



APUSIC
固若长城
睿比世界

快速使用手冊

金蝶Apusic分布式缓存 V2.0.6

版權所有 © 深圳市金蝶天燕雲計算股份有限公司2026。保留所有權利。

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文檔
 - 1.3 技術支持
- 2 緩存核心快速開始
 - 2.1 前提條件
 - 2.2 快速上手步驟
 - 2.2.1 獲取AMDC產品包
 - 2.2.2 上傳至目標服務器並解壓
 - 2.2.3 導入授權文件
 - 2.2.4 啟動AMDC服務
 - 2.2.5 使用amdc-cli工具進行數據讀寫
 - 2.2.6 使用Java語言進行數據讀寫
 - 2.2.7 安全加固（強烈建議在首次啟動後立即配置）
- 3 Web管控制台快速開始
 - 3.1 登錄控制台
 - 3.2 1. 資源準備
 - 3.2.1 1.1 上傳安裝包
 - 3.2.2 1.2 上傳 License
 - 3.2.3 1.3 添加目標機器
 - 3.2.4 1.4 確認配置模板
 - 3.3 2. 自動部署新服務
 - 3.3.1 2.1 基礎配置
 - 3.3.2 2.2 拓撲配置
 - 3.3.3 2.3 配置模板
 - 3.3.4 2.4 確認部署
 - 3.4 3. 導入已有實例
 - 3.4.1 3.1 進入導入頁面
 - 3.4.2 3.2 填寫導入信息
 - 3.4.3 3.3 關聯機器
 - 3.5 4. 服務驗證

1 前言

本文档为金蝶Apusic分布式缓存（AMDC）V2.0.6产品的快速使用手册，提供产品介绍，安装部署产品的基本步骤及验证方式。

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 V2.0.6产品相关的信息，请参阅以下AMDC V2.0.6产品手册文档集：

序 號	手册文档	说明
1	金蝶Apusic分布式缓存 V2.0.6 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 V2.0.6 安装手册	详细介绍如何在各操作系统上安装AMDC，以及AMDC服务启停操作和产品注册过程。
3	金蝶Apusic分布式缓存 V2.0.6 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 V2.0.6 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 V2.0.6 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 V2.0.6 迁移手册	详细介绍AMDC历史版本迁移升级到V2.0.6版本的说明，以及Redis迁移到AMDC的说明。
7	金蝶Apusic分布式缓存 V2.0.6 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 V2.0.6 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：

- 網址: <https://www.apusic.com>
- 電話: 400-855-5800
- 郵箱: support@apusic.com
- 金蝶雲社區: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支持時，請提供如下信息：

1. 您的姓名
2. 公司信息與聯繫方式
3. 操作系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細信息

2 緩存核心快速開始

本快速使用指南將幫助您在幾分鐘內：

1. 成功部署並啟動 AMDC 服務；
2. 使用默認配置連接到 AMDC 實例；
3. 執行基本的讀寫操作驗證功能；
4. 使用Java 客戶端連接AMDC服務。

完成本指南後，您將擁有一個可運行的 AMDC實例，並能通過命令行或客戶端進行基礎交互。

2.1 前提條件

請確保您的系統滿足以下要求：

- 操作系統：Linux（Windows Subsystem for Linux（WSL）推薦）
- CPU架構：x86_64、ARM64
- Java 17+（如 AMDC 客戶端為 Java 編寫）或對應運行時環境（根據實際技術棧調整）
- 網絡端口 6359（AMDC 默認端口）未被佔用

2.2 快速上手步驟

2.2.1 獲取AMDC產品包

- x86_64 CPU架構：`AMDC-V2.0.6-20260604-amd64.tar.gz`
- ARM64 CPU架構：`AMDC-V2.0.6-20260604-arm64.tar.gz`

請根據目標服務器CPU架構選擇對應的產品包。

注：產品包名稱中【20260604】日期可能不一致，請以實際日誌為準。

2.2.2 上傳至目標服務器並解壓

下面以產品包：`AMDC-V2.0.6-20260604-amd64.tar.gz` 為例

- 上傳 `AMDC-V2.0.6-20260604-amd64.tar.gz` 至目標服務器的 `/opt` 目錄下
- 進入/opt目錄：`cd /opt`
- 解壓：`tar -zxvf AMDC-V2.0.6-20260604-amd64.tar.gz`
- 進入AMDC目錄：`cd amdc`

2.2.3 導入授權文件

AMDC產品需要授權文件才能順利啟動，請聯繫對應的AMDC銷售人員/服務支持人員獲取最新的測試授權文件。

下面以測試授權文件壓縮包：**金蝶Apusic分布式缓存软件2.0.zip** 為例

- 獲取授權文件壓縮包：`金蝶Apusic分布式缓存软件2.0.zip`
- 上傳至目標服務器的 `/opt/amdc` 目錄下
- 解壓授權文件：`unzip 金蝶Apusic分布式缓存软件2.0.zip`
- 將KBC開頭的授權文件名修改為 `license.lic`：`mv KBCLIC-251216-000155.lic license.lic`

2.2.4 啟動AMDC服務

- 啟動AMDC服務: `./amdc-server amdc.yaml`

```

root@root:/opt/amdc$ ./amdc-server amdc.yaml
[2026-06-04 14:09:59.775 NOTICE] AMDC version=2.0.6, bits=64,
commit=00000000, modified=1, pid=324619, just started
[2026-06-04 14:09:59.777 NOTICE] AMDC KBC authorization code:
'SZTY1952469739' for mac: AC-DC-CA-7B-2E-02
[2026-06-04 14:09:59.779 NOTICE] KBC: license used to test users
[2026-06-04 14:09:59.779 NOTICE] License file currently loaded by
server: /opt/changzhi/AMDC/./src/license.lic
[2026-06-04 14:09:59.779 NOTICE] License restriction detected,
overwrite maxmemory configuration from 0B to 5.59G due to license
limitations
[2026-06-04 14:09:59.779 NOTICE] licensee=深圳市金蝶天燕云计算股份有限公
司, duration=2026-05-06 00:00:00 to 2026-08-04 23:59:59, ip=0,
maxclients=0, maxmemory=6G, CPUs=0

      _____
     /      |      _
    / / | | /  _ \ / / / / /  _ / /  _ /
00:00:00
    /  _  | / / _ / / _ / (  _ ) / / _
23:59:59
    / _ /  | _ / . _ / \ _ , _ / _ _ / _ / \ _ /
      / _ /
                                     Apusic In-Memory Data Cache
                                     License start: 2026-05-06
                                     License end: 2026-08-04
                                     IP granted: 0
                                     Licensee: 深圳市金蝶天燕云计算股
份有限公司

Version: v2.0.6    Bind: 127.0.0.1 / -::1    Port: 6359    PID:
324619

```

```
[2026-06-04 14:09:59.779 NOTICE] Server initialized
[2026-06-04 14:09:59.779 NOTICE] Loading RDB produced by version
2.0.6
[2026-06-04 14:09:59.779 NOTICE] RDB age 53 seconds
[2026-06-04 14:09:59.779 NOTICE] RDB memory usage when created 0.78
Mb
[2026-06-04 14:09:59.779 NOTICE] Done loading RDB, keys loaded: 0,
keys expired: 0.
[2026-06-04 14:09:59.779 NOTICE] DB loaded from disk: 0.000 seconds
[2026-06-04 14:09:59.780 NOTICE] Ready to accept connections tcp
```

如下所示，AMDC監聽端口為6359，AMDC服務啟動成功。

2.2.5 使用amdc-cli工具進行數據讀寫

- 啟動amdc-cli工具： `./amdc-cli`

```
root@root:/opt/amdc$ ./amdc-cli
127.0.0.1:6359> info    # 輸入info命令，查看AMDC服務信息
# Server
amdc_version:2.0.6
amdc_git_sha1:00000000
amdc_git_dirty:1
amdc_build_id:4d9754b8e1459059
amdc_build_date:2026-05-21
amdc_pid:324619
amdc_mode:standalone
os:Linux 5.10.0-281.0.0.184.oe2203sp4.aarch64 aarch64
arch_bits:64
monotonic_clock:ARM CNTVCT @ 100 ticks/us

# 以下内容略
```

- 創建一個key： `set test "hello world"`

```
127.0.0.1:6359> set test "hello world"
OK
```

- 獲取key的值: `get test`

```
127.0.0.1:6359> get test
"hello world"
```

2.2.6 使用Java語言進行數據讀寫

- AMDC兼容Redis SDK: Jedis
- 添加依賴

```
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
  <version>4.3.1</version>
</dependency>
```

- 建立連接，並寫入數據

```
package com.robin.test;

import redis.clients.jedis.Jedis;

public class JedisDemo1 {

    private Jedis jedis;

    // 測試鏈接
    void SetUp() {
        // 1.建立鏈接
```

```

        jedis = new Jedis("127.0.0.1", 6359);
    }

    // 测试String
    void testString() {

        String res = jedis.set("name", "apusic");
        System.out.println("res"+res);

        String name = jedis.get("name");
        System.out.println("name="+name);

    }

    // 释放资源
    void tearDown() {

        if(jedis!=null){
            jedis.close();
        }

    }

    public static void main(String[] args) {

        JedisDemo1 demo1 = new JedisDemo1();
        demo1.SetUp();
        demo1.testString();
        demo1.tearDown();

    }

}

```

- 其他编程语言可参考Redis客户端SDK使用说明

2.2.7 安全加固（強烈建議在首次啟動後立即配置）

AMDC 默認無密碼、綁定所有接口，嚴禁直接暴露於公網或非信任網絡。

1. 設置訪問密碼

- 編輯 `amdc.yaml`（編輯 `requirepass`）：

```
SECURITY:
  # user: "<user> <acl rules...>"
  user: ""
  # ACL 日誌的最大條目數
  acllog-max-len: 128
  # 外部 ACL 用戶文件路徑，與配置文件內用戶配置不可混用
  aclfile: ""
  # 默認用戶的密碼（等價於 ACL 的 default 用戶密碼）
  requirepass: "PASSWORD" # 修改此行的密碼
```

- 重啟AMDC服務
- 測試： `./amdc-cli -a PASSWORD`

2. 限制監聽地址

- 編輯 `amdc.yaml`（編輯 `bind`）：

```
NETWORK:
  # 監聽的網絡地址，默認僅監聽本地回環地址，註釋此行可監聽所有接口
  # （暴露在公網有安全風險）
  bind:
    - "127.0.0.1" # 修改此行的IP地址為當前機器IP
    - "-::1"
```

- 重啟AMDC服務
- 測試： `./amdc-cli -h IP`

3 Web管控台快速開始

下面以 AMDC 管理員 視角，演示服務上線與已有實例導入的快速操作流程。按本流程操作，一名不了解管控台的實施人員也能獨立完成部署與導入驗證。

3.1 登錄控制台

1. 打開瀏覽器，訪問 `http://<serverIP>:3001/`，進入登錄頁面。
2. 輸入賬號 `amdc_admin` 和對應密碼：
 - 管控台首次啟動時，會自動生成所有預置管理員賬號的初始密碼（包括 `amdc_admin`、三員管理員及租戶管理員），並寫入安裝目錄下的 `./resource/.initial_passwords` 文件。
 - 打開該文件，找到 `amdc_admin:` 後的 12 位隨機密碼作為首次登錄密碼。
 - 首次登錄成功後，系統會強制要求按當前密碼策略修改密碼，修改完成後方可進入控制台主界面。密碼過期後再次登錄也會觸發強制修改。
3. 點擊【登錄】進入控制台首頁。

 天燕云 分布式缓存



登錄



登錄

APUSICcloud APUSIC 云 天燕云  APUSIC天燕云

以上均为金蝶天燕云计算股份有限公司注册商标
© 2000-2024 金蝶天燕云计算股份有限公司版权所有

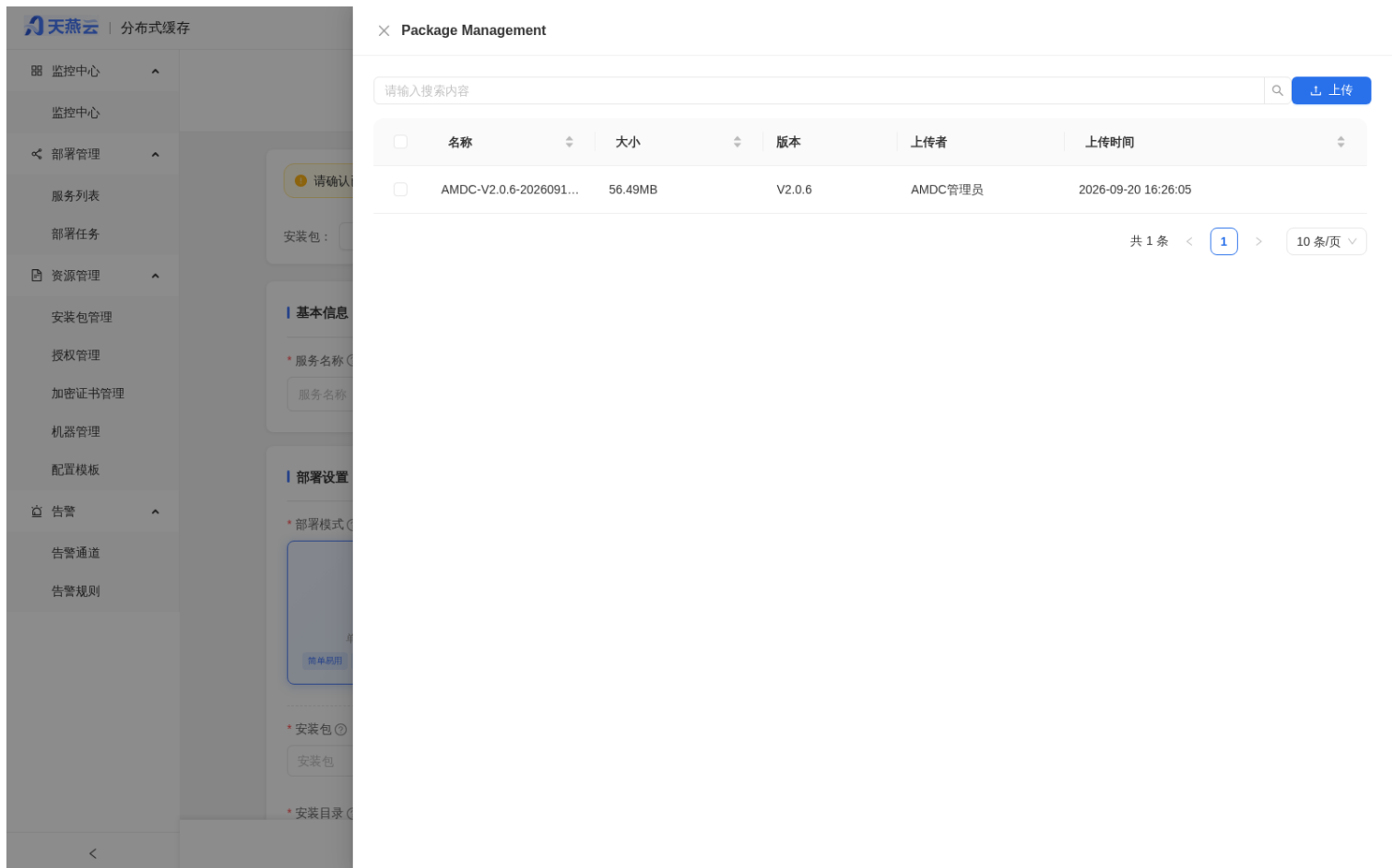
說明：本文檔後續操作均基於 AMDC 管理員 角色。其他角色（系統管理員、安全管理員、審計管理員、租戶管理員）的菜單範圍可能不同，請以實際登錄角色為準。

3.2 1. 資源準備

部署服務前，需要先準備好安裝包、License、機器和配置模板。在自動部署流程中，可直接在【基礎配置】頁面頂部點擊對應【查看】按鈕，在彈出的抽屜中完成上傳或添加，無需提前進入【資源管理】菜單。

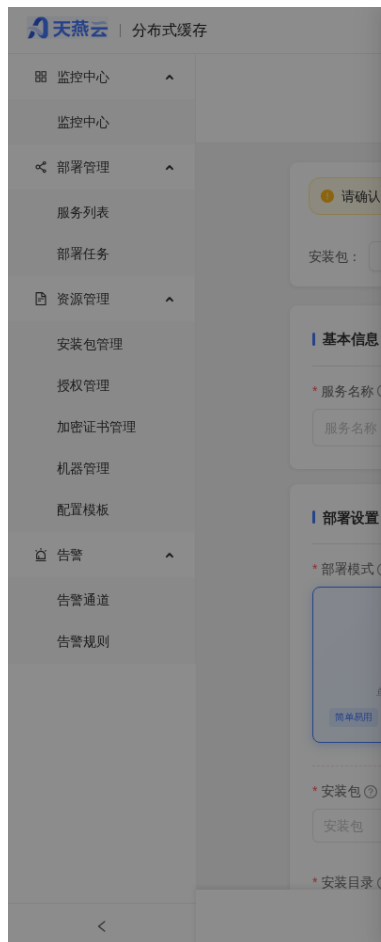
3.2.1 1.1 上傳安裝包

1. 進入【部署管理】>【自動部署】，在【基礎配置】步驟點擊頂部**安裝包**旁的【查看】按鈕。
2. 在彈出的「安裝包管理」抽屜中，點擊【上傳】按鈕，選擇目標版本的 AMDC 安裝包文件（如 `AMDC-V2.0.6-xxxx-amd64.tar.gz`）。
3. 點擊【確定】完成上傳。



3.2.2 1.2 上傳 License

1. 在自動部署【基礎配置】頁面，點擊頂部**證書**旁的【查看】按鈕。
2. 在彈出的「證書管理」抽屜中，點擊【上傳】按鈕，選擇有效的 License 文件，可同時上傳多個。
3. 上傳後系統會自動解析，無效的 License 將無法保存。



× License Management

请输入文件名称 生成授权中心文件

☐	文件名称	类型	容量	使用单位	开始时间	结束时间	上传者	上传时间	操作
☐	license.lic	LIC	6GB	深圳市...	2026-06-17 00:00:00	2026-09-30 23:59:59	AMDC管...	2026-09-20 16:28:36	编辑

共 1 条 < 1 > 10 条/页

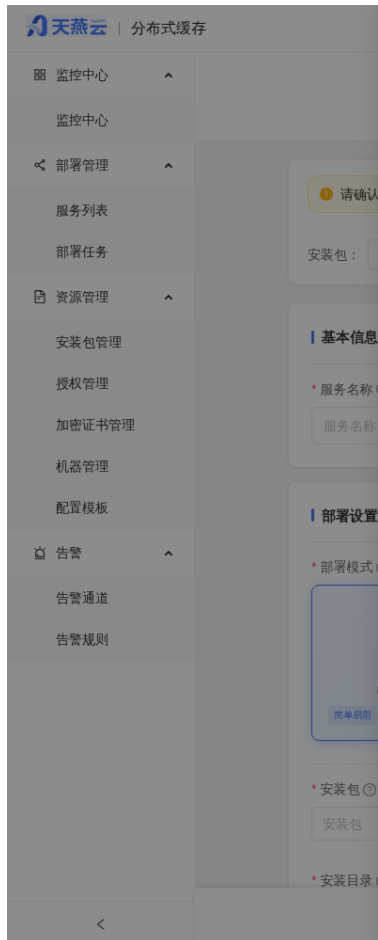
3.2.3 1.3 添加目標機器

1. 在自動部署【基礎配置】頁面，點擊頂部**機器管理**旁的【查看】按鈕。
2. 在彈出的「機器管理」抽屜中，先點擊左側機器組旁的【+】按鈕創建機器組。
3. 選中機器組，點擊左上方【添加】按鈕，填寫遠程服務器信息：

參數名	示例值	說明
用戶名	root	登錄遠程機器的用戶名
密碼	-	登錄遠程機器的密碼
SSH 端口	22	遠程 SSH 端口
Host	172.17.47.213	遠程機器 IP
備註	-	可選

4. 確認目標機器連接狀態正常後再進行後續部署。

注意：同一個機器組只能添加相同架構芯片的機器；自動部署需要目標機器具備 `tar`、`ss` 命令。



× Machine Management

机器列表

请输入搜索内容

添加

请输入搜索的Host

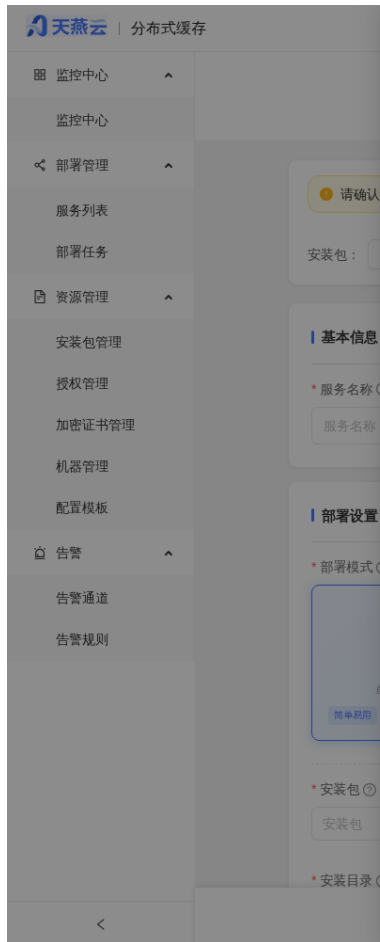
全部 SSH 直连 Apusiclet 代理

Host	状态	连接方式	用户名	机器信息	操作
☐ 172.21.61.53	在线	SSH	root	kylin x86_64 Common K.	测试连接 删除
☐ 172.21.61.66	在线	SSH	root	centos x86_64 Common	测试连接 删除
☐ 172.21.61.46	在线	SSH	root	centos x86_64 Common	测试连接 删除

共 3 条 < 1 > 10 条/页

3.2.4 1.4 確認配置模板

1. 在自動部署【基礎配置】頁面，點擊頂部**配置管理**旁的【查看】按鈕。
2. 在彈出的「配置管理」抽屜中，確認已有緩存核心配置模板和哨兵配置模板（如使用哨兵模式）。
3. 若需要自定義，可在此修改；IP、端口等非公共參數不會生效。



Config Management

配置列表

请输入搜索内容

全部

V2.0.4

V2.0.5

V2.0.6

Cache Config (amdc.yaml)

V2.0.6.amdc V2.0.6

V2.0.5.amdc V2.0.5

V2.0.4.amdc V2.0.4

Cache Config (amdc.conf)

V2.0.6.amdc V2.0.6

V2.0.5.amdc V2.0.5

V2.0.4.amdc V2.0.4

Sentinel Config (sentinel.yaml)

V2.0.6.sentinel V2.0.6

V2.0.5.sentinel V2.0.5

V2.0.4.sentinel V2.0.4

Sentinel Config (sentinel.conf)

V2.0.6.sentinel V2.0.6

V2.0.5.sentinel V2.0.5

V2.0.4.sentinel V2.0.4

编辑

保存

初始化

重置

设为默认

```
1 NETWORK:
2 # 新的网络地址, 默认仅监听本地回环地址
3 bind:
4 # "0.0.0.0"
5 # 监听地址, 连接从指定的本地IP发送, 默认不设置, 由操作系统路由表自动选择源IP
6 bind-source-addr: ""
7 # 监听的 TCP 端口, 设为 0 则不监听 TCP 套接字
8 ports: 6359
9 # 监听端口, 当未指定 bind 且无密码时, 仅允许本地回环地址和 Unix 域套接字连接
10 protected-mode: no
11 # TCP 保持时间 (秒), 支持多链路并发传输, 提升链路聚合带宽与传输吞吐, 需依赖 Linux 5.6 及以上内核版本
12 tcp-keepalive: 300
13 # TCP 监听的 backlog 大小, 实际值可能受系统 sonaxconn 和 tcp_max_syn_backlog 限制
14 tcp-backlog: 511
15 # Unix 域套接字路径, 未指定则不监听 Unix 套接字
16 unixsocket: ""
17 # Unix 域套接字的权限
18 unixsocketperm: 0
19 # 套接字超时时间 (秒), 0 表示禁用
20 timeout: 0
21 # TCP 保持时间 (秒), 用于检测死连接和维持网络连接
22 # 监听套接字指定标记 ID, 用于实现高级路由与流量过滤能力, 默认值为 0, 代表不开启套接字标记、无额外路由 / 过滤绑定
23 socket-mark-id: 0
24
25 GENERAL:
26 # 是否以守护进程模式运行
27 daemonize: no
28 # 与监控系统 (upstart/systemd) 的交互模式, 默认不交互
29 supervised: no
30 # pid 文件路径
31 pidfiles: /var/run/amdc_6359.pid
32 # 日志级别: debug (详细)、verbose (较多)、notice (适中, 推荐生产环境)、warning (仅关键信息)
33 loglevel: notice
34 # 日志文件路径, 空字符串表示输出到标准输出
35 logfile:
36 # 是否启用系统日志
37 syslog-enabled: no
38 # 系统日志标识
39 syslog-facility: local0
40 # 是否启用崩溃日志
41 crash-log-enabled: yes
42 # 是否在崩溃日志中启用快速内存检查
43 crash-check-enabled: yes
44 # 设置工作线程数, 默认为 1
45 worker-threads: 1
46 # 数据库数量, 默认使用 DB 0, 可通过 SELECT 切换
47 databases: 16
48 # 是否在启动日志中显示 ASCII 艺术 logo, 默认仅交互会话显示
49 always-show-logo: no
50 # 是否修改进程标题以显示运行时信息
51 set-proc-title: yes
52 # 进程标题模板, 支持多个变量
53 proc-title-template: "{title} {listen-addr} {server-mode}"
54 # 字符串比较 / 排序用的语言区域规则, 为空则自动继承系统环境变量的语言设置
55 locale-collate: ""
56
57 MODULES:
58 # 启动时加载指定的模块, 若加载失败服务器会终止启动, 可多次使用以加载多个模块
59 loadmodule:
```

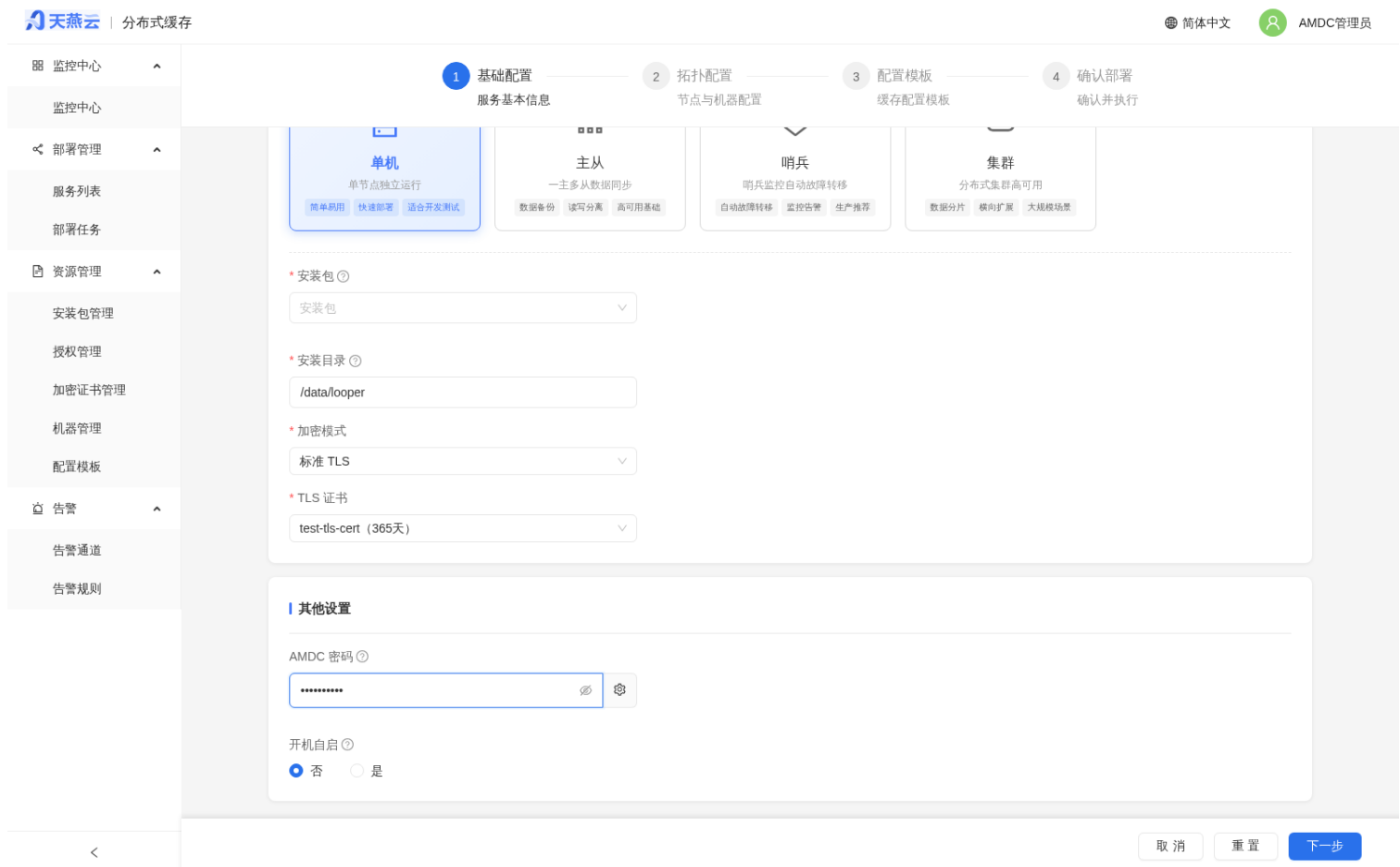
3.3 2. 自動部署新服務

資源準備完成後，繼續在自動部署嚮導中按 4 步流程完成服務部署。

3.3.1 2.1 基礎配置

在【基礎配置】步驟填寫：

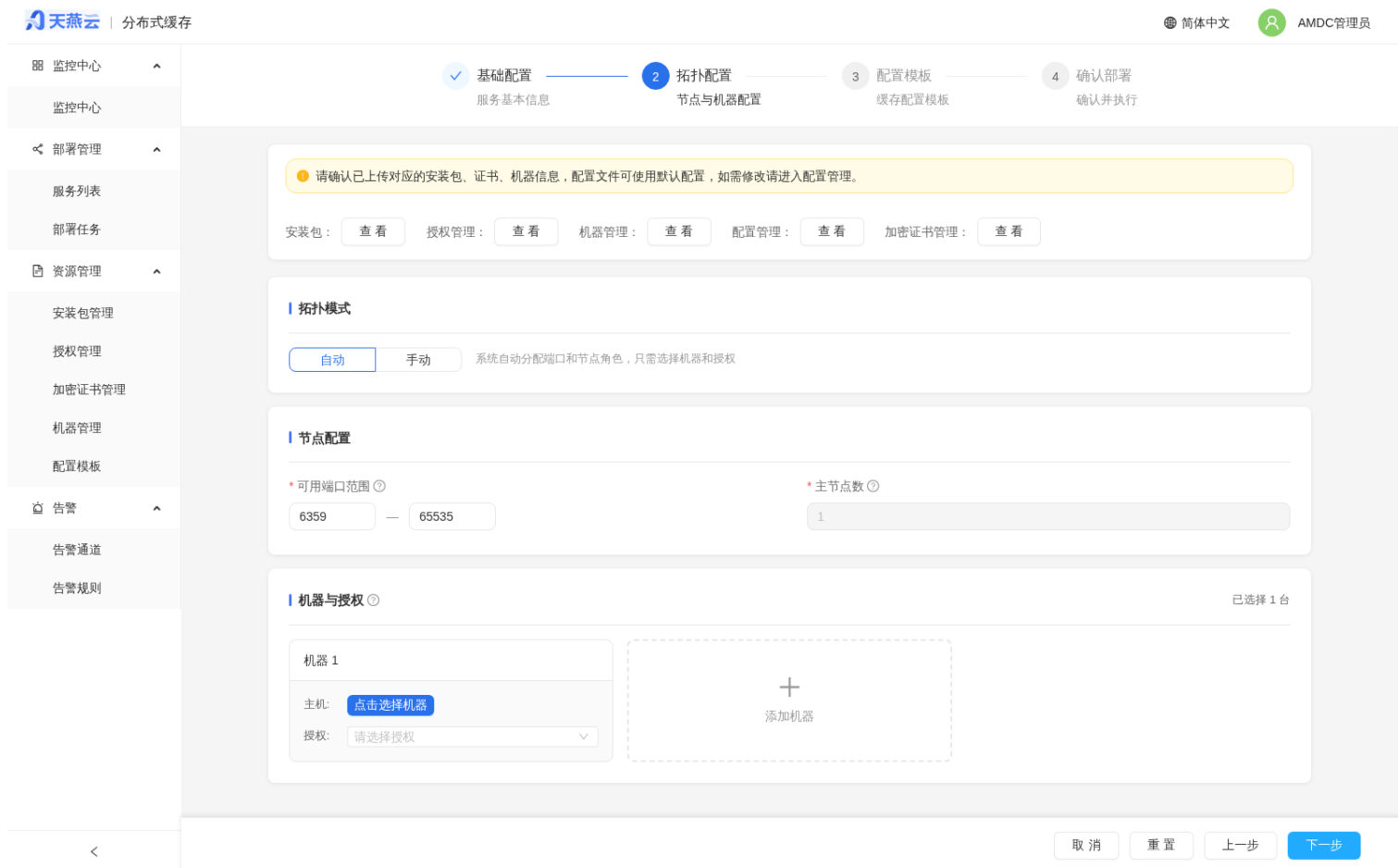
- **服務名稱**：唯一標識，僅支持英文、數字、下劃線、中劃線，不超過 24 個字符。
- **部署模式**：選擇單機 / 主從 / 哨兵 / 集群 / 預生成節點。
- **安裝包**：選擇已上傳的安裝包。
- **安裝目錄**：如 `/data/looper`，節點實際位於 `{seat}/{port}`。
- **AMDC 密碼**：緩存核心管理密碼，可點擊設置按鈕配置複雜度規則。
- **開機自啟**：是否隨系統啟動自動運行。
- **sudo**：是否使用 sudo 權限執行安裝命令。



3.3.2 2.2 拓撲配置

1. 選擇【自動】或【手動】拓撲模式：
 - **自動**：系統自動分配端口和節點角色，只需選擇機器和授權。
 - **手動**：逐節點配置機器、端口、角色和授權。
2. 配置端口範圍（默認 6359–65535）和節點數量。
3. 在【機器與授權】區域選擇目標機器，並為每台機器選擇 License 授權。

注意：所有機器需在同一機器組下；非哨兵節點必須選擇授權。



3.3.3 2.3 配置模板

選擇緩存核心配置模板；若部署模式為哨兵，還需選擇哨兵配置模板。

3.3.4 2.4 確認部署

1. 確認部署信息無誤後，點擊【執行任務】立即部署，或點擊【保存任務】僅保存待後續執行。
2. 部署過程中可進入【部署管理】>【部署任務】查看任務狀態。
3. 等待任務狀態變為「完成」。



3.4 3. 导入已有实例

對於已通過其他方式部署的 AMDC 緩存核心，可快速導入控制台進行統一管理。

3.4.1 3.1 進入導入頁面

進入【部署管理】>【服務列表】，點擊左上角【導入】按鈕，打開「服務導入」彈窗。

3.4.2 3.2 填寫導入信息

按實際實例模式填寫以下信息：

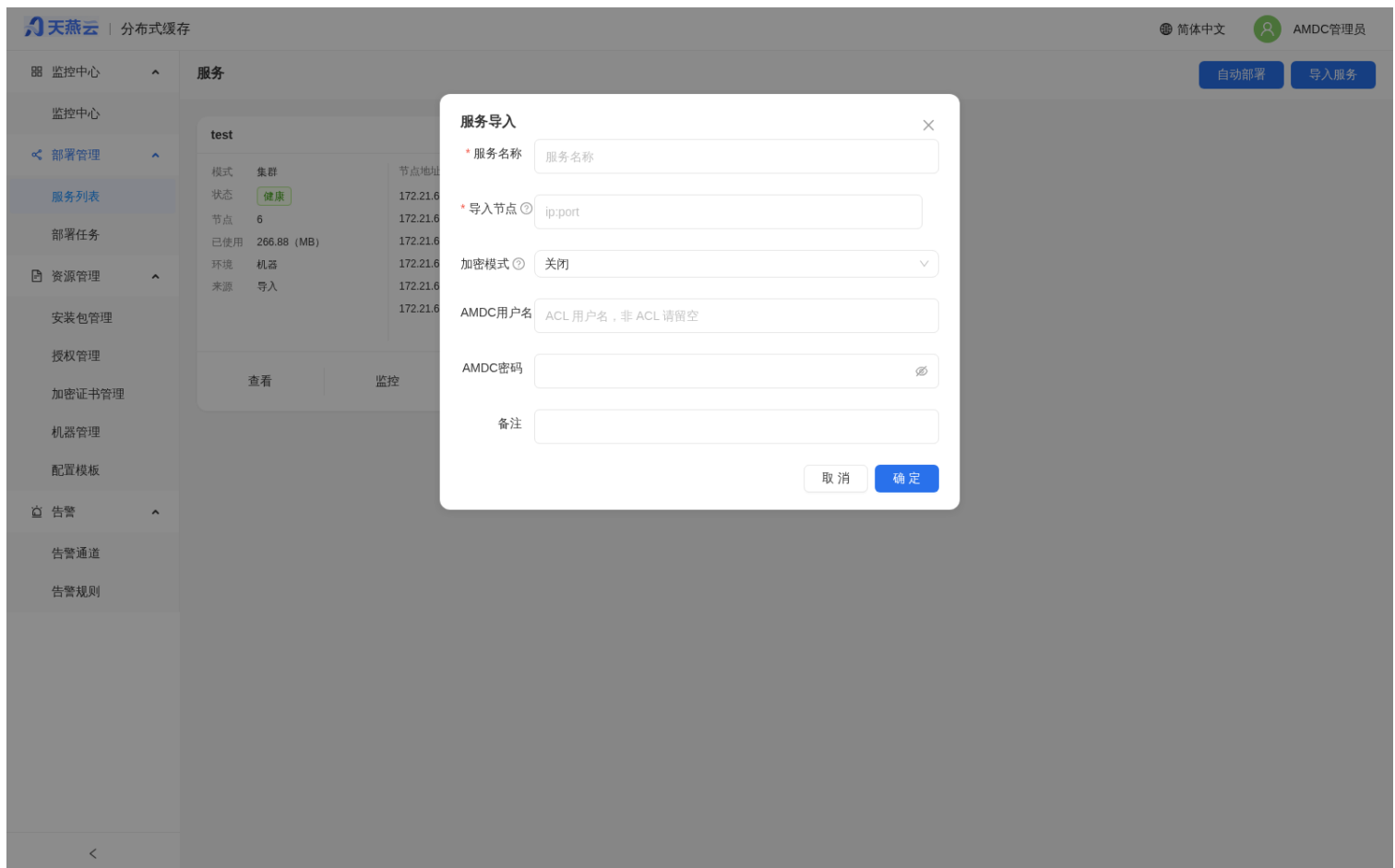
參數名	示例值	說明
服務名稱	imported_cluster	自定義服務名稱
導入節點	172.17.47.213:6359	實例節點的 IP:Port，多個節點可點擊【創建節點】添加
加密模式	關閉	實例未啟用加密傳輸時保持【關閉】；已啟用 TLS 的實例選擇對應模式（標準 TLS / 國密 TLS）

AMDC 用户名	-	ACL 用户名；未启用 ACL 请留空
AMDC 密码	-	缓存核心访问密码
备注	-	可选

注意：請按實例模式導入對應節點：

- **單機/主從模式**：導入主節點。
- **哨兵模式**：導入哨兵節點。
- **集群模式**：導入集群節點。

填寫完成後點擊【確定】，導入成功後服務列表會顯示該實例。



3.4.3 3.3 關聯機器

導入的節點默認未關聯機器信息。為使用控制台提供的安裝包升級、配置下發等完整管理功能，需進入服務詳情 > 【設置】，選中節點後點擊【關聯機器】，將節點與機器管理中的目標機器綁定。

监控中心

部署管理

服务列表

部署任务

资源管理

安装包管理

授权管理

加密证书管理

机器管理

配置模板

告警

告警通道

告警规则

🏠 / 服务列表 / 设置

test 健康

模式 集群 | 机器组 默认机器组 | 主节点 3 | 从节点 3 | 加密模式 关闭

节点 | 任务 | 连接管理 | 数据管理

命令行

启动

停止

重启

扩容

节点操作 ▾

关联机器

备份/恢复 ▾

配置 ▾

升级回退 ▾

集群运维 ▾

更新许可

■	连接状态	身份	运行状态	主机	端口	时间	关联机器	授权码	节点信息				
✓	已连接	master	正在运行	172.21.61.66	6362	2026-09-20 16:07:10	已关联	SZTY1583874825	节点信息	配置	日志	监控	升级
☐	已连接	slave	正在运行	172.21.61.66	6359	2026-09-20 16:07:10	已关联	SZTY1583874825	节点信息	配置	日志	监控	升级
☐	已连接	master	正在运行	172.21.61.46	6360	2026-09-20 16:07:10	已关联	SZTY1433595995	节点信息	配置	日志	监控	升级
☐	已连接	slave	正在运行	172.21.61.46	6363	2026-09-20 16:07:10	已关联	SZTY1433595995	节点信息	配置	日志	监控	升级
☐	已连接	master	正在运行	172.21.61.53	6361	2026-09-20 16:07:10	已关联	SZTY3966403099	节点信息	配置	日志	监控	升级
☐	已连接	slave	正在运行	172.21.61.53	6364	2026-09-20 16:07:10	已关联	SZTY3966403099	节点信息	配置	日志	监控	升级

3.5 4. 服務驗證

完成自動部署或導入後，進入【部署管理】>【服務列表】，點擊服務名稱進入服務詳情，即可查看節點狀態，或點擊頂部【命令行】按鈕執行 AMDC 命令驗證數據寫入/讀取。

全國統一服務熱線
4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。



雲計算國家標準制定企業
金蝶集團旗下基礎軟件企業
資訊技術應用創新核心企業
官網：www.apusic.com

