMDC节点，确保无报错；
- 验证数据及功能正常，切换业务流量，保留源端用于回滚。

遵循本文档操作流程，可实现版本AMDC V2.0.2及以下版本与Redis 6.2及以下版本向AMDC V2.0.4版本的平滑迁移，保障缓存服务的连续性和数据安全性。

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。

Apusic
金蝶天燕

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

