



命令行操作指南

金蝶Apusic应用服务器V10标准版

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 基本介绍
 - .2.1 简介
 - .2.2 术语
- .3 系统环境要求
 - .3.1 平台环境
 - .3.2 系统环境
- .4 产品清单
- .5 安装与卸载
 - .5.1 Java环境
 - .5.2 安装
 - .5.3 卸载
- .6 许可证授权
 - .6.1 普通授权
 - .6.2 集中授权
 - .6.3 获取特征码
 - .6.4 嵌入式部署方式获取特征码
- .7 启动
 - .7.1 在Linux环境下
 - .7.2 在 Windows环境下
- .8 停止
 - .8.1 在Linux环境下
 - .8.2 在 Windows环境下
- .9 命令行工具 aascli
 - .9.1 简介
 - .9.2 登录
 - .9.2.1 概要
 - .9.2.2 选项
 - .9.2.3 示例

- 9.3 退出登录
 - 9.3.1 概要
 - 9.3.2 示例
- 9.4 查看可用命令
- 9.5 命令通用约定
- 10 系统管理员功能模块
 - 10.1 应用管理
 - 10.1.1 应用管理 (app)
 - 10.1.1.1 概要
 - 10.1.1.2 选项
 - 10.1.2 列出应用 (app list)
 - 10.1.2.1 概要
 - 10.1.2.2 选项
 - 10.1.2.3 示例
 - 10.1.3 查看应用详情 (app get)
 - 10.1.3.1 概要
 - 10.1.3.2 选项
 - 10.1.3.3 示例
 - 10.1.4 更新应用配置 (app update)
 - 10.1.4.1 概要
 - 10.1.4.2 选项
 - 10.1.4.3 示例
 - 10.1.5 卸载应用 (app delete)
 - 10.1.5.1 概要
 - 10.1.5.2 选项
 - 10.1.5.3 示例
 - 10.1.6 重新加载应用 (app reload)
 - 10.1.6.1 概要
 - 10.1.6.2 示例
 - 10.1.7 部署应用 (app deploy)
 - 10.1.7.1 概要
 - 10.1.7.2 选项
 - 10.1.7.3 示例
 - 10.1.8 重新部署应用 (app redeploy)

- 10.1.8.1 概要
- 10.1.8.2 选项
- 10.1.8.3 示例
- 10.1.9 应用类库管理 (app class-lib)
 - 10.1.9.1 概要
 - 10.1.9.2 选项
 - 10.1.9.3 示例
- 10.1.10 应用增量更新 (app increment)
 - 10.1.10.1 概要
 - 10.1.10.2 选项
 - 10.1.10.3 示例
- 10.1.11 备份应用 (app backup)
 - 10.1.11.1 概要
 - 10.1.11.2 示例
- 10.1.12 应用备份管理 (app-backup)
 - 10.1.12.1 概要
 - 10.1.12.2 示例
- 10.1.13 应用回收管理 (app-recycle)
 - 10.1.13.1 概要
 - 10.1.13.2 示例
- 10.2 配置管理
 - 10.2.1 虚拟主机管理 (host)
 - 10.2.1.1 概要
 - 10.2.1.2 示例
 - 10.2.2 线程池管理 (threadpool)
 - 10.2.2.1 概要
 - 10.2.2.2 示例
 - 10.2.3 会话管理器管理 (session-manager)
 - 10.2.3.1 概要
 - 10.2.3.2 示例
 - 10.2.4 HTTP 通道管理 (httpchannel)
 - 10.2.4.1 概要
 - 10.2.4.2 示例
 - 10.2.5 JVM 配置管理 (jvm)

- 10.2.5.1 概要
- 10.2.5.2 示例
- 10.2.6 JMX 配置管理 (jmx)
 - 10.2.6.1 概要
 - 10.2.6.2 示例
- 10.2.7 证书管理 (cert)
 - 10.2.7.1 概要
 - 10.2.7.2 示例
- 10.3 资源管理
 - 10.3.1 数据源管理 (datasource)
 - 10.3.1.1 概要
 - 10.3.1.2 示例
- 10.4 监控管理
 - 10.4.1 监控配置管理 (monitor-config)
 - 10.4.1.1 概要
 - 10.4.1.2 示例
 - 10.4.2 告警配置管理 (alert-config)
 - 10.4.2.1 概要
 - 10.4.2.2 示例
 - 10.4.3 SNMP 代理配置管理 (snmp-config)
 - 10.4.3.1 概要
 - 10.4.3.2 示例
 - 10.4.4 一键巡检管理 (inspection)
 - 10.4.4.1 概要
 - 10.4.4.2 示例
 - 10.4.5 监控快照管理 (snapshot)
 - 10.4.5.1 概要
 - 10.4.5.2 示例
- 10.5 日志服务
 - 10.5.1 系统日志配置 (log-config system)
 - 10.5.1.1 概要
 - 10.5.1.2 示例
 - 10.5.2 访问日志配置 (log-config access)
 - 10.5.2.1 概要

- 10.5.2.2 示例
- 10.6 用户管理
 - 10.6.1 用户管理 (user)
 - 10.6.1.1 概要
 - 10.6.1.2 示例
- 10.7 安全服务
 - 10.7.1 安全域管理 (security-domain)
 - 10.7.1.1 概要
 - 10.7.1.2 示例
- 10.8 系统与账户
 - 10.8.1 用户设置 (usersettings)
 - 10.8.1.1 概要
 - 10.8.1.2 选项
 - 10.8.1.3 示例
 - 10.8.2 修改当前登录用户的密码 (changepassword)
 - 10.8.2.1 概要
 - 10.8.2.2 选项
 - 10.8.2.3 示例
- 11 安全管理员功能模块
 - 11.1 用户管理 (user)
 - 11.1.1 概要
 - 11.1.2 列出用户 (user list)
 - 11.1.2.1 示例
 - 11.1.3 查看用户详情 (user get)
 - 11.1.3.1 示例
 - 11.1.4 重置用户密码 (user reset-password)
 - 11.1.4.1 概要
 - 11.1.4.2 选项
 - 11.1.4.3 示例
 - 11.1.5 删除用户 (user delete)
 - 11.1.5.1 示例
 - 11.1.6 用户权限配置 (user permission)
 - 11.1.6.1 概要
 - 11.1.6.2 选项

- 11.1.6.3 示例
- 11.2 角色管理 (role)
 - 11.2.1 概要
 - 11.2.2 列出角色 (role list)
 - 11.2.2.1 概要
 - 11.2.2.2 示例
 - 11.2.3 查看角色详情 (role get)
 - 11.2.3.1 概要
 - 11.2.3.2 示例
 - 11.2.4 列出可用权限资源 (role permissions)
 - 11.2.4.1 概要
 - 11.2.4.2 示例
 - 11.2.5 创建角色 (role add)
 - 11.2.5.1 概要
 - 11.2.5.2 选项
 - 11.2.5.3 示例
 - 11.2.6 更新角色 (role update)
 - 11.2.6.1 概要
 - 11.2.6.2 示例
 - 11.2.7 删除角色 (role delete)
 - 11.2.7.1 概要
 - 11.2.7.2 示例
- 11.3 安全配置管理 (security-config)
 - 11.3.1 概要
 - 11.3.2 密码策略配置 (security-config password-policy)
 - 11.3.2.1 概要
 - 11.3.2.2 示例
 - 11.3.3 审计日志配置 (security-config audit-log)
 - 11.3.3.1 概要
 - 11.3.3.2 示例
 - 11.3.4 邮件验证配置 (security-config mail)
 - 11.3.4.1 概要
 - 11.3.4.2 示例
 - 11.3.5 动态码验证配置 (security-config otp)

- 11.3.5.1 概要
- 11.3.5.2 示例
- 11.3.6 业务配置备份管理 (security-config backup)
 - 11.3.6.1 概要
 - 11.3.6.2 示例

1 前言

本文档主要介绍金蝶Apusic应用服务器V10标准版命令行操作指引，提供了系统环境要求、安装方法等，适用于使用金蝶Apusic应用服务器进行开发的开发人员、生产环境的系统管理员、运维人员等。

1.1 适用对象

本手册主要面向对象为使用金蝶Apusic应用服务器进行应用开发的开发人员，生产环境的系统管理员，应用发布人员，技术运维人员等。具备以下技能可能会更好理解和使用金蝶Apusic应用服务器命令行操作指南内容：

- 熟悉Linux常用命令
- 基本的系统管理任务
- 安装和管理软件

1.2 相关文档

了解更多AAS-V10标准版产品相关的信息，请参阅以下AAS-V10产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic应用服务器V10标准版安装手册	详细介绍如何在各操作系统上安装AAS，以及AAS服务启停操作，产品的注册过程
2	金蝶Apusic应用服务器V10标准版快速入门	简单介绍了如何快速上手使用AAS
3	金蝶Apusic应用服务器V10标准版用户手册	详细介绍AAS相关功能的使用和操作说明，指导用户对系统进行安装部署、功能操作、参数配置、日常运维及问题自查
4	金蝶Apusic应用服务器V10标准版管控台用户手册	详细介绍AAS管控台相关功能的使用和操作说明，指导用户通过可视化管控台进行操作
5	金蝶Apusic应用服务器V10标准版命令行操作指南	详细介绍AAS通过命令行进行操作使用
6	金蝶Apusic应用服务器V10标准版监控指标文档	详细介绍AAS各类型监控指标
7	金蝶Apusic应用服务器V10标准版API文档	详细介绍AAS的API信息

1.3 技术支持

金蝶Apusic应用服务器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 基本介绍

2.1 简介

金蝶Apusic应用服务器V10标准版是一款符合JAVAE规范Web应用服务器，属于轻量级应用服务器，通常用于应用系统高效开发场合。

2.2 术语

AAS：金蝶Apusic应用服务器

AE：金蝶Apusic应用服务器标准版产品包标识

APUSIC_HOME：金蝶Apusic应用服务器安装目录

3 系统环境要求

3.1 平台环境

金蝶Apusic应用服务器兼容各类型操作系统，如下表所示。

表1：金蝶Apusic应用服务器V10标准版适应的操作系统

操作系统	说明
Windows	Windows7、Windows10、Windows Server
Linux	CentOS、Ubuntu、SUSE、Redhat
国产平台	银河麒麟系列、中标麒麟系列、普华、中科红旗、深度等

3.2 系统环境

金蝶Apusic应用服务器系统环境建议如下。

表2：金蝶Apusic应用服务器V10标准版系统环境

系统组件	系统要求
Java环境	JDK8及以上，针对HTTP2建议使用8U252以上
内存	至少需要 512MB 的内存，建议1024MB以上
磁盘空间	至少需要 1024MB 的磁盘空间，建议10G以上

4 产品清单

产品安装包通常打包为 `AAMS-[version].tar.gz` 或 `AAMS-[version].zip`。

文中 `${APUSIC_HOME}` 表示应用服务器的安装目录。

表3：金蝶Apusic应用服务器V10标准版产品包

产品包	说明
<code>AAMS-[version].zip/AAMS-[version].tar.gz</code>	金蝶Apusic应用服务器V10标准版安装包

金蝶Apusic应用服务器V10标准版产品包说明。

表4：金蝶Apusic应用服务器V10标准版产品包说明

文件	说明
<code>bin</code>	应用服务器启动、停止等脚本所在的目录
<code>conf</code>	应用服务器配置文件所在的目录
<code>console</code>	应用服务器管控台所在的目录
<code>lib</code>	应用服务器JAR文件所在的目录
<code>logs</code>	应用服务器日志文件默认目录
<code>plugins</code>	应用服务器插件所在的目录
<code>store</code>	数据仓库目录，各类数据的集中存放地
<code>webapps</code>	自动部署应用程序包的目录
<code>webapps-console</code>	通过管控台部署应用程序包的目录
<code>webapps-jakarta</code>	部署JakartaEE规范应用程序包的目录
<code>temp</code>	应用服务器临时文件存放的目录
<code>work</code>	存放JSP编译后产生的class文件
<code>license.xml</code>	产品包的授权文件

5 安装与卸载

5.1 Java环境

安装前，需要先确认java环境。JDK版本建议在1.8.0_201及以上，查看JDK版本命令如下：

```
[user@test bin]# java -version
java version "1.8.0_212"
Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
```

支持指定 JAVA 运行环境，通过在 `apusic.sh` 或 `apusic.bat` 设置，在开头添加 `JAVA_HOME`，如：

```
set JAVA_HOME="/opt/java/jdk1.8.0_271"
```

5.2 安装

准备安装包 `AAMS-V10.0.8-SP12-AE-20260701.tar.gz` 或 `AAMS-V10.0.8-SP12-AE-20260701.zip`，拷贝至有权限的目录，进行解压。

解压即安装成功。

```
执行 tar -zxvf AAMS-V10.0.8-SP12-AE-20260701.tar.gz
或 unzip AAMS-V10.0.8-SP12-AE-20260701.zip
```

5.3 卸载

删除安装目录 `${APUSIC_HOME}/` 下所有文件即可完成卸载操作。卸载将会丢失当前AAS所有已配置的文件，请谨慎操作。

6 许可证授权

AAS需要有对应的许可证才能正常使用，通常情况下，金蝶天燕会根据用户购买的产品版本配套对应的许可证，如果在使用过程中出现许可证过期或无效等问题，建议优先联系对接的天燕服务人员，重新申请对应许可证。

产品授权方式分为普通授权和集中授权。

6.1 普通授权

普通授权指根据IP、域名等方式生成 `license.xml` 文件，将授权文件放置安装目录下，

`${APUSIC_HOME}/license.xml`。



如若安装有管控平台，系统管理员admin登录管控平台，点击右上角的授权图标，点击“更新授权”，上传可用的授权文件，保存即可完成更新。重启AAS后新授权文件生效。

6.2 集中授权

集中授权指连接授权中心，进行统一授权。需要先搭建金蝶Apusic授权中心，操作方式可参考《金蝶Apusic许可授权中心用户手册》，或联系金蝶天燕技术支持人员。

在系统环境中配置环境变量，或在AAS安装目录根目录创建 `acls.properties` 文件，添加以下参数：

```
apusic_acls_enable=true
apusic_acls_authUrls=127.0.0.1:6886
```

```
apusic_acls_ns=apusic
apusic_acls_tenant=ApusicTest
```

连接参数说明：

表5：金蝶Apusic许可授权中心连接参数说明

参数名	参数值说明
apusic_acls_enable	是否开启授权中心认证，取值为true或false，为true则表示开启授权中心认证。没有该参数或该参数值为false，都表示没有开启授权中心认证；
apusic_acls_authUrls	授权中心的地址，可设置多个授权地址，格式为ip1:port1,ip2:port2，如果一个授权地址连接失败，会轮询其他的地址；如果开启授权中心认证，则为必填参数，其中端口为授权中心的https端口；
apusic_acls_ns	设置该实例所属的命名空间名称，可选参数；默认值为public，具体的命名空间可以在授权中心管理控制台-系统管理-授权管理查看。
apusic_acls_tenant	设置该实例所属的租户名称，可选参数。

AAS启动时将会自动连接到Apusic授权中心。

6.3 获取特征码

如果在使用过程中出现许可证过期或无效等问题，建议优先联系对接的天燕服务人员，重新申请对应许可证。重新申请对应许可证时，需要将产品的特征码(auth code)提供到天燕对接人员。

在 `${Apusic_HOME}/bin`，执行 `apusic.sh -ac [ethname or ip]`，`[ethname or ip]` 取值为ip地址或者网卡名称，类似如下：

```
apusic.sh -ac 127.0.0.1
```

打印特征码信息，类似如下，Auth Code=特征码内容：

```
Using APUSIC_BASE:      /opt/AAS/AAMS-V10.0.0
Using APUSIC_HOME:      /opt/AAS/AAMS-V10.0.0
Using APUSIC_TMPDIR:    /opt/AAS/AAMS-V10.0.0/temp
Using JRE_HOME:         /usr/local/java/jdk1.8.0_231/jre
Using CLASSPATH:        /opt/AAS/AAMS-
V10.0.0/bin/bootstrap.jar:/opt/AAS/AAMS-V10.0.0/bin/aas-juli.jar
```

```
Using APUSIC_OPTS:  
Auth Code=SZTY829390371
```

获取特征码后再提供特征码申请授权文件。

6.4 嵌入式部署方式获取特征码

在与springboot整合的方式下，没有获取特征码的 `apusic` 命令，需要依赖用户的应用进行获取。

如果已经有临时的授权文件可以启动应用，则先把授权文件移除，然后在启动命令中增加JVM参数 `apusic-ac`，取值为ip地址或者网卡名称，类似如下,eth3为测试机器的网卡名称，需要根据实际的网络信息修改：

```
java -jar -Dapusic-ac=eth3 boot-quickstart-1.0.jar
```

由于没有授权文件，应用启动会停止，在停止前会打印特征码信息，类似如下，Auth Code=特征码内容：

```
Auth Code=SZTY263537626  
license is invalid.
```

7 启动

7.1 在Linux环境下

安装完成并更新授权文件后，进入安装目录 `${APUSIC_HOME}/bin`，执行 `apusic.sh run`。

也可执行 `./startapusic.sh` 启动服务。

7.2 在 Windows环境下

安装完成并更新授权文件后，进入安装目录 `${APUSIC_HOME}\bin`，执行 `apusic.bat run`。

也可执行 `./startapusic.bat` 启动服务。

8 停止

8.1 在Linux环境下

1、Ctrl + C

前端启动状态下，可以直接Ctrl + C退出，停止AAS。

2、stopapusic.sh

进入安装目录 `${APUSIC_HOME}/bin`，执行 `stopapusic.sh`。

3、kill进程

可以通过kill [进程号] 方式停止。

8.2 在 Windows环境下

1、Ctrl + C

前端启动状态下，可以直接Ctrl + C退出，停止AAS

2、stopapusic.bat

进入安装目录 `${APUSIC_HOME}\bin`，执行 `stopapusic.bat`。

3、停止进程

可以打开任务管理器，找到对应进程，停止运行。

9 命令行工具 aascli

9.1 简介

AAS 提供命令行工具 `aascli`，用于对应用服务器进行日常管理与运维操作，功能与管控台保持一致。命令行工具位于 `${APUSIC_HOME}/bin` 目录下：

- Linux 环境：`aascli.sh`
- Windows 环境：`aascli.bat`

使用 `aascli` 前，需确认：

1. 已配置 `JAVA_HOME` 或 `JRE_HOME` 环境变量（JDK8 及以上）；
2. AAS 服务已启动（`aascli` 通过管控台端口与服务端通信，默认端口 6888）；
3. 拥有可登录管控台的账号。

9.2 登录

`login` 命令用于登录到 AAS 控制台，登录成功后进入交互式命令空间。系统内置三个管理员账号，对应"三员"分立的管理模型：

用户名	角色	说明
admin	sysadmin	系统管理员，负责应用、配置、资源、监控、日志等系统运维管理
secure	security	安全管理员，负责角色、安全配置等安全管理（本文档不展开）
audit	auditor	审计管理员，负责审计日志管理（本文档不展开）

9.2.1 概要

```
Usage: aascli login [-h] [--no-interactive] [--file=<loginFile>]
                  [--generate-file=<generateFile>] [-p=<password>]
```

```
[-u=<username>]
```

登录到 AAS 控制台

```
--file=<loginFile>
```

登录文件路径 (文件内含加密的登录信息，

由 `--generate-file` 生成)

```
--generate-file=<generateFile>
```

校验账号密码后生成包含加密登录信息的

登录文件并退出，不进入交互式命令空间

```
-h, --help
```

显示帮助信息

```
--no-interactive
```

出(用于脚本自动化)

登录成功后不进入交互式命令空间,直接退出

```
-p, --password=<password>
```

密码

```
-u, --username=<username>
```

用户名

9.2.2 选项

```
-u, --username=<username>
```

登录用户名。系统管理员默认为 admin。未提供时交互式输入。

```
-p, --password=<password>
```

登录密码。未提供时交互式输入（输入不回显）。脚本自动化场景可直接指定。

```
--no-interactive
```

登录成功后不进入交互式命令空间，直接退出，用于脚本自动化中校验账号可用性。

```
--generate-file=<generateFile>
```

校验账号密码后生成包含加密登录信息的登录文件并退出，不进入交互式命令空间。生成的登录文件可配合 `--file` 使用，避免在脚本中明文保存密码。

```
--file=<loginFile>
```

使用登录文件登录，文件内含加密的登录信息，由 `--generate-file` 生成。

9.2.3 示例

示例1 以系统管理员身份登录

进入 `${APUSIC_HOME}/bin` 目录执行 `aascli`，在交互式提示中输入用户名、密码（密码不回显）：

```
[user@test bin]$ ./aascli.sh
aas> login
用户名: admin
密码:
```

```
admin登录成功
admin@localhost>
```

也可以在登录命令中直接指定用户名密码：

```
[user@test bin]$ ./aascli.sh login -u admin -p [YOURPASSWORD]
admin登录成功
admin@localhost>
```

登录成功后进入交互式命令空间，提示符为 `用户名@主机名>`，即可输入各功能子命令。

示例2 密码错误时的提示

密码连续错误会锁定账号，请谨慎操作：

```
admin@localhost> 登录失败：用户名或密码错误，剩余重试次数 4 次
```

示例3 生成登录文件并用于脚本自动化

先生成加密登录文件，再使用登录文件登录，避免脚本中出现明文密码：

```
[user@test bin]$ ./aascli.sh login -u admin -p [YOURPASSWORD] --
generate-file=/home/user/aas.login
登录文件已生成： /home/user/aas.login

[user@test bin]$ ./aascli.sh login --file=/home/user/aas.login
admin登录成功
admin@localhost>
```

示例4 校验账号可用性后直接退出

```
[user@test bin]$ ./aascli.sh login -u admin -p [YOURPASSWORD] --no-
interactive
admin登录成功
```

9.3 退出登录

`logout` 命令用于退出当前登录，退出交互式命令空间可直接使用 `exit` 或 `Ctrl + D`。

9.3.1 概要

```
Usage: logout [-h]
```

退出登录

-h, --help 显示帮助信息

9.3.2 示例

示例1 退出登录

```
admin@localhost> logout
```

9.4 查看可用命令

`help` 命令用于显示当前登录用户可用的命令列表。命令列表与登录用户的角色权限相关，系统管理员登录后可见的命令如下：

```
admin@localhost> help
```

应用管理：

app	应用列表
app-backup	应用备份
app-recycle	应用回收

配置管理：

host	虚拟主机管理
threadpool	线程池管理
session-manager	会话管理器管理
httpchannel	HTTP 通道管理
jvm	JVM 配置管理
jmx	JMX 配置管理
cert	证书管理

资源管理：

datasource	数据源管理
------------	-------

监控管理：

monitor-config	监控配置管理
----------------	--------

```

alert-config 告警配置管理
snmp-config SNMP 代理配置管理
inspection 一键巡检管理
snapshot 监控快照管理（手动创建、下载、删除、自动配置）

```

日志服务：

```
log-config 日志配置管理
```

用户管理：

```
user 用户管理
```

安全服务：

```
security-domain 安全域管理
```

系统与账户：

```

usersettings 用户设置
changepassword 修改当前登录用户的密码
logout 退出登录
help 显示可用命令

```

不同角色登录后可见的命令集合不同：系统管理员的功能模块参见"系统管理员功能模块"章节，安全管理员的功能模块参见"安全管理员功能模块"章节；审计管理员的功能模块本文档暂不描述。

9.5 命令通用约定

- 命令结构：**命令采用 **主命令** [**子命令**] [**参数**] [**选项**] 的结构，如 `app list --page=2`。输入 **主命令** `--help` 或 **主命令** **子命令** `--help` 可查看对应层级的使用说明。
- 帮助选项：**所有命令均支持 `-h` 或 `--help` 显示帮助信息。
- 确认提示：**删除类操作默认会提示确认，输入 `y` 确认执行，其他输入取消；指定 `-f` 或 `--force` 则不再提示确认，适用于脚本自动化场景。
- 输出格式：**列表查询类命令支持 `--output=table|json` 设置输出格式，默认 `table`；`--page=<页码>` 指定显示第几页；`--no-paging` 关闭分页、一次输出全部记录。详情查询类命令支持 `--output=json` 以 `json` 格式输出。
- 生效说明：**部分配置类操作提示"重启服务器后生效"，需重启 AAS 后新配置才生效。

10 系统管理员功能模块

10.1 应用管理

10.1.1 应用管理 (app)

`app` 子命令用于应用管理，包括应用的部署、卸载、更新、备份、类库与增量包管理等，通常需要带具体的请求操作执行。

10.1.1.1 概要

```
admin@localhost> app
Missing required subcommand
Usage:  app [-h] [COMMAND]
```

应用列表

`-h, --help` 显示帮助信息

Commands:

<code>list</code>	列出应用
<code>get</code>	查看应用详情
<code>update</code>	更新应用配置
<code>delete</code>	卸载应用
<code>reload</code>	重新加载应用
<code>deploy</code>	部署应用
<code>redeploy</code>	重新部署应用
<code>class-lib</code>	应用类库管理
<code>increment</code>	应用增量更新
<code>backup</code>	备份应用

10.1.1.2 选项

```
list
```

列出应用。

```
get
```

查看应用详情。

```
update
```

更新应用配置。

```
delete
```

卸载应用。

```
reload
```

重新加载应用。

```
deploy
```

部署应用。

```
redeploy
```

重新部署应用。

```
class-lib
```

应用类库管理。

```
increment
```

应用增量更新。

```
backup
```

备份应用。

10.1.2 列出应用 (app list)

`app list` 子命令用于列出当前已部署的应用，展示应用名称、上下文路径、虚拟主机等信息。

10.1.2.1 概要

```
Usage: app list [--page=<page>] [--output=<format>] [--no-paging] [-h]  
列出应用
```

<code>-h, --help</code>	显示帮助信息
<code>--page=<page></code>	页码
<code>--no-paging</code>	本次命令关闭分页输出
<code>--output=<output></code>	输出格式: table 或 json

10.1.2.2 选项

```
--page=<page>
```

设置页码，展示第几页的信息。

```
--no-paging
```

在当前子命令中禁用分页。

```
--output=<output>
```

设置输出的格式，可设置为table或json。

table：表示输出的格式为表格，为默认值。

json：表示输出的格式为json。

10.1.2.3 示例

示例1 列出当前已部署的应用

```
admin@localhost> app list
| 应用名称 | 上下文路径 | 虚拟主机 | 部署顺序 | 部署时间 | 密级 |
+-----+-----+-----+-----+-----+-----+
---+
| testapp | /testapp | localhost | 100 | 2026-09-27 14:39 | 秘密 |
| ROOT | / | localhost | 100 | 2026-09-27 14:13 | 秘密 |
第 1 / 1 页, 共 2 条
```

示例2 以json格式列出当前已部署的应用

```
admin@localhost> app list --output=json
{
  "pagination" : {
    "total" : 1,
    "pages" : 1,
    "page" : 1
  },
  "data" : [ {
    "id" : 1,
    "appName" : "ROOT",
    "contextPath" : "/",
    "status" : 1,
    "classLoaderOrder" : null,
    "jspPrecompile" : false,
    "virtualHost" : "localhost",
    "sessionManager" : null,
    "deployOrder" : 100,
    "sharedLib" : null,
    "sharedLibId" : null,
    "deployTimeout" : 300,
    "classLevel" : "秘密",
    "description" : null,
    "deployTime" : "2026-09-27 14:13",
    "recycledTime" : null,
    "securityDomain" : null,
    "deployProps" : null,
    "managedByConsole" : true,
    "docBase" : "D:\\AAMS-V10.0.8\\webapps-console\\ROOT",
    "deploySource" : "CONSOLE"
  } ]
}
```

10.1.3 查看应用详情 (app get)

`app get` 子命令用于查看指定应用的配置详情。

10.1.3.1 概要

```
Usage: app get <NAME> [--virtual-host=<host>] [--output=<format>] [-h]
```

查看应用详情

<code><name></code>	应用名称
<code>--virtual-host=<virtualHost></code>	虚拟主机名称, 可通过 <code>aascli host list</code> 查询; 存在多个同名应用时必须填
<code>--output=<output></code>	输出格式: json
<code>--help</code>	显示帮助信息

10.1.3.2 选项

```
<name>
```

应用名称, 可通过 `app list` 查询。

```
--virtual-host=<virtualHost>
```

虚拟主机名称。当不同虚拟主机下存在多个同名应用时必须填, 用于定位具体应用。

```
--output=<output>
```

输出格式: json。不指定时以格式化文本展示。

10.1.3.3 示例

示例1 查看应用详情

```
admin@localhost> app get ROOT
```

应用名称: ROOT

上下文路径: /

类加载顺序:

JSP 预编译: 否

虚拟主机: localhost

部署顺序: 100

安全域: -

密级: 秘密

应用描述: -

10.1.4 更新应用配置 (app update)

`app update` 子命令用于更新已部署应用的配置项，可选择更新后立即重新加载应用使配置生效。

10.1.4.1 概要

Usage: `app update <NAME> [--virtual-host=<host>] [选项] [--reload]`

更新应用配置

<code><name></code>	应用名称
<code>--virtual-host=<virtualHost></code>	虚拟主机名称，可通过
<code>aascli host list</code> 查询；存在多个同名应用时必须填	
<code>--reload</code>	更新成功后立即重新加载应用，使配置生效
<code>--context-path=<contextPath></code>	访问应用的上下文路径，默认为应用名称，同一虚拟主机内不可重复
<code>--class-loader-order=<classLoaderOrder></code> (父类优先) 或 <code>child-first</code> (应用优先)	类加载顺序: <code>parent-first</code>
<code>--jsp-precompile=<jspPrecompile></code> 文件: <code>true</code> 或 <code>false</code>	部署时是否预编译所有 JSP
<code>--deploy-order=<deployOrder></code> 优先, 范围 0-2147483647	应用启动优先级, 数值越小越
<code>--security-domain=<securityDomain></code> <code>aascli security-domain list</code> 查看可选安全域	应用使用的安全域, 可通过
<code>--session-manager=<sessionManager></code> 过 <code>aascli session-manager list</code> 查看可选项; 传	应用绑定的会话管理器, 可通
<code>--class-level=<classLevel></code> 密、4=绝密	空字符串可解绑 应用安全密级: 2=秘密、3=机
<code>--description=<description></code>	应用备注说明, 最多1000字符
<code>--deploy-props=<deployProps></code> 式: <code>key1=value1,key2=value2</code> ; 置空可传空字符串	自定义部署参数键值对, 格

10.1.4.2 选项

`--reload`

更新成功后立即重新加载应用，使配置即时生效。不指定时配置在应用下次加载时生效。

`--context-path=<contextPath>`

访问应用的上下文路径，默认为应用名称，同一虚拟主机内不可重复。

```
--class-loader-order=<classLoaderOrder>
```

类加载顺序：parent-first（父类优先）或 child-first（应用优先）。

```
--jsp-precompile=<jspPrecompile>
```

部署时是否预编译所有 JSP 文件：true 或 false。

```
--deploy-order=<deployOrder>
```

应用启动优先级，数值越小越优先，范围 0-2147483647。

```
--security-domain=<securityDomain>
```

应用使用的安全域，可通过 `security-domain list` 查看可选安全域。

```
--session-manager=<sessionManager>
```

应用绑定的会话管理器，可通过 `session-manager list` 查看可选项；传空字符串可解绑。

```
--class-level=<classLevel>
```

应用安全密级：2=秘密、3=机密、4=绝密。

```
--description=<description>
```

应用备注说明，最多1000字符。

```
--deploy-props=<deployProps>
```

自定义部署参数键值对，格式：key1=value1,key2=value2；置空可传空字符串。

10.1.4.3 示例

示例1 更新应用备注并立即重新加载应用

```
admin@localhost> app update testapp --description=cli-guide-demo --
reload
```

应用配置更新成功。
应用重新加载成功。

示例2 修改应用启动优先级

```
admin@localhost> app update testapp --deploy-order=50
```

应用配置更新成功。

10.1.5 卸载应用 (app delete)

`app delete` 子命令用于卸载应用。若回收站功能已启用，卸载的应用将进入回收站；否则直接删除。

10.1.5.1 概要

```
Usage: app delete <NAME> [--virtual-host=<host>] [-f] [-h]
```

卸载应用

<code><name></code>	应用名称
<code>--virtual-host=<virtualHost></code>	虚拟主机名称，可通过 <code>aascli host list</code> 查询；存在多个同名应用时必填
<code>-f, --force</code>	不提示确认
<code>--help</code>	显示帮助信息

10.1.5.2 选项

```
-f, --force
```

不提示确认，直接卸载。适用于脚本自动化场景。

10.1.5.3 示例

示例1 卸载应用（需确认）

确认提示中显示的是应用的数字 ID，可通过 `app get` 查询：

```
admin@localhost> app delete testapp
```

确定要卸载应用 11 吗？(y/N) y

应用卸载成功。

示例2 卸载应用（免确认）

```
admin@localhost> app delete testapp -f
```

应用卸载成功。

10.1.6 重新加载应用 (app reload)

`app reload` 子命令用于重新加载应用，通常用于应用内容更新后使其重新生效。

10.1.6.1 概要

```
Usage: app reload <NAME> [--virtual-host=<host>] [-h]
```

重新加载应用

<code><name></code>	应用名称
<code>--virtual-host=<virtualHost></code>	虚拟主机名称，可通过 <code>aascli host list</code> 查询；存在多个同名应用时必须填
<code>--help</code>	显示帮助信息

10.1.6.2 示例

示例1 重新加载应用

```
admin@localhost> app reload testapp
```

应用重新加载成功。

10.1.7 部署应用 (app deploy)

`app deploy` 子命令用于部署一个新的应用，应用包为 WAR 文件或应用文件夹。

10.1.7.1 概要

```
Usage: app deploy --app-name=<name> --app-path=<path> [选项] [-h]
```

部署应用

<code>--app-name=<appName></code>	应用名称，仅支持字母、数字、下划线、短横线或点号，最多255字符
<code>--app-path=<appPath></code>	应用 WAR 文件或应用文件夹在服务器上的路径
<code>--context-path=<contextPath></code>	访问应用的上下文路径，默认为应用名称，同一虚拟主机内不可重复
<code>--class-loader-order=<classLoaderOrder></code>	类加载顺序：parent-first (父类优先) 或 child-first (应用优先)，默认

`--jsp-precompile=<jspPrecompile>`
文件, 默认 false

`--virtual-host=<virtualHost>`
localhost, 可通过 `aascli host list` 查询可选虚拟

`--deploy-order=<deployOrder>`
优先, 范围 0-2147483647, 默认 100

`--security-domain=<securityDomain>`
`aascli security-domain list` 查看可选安全域

`--session-manager=<sessionManager>`
过 `aascli session-manager list` 查看可选项

`--class-level=<classLevel>`
密、4=绝密, 默认 2

`--description=<description>`

`--deploy-props=<deployProps>`
式: `key1=value1,key2=value2`

`--help`

child-first

部署时是否预编译所有 JSP

应用绑定的虚拟主机, 默认

主机

应用启动优先级, 数值越小越

应用使用的安全域, 可通过

应用绑定的会话管理器, 可通

应用安全密级: 2=秘密、3=机

应用备注说明, 最多1000字符

自定义部署参数键值对, 格

显示帮助信息

10.1.7.2 选项

`--app-name=<appName>`

应用名称, 仅支持字母、数字、下划线、短横线或点号, 最多255字符。必填。

`--app-path=<appPath>`

应用 WAR 文件或应用文件夹在服务器上的路径。必填。路径需为服务器可访问的本地路径, 建议避免包含中文。

`--context-path=<contextPath>`

访问应用的上下文路径, 默认为应用名称, 同一虚拟主机内不可重复。

`--class-loader-order=<classLoaderOrder>`

类加载顺序: parent-first (父类优先) 或 child-first (应用优先), 默认 child-first。

`--jsp-precompile=<jspPrecompile>`

部署时是否预编译所有 JSP 文件，默认 false。

```
--virtual-host=<virtualHost>
```

应用绑定的虚拟主机，默认 localhost。

```
--deploy-order=<deployOrder>
```

应用启动优先级，数值越小越优先，范围 0-2147483647，默认 100。

```
--security-domain=<securityDomain>
```

应用使用的安全域。

```
--session-manager=<sessionManager>
```

应用绑定的会话管理器。

```
--class-level=<classLevel>
```

应用安全密级：2=秘密、3=机密、4=绝密，默认 2。

```
--description=<description>
```

应用备注说明，最多1000字符。

```
--deploy-props=<deployProps>
```

自定义部署参数键值对，格式：key1=value1,key2=value2。

10.1.7.3 示例

示例1 部署应用文件夹

```
admin@localhost> app deploy --app-name=testapp --app-path=D:/aas-
testapp --description=cli-demo
应用创建成功，ID：6
应用部署成功。
```

示例2 部署 WAR 包并指定上下文路径与密级

```
admin@localhost> app deploy --app-name=demo --app-
path=/opt/apps/demo.war --context-path=/demo --class-level=3
应用创建成功, ID: 7
应用部署成功。
```

注意: `--app-path` 必须为服务器本机可访问的路径, 建议使用不含中文与空格的路径。

10.1.8 重新部署应用 (app redeploy)

`app redeploy` 子命令用于以新的应用包重新部署指定应用, 应用配置未指定的项保持不变。

10.1.8.1 概要

```
Usage: app redeploy <NAME> [--virtual-host=<host>] --app-path=<path>
[选项] [-h]
重新部署应用
    <name>                                应用名称
    --virtual-host=<virtualHost>          虚拟主机名称, 可通过
aascli host list 查询; 存在多个同名应用时必须填
    --app-path=<appPath>                  应用 WAR 文件或应用文件夹
在服务器上的路径
    --context-path=<contextPath>          访问应用的上下文路径, 默认
为应用名称, 同一虚拟主机内不可重复
    --class-loader-order=<classLoaderOrder> 类加载顺序: parent-first
(父类优先) 或 child-first (应用优先)
    --jsp-precompile=<jspPrecompile>      部署时是否预编译所有 JSP
文件: true 或 false
    --deploy-order=<deployOrder>          应用启动优先级, 数值越小越
优先, 范围 0-2147483647
    --security-domain=<securityDomain>    应用使用的安全域, 可通过
aascli security-domain list 查看可选安全域
    --class-level=<classLevel>            应用安全密级: 2=秘密、3=机
密、4=绝密
    --description=<description>           应用备注说明, 最多1000字符
    --deploy-props=<deployProps>         自定义部署参数键值对, 格
式: key1=value1,key2=value2
    --help                                显示帮助信息
```

10.1.8.2 选项

```
--app-path=<appPath>
```

新的应用 WAR 文件或应用文件夹在服务器上的路径。必填。

其余选项同 `app deploy`。

10.1.8.3 示例

示例1 重新部署应用

```
admin@localhost> app redeploy testapp --app-path=D:/aas-testapp
应用重新部署成功。
```

10.1.9 应用类库管理 (app class-lib)

`app class-lib` 子命令用于管理指定应用的共享类库（JAR 包），包括列出、添加、删除类库。

10.1.9.1 概要

```
Usage:  app class-lib [-h] [COMMAND]
```

应用类库管理

```
-h, --help    显示帮助信息
```

Commands:

```
list      列出应用类库
add       添加应用类库
delete    删除应用类库
```

```
Usage: app class-lib list <NAME> [--virtual-host=<host>] [--page=
<page>] [--output=<format>] [--no-paging] [-h]
```

列出应用类库

```
Usage: app class-lib add <NAME> [--virtual-host=<host>] --file <jar>
[-h]
```

添加应用类库

```
Usage: app class-lib delete <NAME> [--virtual-host=<host>] --jar-name
<name> [-f] [-h]
```

删除应用类库

10.1.9.2 选项

```
--file=<file>
```

`add` 时指定 JAR 文件路径。

```
--jar-name=<jarName>
```

`delete` 时指定类库 JAR 文件名, 可通过 `app class-lib list` 查询。

10.1.9.3 示例

示例1 添加并查看应用类库

```
admin@localhost> app class-lib add testapp --file=D:/aas-h2.jar
类库上传成功。
```

```
admin@localhost> app class-lib list testapp
| 类库名称 | 大小 | 版本 |
+-----+-----+-----+
| aas-h2.jar | 2.49 MB | - |
| demo.jar | 4.69 KB | - |
第 1 / 1 页, 共 2 条
```

示例2 删除应用类库

```
admin@localhost> app class-lib delete testapp --jar-name=aas-h2.jar -f
类库删除成功。
```

10.1.10 应用增量更新 (app increment)

`app increment` 子命令用于管理指定应用的增量更新包, 适用于仅更新应用部分文件的场景。

10.1.10.1 概要

```
Usage: app increment [-h] [COMMAND]
```

应用增量更新

`-h, --help` 显示帮助信息

Commands:

`list` 列出应用增量包
`add` 添加应用增量包

delete 删除应用增量包

```
Usage: app increment list <NAME> [--virtual-host=<host>] [--page=
<page>] [--output=<format>] [--no-paging] [-h]
```

列出应用增量包

```
Usage: app increment add <NAME> [--virtual-host=<host>] --file <pkg>
[--description <desc>] [-h]
```

添加应用增量包

```
Usage: app increment delete <NAME> [--virtual-host=<host>] --
increment-id <ID> [-f] [-h]
```

删除应用增量包

10.1.10.2 选项

```
--file=<file>
```

add 时指定增量包文件路径（zip 格式压缩包）。

```
--description=<description>
```

增量包描述。

```
--increment-id=<incrementId>
```

delete 时指定增量包 ID，可通过 `app increment list` 查询。

10.1.10.3 示例

示例1 添加增量包

增量包上传后将自动应用到目标应用：

```
admin@localhost> app increment add testapp --file=D:/aas-increment.zip
--description=update-web-xml
增量包上传并应用成功，ID: 1
```

示例2 列出应用增量包

```
admin@localhost> app increment list testapp
```

id	名称	大小	描述	上传时间
----	----	----	----	------

```
+-----+-----+-----+-----+-----+-----+
| 1 | aas-increment.zip | 236 B | update-web-xml | 2026-09-27 14:53 |
第 1 / 1 页, 共 1 条
```

示例3 删除应用增量包

```
admin@localhost> app increment delete testapp --increment-id=1 -f
增量包删除成功。
```

10.1.11 备份应用 (app backup)

`app backup` 子命令用于对指定应用进行手动备份，备份文件可用于恢复应用。

10.1.11.1 概要

```
Usage: app backup <NAME> [--virtual-host=<host>] [-h]
```

备份应用

<code><name></code>	应用名称
<code>--virtual-host=<virtualHost></code>	虚拟主机名称，可通过 <code>aascli host</code>
<code>list</code>	查询；存在多个同名应用时必须填
<code>--help</code>	显示帮助信息

10.1.11.2 示例

示例1 备份应用

```
admin@localhost> app backup testapp
应用备份创建成功。
```

10.1.12 应用备份管理 (app-backup)

`app-backup` 子命令用于管理应用备份，包括查看、下载、恢复、删除备份及备份设置。

10.1.12.1 概要

```
Usage: app-backup [-h] [COMMAND]
```

应用备份

<code>-h, --help</code>	显示帮助信息
Commands:	
<code>list</code>	列出应用备份

delete	删除应用备份
download	下载应用备份
restore	从备份恢复应用
settings	备份设置

10.1.12.2 示例

示例1 列出应用备份

```
admin@localhost> app-backup list
| 备份名称 | 应用名称 | 上下文路径 | 备份时间 |
大小 |
+-----+-----+-----+-----+
----+-----+
| testapp_1790491044660.tar.gz | testapp | /testapp | 2026-09-27
14:37 | 4.45 KB |
第 1 / 1 页, 共 1 条
```

示例2 下载应用备份

```
admin@localhost> app-backup download testapp_1790491044660.tar.gz --
file=D:/aas-testapp-backup.tar.gz
备份已下载到: D:\aas-testapp-backup.tar.gz
```

示例3 从备份恢复应用

```
admin@localhost> app-backup restore testapp_1790491044660.tar.gz
应用从备份恢复成功。
```

`restore` 支持在恢复时覆盖原备份中的应用配置（上下文路径、虚拟主机、密级等），不指定时使用备份中的原值：

```
admin@localhost> app-backup restore testapp_1790491044660.tar.gz --
context-path=/testapp2 --class-level=3
应用从备份恢复成功。
```

示例4 删除应用备份

```
admin@localhost> app-backup delete testapp_1790491044660.tar.gz -f
```

备份删除成功。

示例5 查看备份设置

```
admin@localhost> app-backup settings get
```

备份设置

备份目录: \${apusic.home}/store/app_backup

备份保留天数: 7

示例6 修改备份设置

```
admin@localhost> app-backup settings set --
path=${apusic.home}/store/app_backup --retention-days=7
```

备份设置已保存。

`settings set` 选项说明:

选项	说明
<code>--path=</code>	备份目录
<code>--retention-days=</code>	备份保留天数, 范围 1-365

10.1.13 应用回收管理 (app-recycle)

`app-recycle` 子命令用于管理应用回收站。启用回收站后, 卸载的应用将进入回收站, 可还原或彻底删除。

10.1.13.1 概要

```
Usage: app-recycle [-h] [COMMAND]
```

应用回收

`-h, --help` 显示帮助信息

Commands:

`list` 列出回收站应用

`restore` 还原回收站应用

`delete` 彻底删除回收站应用

`settings` 回收设置

10.1.13.2 示例

示例1 查看回收设置并启用回收站

```
admin@localhost> app-recycle settings get
回收设置
  启用回收： 关闭
  回收路径： ${apusic.home}/store/app_recycle
  保存天数： 30

admin@localhost> app-recycle settings set --enabled=true
回收设置已保存。
```

`settings set` 选项说明：

选项	说明
<code>--enabled=true false</code>	是否启用回收
<code>--path=</code>	回收路径
<code>--retention-days=</code>	回收保留天数，范围 1-365

示例2 列出回收站应用

卸载应用后，可在回收站中查看：

```
admin@localhost> app-recycle list
| 回收文件名          | 应用名称 | 上下文路径 | 卸载时间          | 大小
|
+-----+-----+-----+-----+
+-----+
| testapp_20260927143826 | testapp | /testapp   | 2026-09-27 14:38 |
4.50 KB |
第 1 / 1 页, 共 1 条
```

示例3 还原回收站应用

`restore` 需指定完整的回收文件名称（而非应用名称）：

```
admin@localhost> app-recycle restore testapp_20260927143826
应用还原成功。
```

还原时同样支持覆盖上下文路径、虚拟主机、密级等配置项，不指定时使用回收站中的原值。

示例4 彻底删除回收站应用

```
admin@localhost> app-recycle delete testapp_20260927143910 -f
```

回收站应用已彻底删除。

10.2 配置管理

10.2.1 虚拟主机管理 (host)

`host` 子命令用于管理虚拟主机，包括查看、创建、更新、删除虚拟主机及设置默认虚拟主机。

10.2.1.1 概要

```
Usage: host [-h] [COMMAND]
```

虚拟主机管理

`-h, --help` 显示帮助信息

Commands:

<code>list</code>	列出虚拟主机
<code>get</code>	查看虚拟主机详情
<code>add</code>	创建虚拟主机
<code>update</code>	更新虚拟主机
<code>delete</code>	删除虚拟主机
<code>set-default</code>	将虚拟主机设为默认

`host add` / `host update` 选项:

选项	说明
<code>--name=</code>	虚拟主机唯一业务标识 (add 必填)
<code>--docBase=</code>	文档根目录路径，为空时默认 webapps；同一文档根目录不允许被多个虚拟主机使用
<code>--alias=</code>	主机名称或IP地址，多个用英文逗号分隔
<code>--sso=true false</code>	单点登录模式配置，启用后当前虚拟主机启用 SSO 登录逻辑
<code>--sso-http-only=true false</code>	设置 SSO Cookie 的 HttpOnly 属性
<code>--access-ip-whitelist=</code>	允许访问的 IP，可用 * 通配，逗号分隔，留空代表不限制，示例：

	192.168.*,10.0.0.8
--access-ip-blacklist=	禁止访问的 IP，格式同上
--access-domain-whitelist=	允许访问的域名，可用 * 通配，逗号分隔，留空代表不限制，示例： *.example.com
--access-domain-blacklist=	禁止访问的域名，格式同上
--access-time=	仅配置时段内允许客户端访问；留空代表无限制；格式为时:分:秒-时:分:秒，逗号分隔多个时段
--extra-props= <header=value,...>	HTTP 响应头字段，如 X-Frame-Options=SAMEORIGIN

10.2.1.2 示例

示例1 列出虚拟主机

```
admin@localhost> host list
| 名称      | 别名 | 文档根目录 | 默认 |
+-----+-----+-----+-----+
| admin     |      | admin      | 是   |
| localhost |      | webapps    | 是   |
第 1 / 1 页, 共 2 条
```

示例2 查看虚拟主机详情

```
admin@localhost> host get localhost
名称: localhost
文档根目录: webapps
主机名称或IP地址，多个用英文逗号分隔:
域名:
单点登录模式配置，启用后当前虚拟主机启用 SSO 登录逻辑，关闭则本虚拟主机不启用单点登录: 禁用
设置SSO Cookie的HttpOnly属性: 否
允许访问的IP，可用 * 通配，逗号分隔，留空代表不限制，示例: 192.168.*,10.0.0.8:
禁止访问的IP，可用 * 通配，逗号分隔，留空代表不限制:
允许访问的域名，可用 * 通配，逗号分隔，留空代表不限制，示例: *.example.com:
禁止访问的域名，可用 * 通配，逗号分隔，留空代表不限制:
仅配置时段内允许客户端访问；留空代表无限制；格式为时:分:秒-时:分:秒，逗号分隔多个时
```

段：

HTTP 响应头字段，如 X-Frame-Options=SAMEORIGIN，响应头名只能包含字母、数字及 !#\$%&*+-.^_`|~:

示例3 创建虚拟主机

```
admin@localhost> host add --name=test-host --docBase=webapps-test --
alias=www.test.com --sso=false
```

虚拟主机创建成功，重启服务器后生效。

示例4 更新虚拟主机访问 IP 白名单

```
admin@localhost> host update test-host --access-ip-
whitelist=192.168.*,10.0.0.8
```

虚拟主机配置保存成功，重启服务器后生效。

示例5 将虚拟主机设为默认

```
admin@localhost> host set-default test-host
```

确定要将虚拟主机 test-host 设为默认吗？(y/N) y

虚拟主机 test-host 已设为默认。

示例6 删除虚拟主机

```
admin@localhost> host delete test-host -f
```

虚拟主机删除成功，重启服务器后生效。

10.2.2 线程池管理 (threadpool)

`threadpool` 子命令用于管理线程池，包括查看、创建、更新、删除线程池。

10.2.2.1 概要

```
Usage: threadpool [-h] [COMMAND]
```

线程池管理

-h, --help 显示帮助信息

Commands:

list 列出线程池

get 查看线程池详情

add	创建线程池
update	更新线程池
delete	删除线程池

threadpool add 选项 (update 时名称由参数指定，其余选项相同)：

选项	说明
--name=	线程池名称 (必填)
--min-threads=	最小线程数 (必填, 1-65535)
--max-threads=	最大线程数 (必填, 1-65535)
--queue-size=	队列大小 (必填, 1-65535)
--max-idle-time=	空闲超时时间 ms (必填, 1-86400000)
--priority=	线程优先级 (必填, 1-10)
--connection-auto-reclaim=	数据源连接自动回收 (必填: false/reclaim/detect)
--thread-step-reclaim=	阶梯式回收 (true 或 false, 默认 false)
--thread-step-size=	单次回收数 (thread-step-reclaim=true 时必填, 1-65535)
--thread-step-time=	回收时间间隔 ms (thread-step-reclaim=true 时必填, 1-86400000)

10.2.2.2 示例

示例1 列出线程池

```
admin@localhost> threadpool list
| 线程池名称          | 最大线程数 | 核心线程数 | 队列大小 | 空闲时间 | 优先级 |
+-----+-----+-----+-----+-----+-----+
-----+
| admin-thread-pool | 100        | 25         | 5000     | 60000 ms | 5      |
| http-thread-pool  | 200        | 25         | 5000     | 60000 ms | 5      |
第 1 / 1 页, 共 2 条
```

示例2 查看线程池详情

```
admin@localhost> threadpool get http-thread-pool
线程池名称: http-thread-pool
最小线程数: 25
最大线程数: 200
队列大小: 5000
空闲超时时间: 60000 ms
阶梯式回收: 关闭
数据源连接自动回收: 不启用
线程优先级: 5
```

示例3 创建线程池

```
admin@localhost> threadpool add --name=test-pool --min-threads=10 --
max-threads=100 --queue-size=1000 --max-idle-time=60000 --priority=5 -
-connection-auto-reclaim=false
线程池创建成功，重启服务器后生效。
```

示例4 更新线程池

```
admin@localhost> threadpool update test-pool --max-threads=200
线程池配置保存成功，重启服务器后生效。
```

示例5 删除线程池

```
admin@localhost> threadpool delete test-pool -f
线程池删除成功，重启服务器后生效。
```

10.2.3 会话管理器管理 (session-manager)

`session-manager` 子命令用于管理会话管理器，支持内存、文件、Redis、数据库四种存储类型。

10.2.3.1 概要

```
Usage: session-manager [-h] [COMMAND]
会话管理器管理
    -h, --help    显示帮助信息
Commands:
    list          列出会话管理器
```

```

get      查看会话管理器详情
add      创建会话管理器
update   更新会话管理器（名称与类型不可修改）
delete   删除会话管理器（被应用绑定的会话管理器不允许删除）

```

`session-manager add` 主要选项（`update` 时名称与类型不可修改，其余选项相同）：

选项	说明
<code>--name=</code>	会话管理器名称（必填，字母、数字、下划线、短横线，最长64字符）
<code>--type=</code>	存储类型（必填）：memory、file、redis、database
<code>--max-sessions=</code>	最大会话数，-1 表示不限制，默认 -1
<code>--persist-path=</code>	memory 类型：持久化文件路径，默认 SESSIONS.ser
<code>--save-on-restart=</code>	file/database 类型：重启时保存会话（默认 true）
<code>--max-idle-backup=</code>	file/database 类型：空闲备份阈值秒（-1 不启用，默认 -1）
<code>--min-idle-swap=</code>	file/database 类型：最小空闲换出阈值秒（-1 不启用，默认 -1）
<code>--max-idle-swap=</code>	file/database 类型：最大空闲换出阈值秒（-1 不启用，默认 -1）
<code>--persistence-dir=</code>	file 类型：持久化目录
<code>--data-source-name=</code>	database 类型：数据源 JNDI 名称（database 类型必填）
<code>--local-data-source=</code>	database 类型：使用应用本地数据源（默认 false）
<code>--redis-mode=</code>	redis 类型：standalone 或 sentinel，默认 standalone
<code>--redis-nodes=host:port[,...]</code>	redis 类型：节点地址，standalone 单个 host:port，sentinel 逗号分隔多个
<code>--redis-master=</code>	redis 类型：哨兵主节点名称（sentinel 模式必填）
<code>--redis-database= <0-15></code>	redis 类型：数据库索引，默认 0
<code>--redis-password=</code>	redis 类型：密码，最长256字符
<code>--session-persist-policies=</code>	redis 类型：持久化策略 SAVE_ON_CHANGE、ALWAYS_SAVE_AFTER_REQUEST，可逗号组合，默认空
<code>--redis-connect-timeout=</code>	redis 类型：连接超时 ms，默认 2000

<code>--redis-max-conn=</code>	redis 类型：最大连接数，默认 100
<code>--redis-min-idle=</code>	redis 类型：最小空闲连接数，默认 10
<code>--redis-max-idle=</code>	redis 类型：最大空闲连接数，留空使用默认值 8
<code>--redis-max-wait=</code>	redis 类型：获取连接最大等待 ms，默认 3000

另有会话属性过滤相关高级选项（`--session-attribute-name-filter`、`--session-attribute-value-class-name-filter`、`--warn-on-filter-failure`、`--notify-attribute-on-unchanged`、`--notify-binding-on-unchanged`），可通过 `session-manager add --help` 查看详细说明。

10.2.3.2 示例

示例1 创建内存类型会话管理器

```
admin@localhost> session-manager add --name=test-sm --type=memory
会话管理器创建成功。
```

示例2 列出会话管理器

```
admin@localhost> session-manager list
| 名称      | 类型 | 创建时间          | 更新时间          |
+-----+-----+-----+-----+
| test-sm | 内存 | 2026-09-27 14:39:37 | 2026-09-27 14:39:37 |
第 1 / 1 页, 共 1 条
```

示例3 查看会话管理器详情

```
admin@localhost> session-manager get test-sm
名称: test-sm
类型: 内存
最大会话数: 不限制
持久化文件路径: SESSIONS.ser
会话属性名过滤器:
会话属性值类名过滤器:
过滤失败时告警: 关闭
属性未变化仍派发事件: 开启
绑定未变化仍派发事件: 关闭
```

示例4 删除会话管理器

被应用绑定的会话管理器不允许删除，需先解绑应用：

```
admin@localhost> session-manager delete test-sm -f
会话管理器删除成功。
```

10.2.4 HTTP 通道管理 (httpchannel)

`httpchannel` 子命令用于管理 HTTP/HTTPS 通道（网络监听通道），包括查看、创建、更新、删除通道。

10.2.4.1 概要

```
Usage: httpchannel [-h] [COMMAND]
```

HTTP 通道管理

`-h, --help` 显示帮助信息

Commands:

```
list      列出 HTTP 通道
get       查看 HTTP 通道详情
add       创建 HTTP 通道
update    更新 HTTP 通道
delete    删除 HTTP 通道
```

`httpchannel add` 基本选项（`update` 时不指定名称选项，其余相同，默认保持当前配置）：

选项	说明
<code>--name=</code>	通道名称（必填）。长度 1-64 字符，仅支持字母、数字、下划线、短横线
<code>--protocol=</code>	IO 模式（必填）：NIO、NIO2、APR
<code>--host=</code>	监听地址，0.0.0.0 代表监听所有网卡，默认 0.0.0.0
<code>--port=</code>	监听端口（必填），1-65535，1024 以下需 root 权限
<code>--executor=</code>	业务线程池（必填），可用 <code>aascli threadpool list</code> 查询候选项
<code>--redirect-port=</code>	HTTP 请求需加密时自动跳转的 HTTPS 端口，1-65535
<code>--receive-buffer-size=</code>	接收缓冲区大小（TCP SO_RCVBUF），默认 8192 字节
<code>--send-buffer-size=</code>	发送缓冲区大小（TCP SO_SNDBUF），默认 8192 字节
<code>--tcp-no-delay=</code>	设置 TCP_NODELAY，默认 true

--connection-timeout=	连接超时（TCP SO_TIMEOUT），默认 20000 毫秒
--accept-count=	TCP backlog，默认 100
--max-connections=	最大并发 TCP 连接数，-1 表示无限制，默认 8192
--max-keep-alive-requests=	单个 TCP 连接处理的最大请求数，默认 100
--keep-alive-timeout=	TCP 连接空闲超时时间，默认 20 秒
--uri-encoding=	解析 URL 路径使用的字符编码，默认 UTF-8
--compression=	压缩模式 on/off/force，默认 on
--compressible-mime-type=	仅列表内资源类型才执行 GZIP 压缩，逗号分隔
--compression-min-size=	响应体小于该值时不执行压缩，默认 2048 字节
--max-post-size=	POST 请求体的数据上限，-1 表示无限制，默认 2097152 字节
--max-parameter-count=	单次请求的最大参数总数，默认 10000
--max-header-count=	请求头总字段上限，默认 100
--max-http-header-size=	请求头的总字节上限，默认 8192 字节
--response-headers=	自定义 HTTP 响应头，格式 Name=Value,Name2=Value2

SSL/HTTPS 相关选项：

选项	说明
--ssl-enabled=	启用 SSL 加密，仅 HTTPS 协议通道需配置，默认 false
--gm-enabled=	启用国密认证，开启后需要校验是否配置国密证书，默认 false
--tls-protocols=	TLS 版本，多个逗号分隔，默认 TLSv1.2,TLSv1.3
--client-auth=	客户端验证：false=禁用、want=可选、true=需要（双向认证），默认 false
--certificate-custom=	证书自定义，关闭时选择证书库，开启时输入自定义证书库信息
--keystore-name=	密钥库名称（证书自定义关闭时使用），可用 aascli cert list 查询
--keystore-entry=	密钥库证书条目，可用 aascli cert get 查询候选项
--truststore-name=	信任库名称（证书自定义关闭且客户端验证开启时使用）
--truststore-entry=	信任库证书条目

--keystore-path=	密钥库文件路径（证书自定义开启时使用）
--keystore-password=	密钥库访问密码（证书自定义开启时使用）
--certificate-format=	密钥库类型：JKS、PKCS12，默认 JKS
--certificate-alias=	证书别名
--key-password=	证书条目的保护密码
--truststore-path=	信任库文件路径（证书自定义开启且客户端验证开启时使用）
--truststore-password=	信任库访问密码
--truststore-type=	信任库类型：JKS、PKCS12，默认 JKS
--ssl-cipher-suites=	SSL 非国密时的密码套件，可用 aascli cert cipher-suites 查询候选项，逗号分隔
--http2-enabled=	开启 HTTP/2 需先启用 SSL，默认 false

另提供慢请求监控（--long-thread-*）、SDOS 防护（--sdos-*）及 HTTP/2 详细参数（--http2-*）等高级选项，可通过 `httpchannel add --help` 查看完整说明。

10.2.4.2 示例

示例1 列出 HTTP 通道

```
admin@localhost> httpchannel list
| 名称          | 地址 | 端口 | IO模式 |
+-----+-----+-----+-----+
| admin-http    |      | 6848 | NIO     |
| ams-http      |      | 6888 | NIO     |
第 1 / 1 页, 共 2 条
```

示例2 查看 HTTP 通道详情

```
admin@localhost> httpchannel get ams-http
名称: ams-http
IO模式: NIO
监听地址:
监听端口: 6888
重定向端口: 6443
线程池: http-thread-pool
```

接收缓冲区: 字节
发送缓冲区: 字节
TCP NoDelay: 开启
连接超时时间: 30000 毫秒
TCP等待队列大小: 100
协议处理器缓存大小:
优雅停机: 0 毫秒
最大连接数:
长连接最大请求数: 100
长连接超时时间: 15 秒
IP地址匹配虚拟主机: 关闭
慢请求监控: 关闭
黑名单检测: 关闭
URI编码: utf-8
请求体编码解析URI: 关闭
响应默认字符编码:
DNS反向查找: 关闭
压缩:
可压缩的MIME类型:
允许压缩的最小容量:
允许TRACE方法: 关闭
Powered By: 关闭
异步请求的超时时间: 30000 毫秒
禁用上传超时: 开启
连接上传超时: 毫秒
最大POST大小: 104857600
最大参数个数:
最大请求头个数: 100
最大请求头大小: 8192 字节
自定义响应头:
启用SSL: 关闭
启用HTTP/2: 关闭

示例3 创建 HTTP 通道

```
admin@localhost> httpchannel add --name=test-channel --protocol=NIO --port=6999 --executor=http-thread-pool
```

HTTP 通道创建成功，重启服务器后生效。

示例4 更新 HTTP 通道最大连接数

```
admin@localhost> httpchannel update test-channel --max-connections=4096
```

HTTP 通道配置保存成功，重启服务器后生效。

示例5 删除 HTTP 通道

```
admin@localhost> httpchannel delete test-channel -f
```

HTTP 通道删除成功，重启服务器后生效。

10.2.5 JVM 配置管理 (jvm)

`jvm` 子命令用于查看、更新、重置 JVM 启动配置（堆大小、GC 算法、调试选项等）。

10.2.5.1 概要

```
Usage:  jvm [-h] [COMMAND]
JVM 配置管理
    -h, --help    显示帮助信息
Commands:
    get          查看 JVM 配置
    update       更新 JVM 配置
    reset       恢复 JVM 默认配置
```

`jvm update` 选项：

选项	说明
<code>--xms=</code>	堆最小值 (-Xms)，单位 MB
<code>--xmx=</code>	堆最大值 (-Xmx)，单位 MB
<code>--metaspace-size=</code>	元空间初始值，单位 MB
<code>--max-metaspace-size=</code>	元空间最大值，单位 MB

--gc-type=	垃圾回收算法，可选：ParallelGC、G1GC、ConcMarkSweepGC、ParallelOldGC、ZGC、SerialGC
--debug=	开启调试（true 或 false）
--debug-options=	调试选项，例如：- agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000
--native-library-path=	本地库路径，多个路径用分号分隔
--jvm-options=	JVM 选项，多条以分号分隔

10.2.5.2 示例

示例1 查看 JVM 配置

```
admin@localhost> jvm get
JDK路径: D:\jdk\jdk-17.0.7
Java版本: 17.0.7
堆最小值 (-Xms): 1024
堆最大值 (-Xmx): 4096
元空间初始值:
元空间最大值:
垃圾回收算法:
开启调试: 关闭
调试选项:
本地库路径:
JVM选项列表:
```

示例2 更新 JVM 堆内存配置

```
admin@localhost> jvm update --xms=1024 --xmx=4096
JVM 配置保存成功，重启服务器后生效。
```

示例3 恢复 JVM 默认配置

```
admin@localhost> jvm reset
```

10.2.6 JMX 配置管理 (jmx)

`jmx` 子命令用于查看、更新、重置 JMX 远程监控配置。

10.2.6.1 概要

Usage: `jmx [-h] [COMMAND]`

JMX 配置管理

`-h, --help` 显示帮助信息

Commands:

`get` 查看 JMX 配置
`update` 更新 JMX 配置
`reset` 恢复 JMX 默认配置

`jmx update` 主要选项:

选项	说明
<code>--enabled=</code>	启用 JMX (true 或 false)
<code>--host=</code>	监听地址, 留空默认监听本地 127.0.0.1
<code>--port=</code>	监听端口, 1-65535, 留空默认 6868
<code>--authenticate=</code>	启用认证 (true 或 false)
<code>--password-file=</code>	密码文件路径, 如 <code>conf/jmxremote.password</code>
<code>--access-file=</code>	访问控制文件路径, 如 <code>conf/jmxremote.access</code>
<code>--ssl-enabled=</code>	启用 SSL (true 或 false)
<code>--client-auth=</code>	客户端校验 (true 或 false)
<code>--certificate-custom=</code>	证书自定义 (true 或 false)
<code>--key-store-file=</code>	密钥库路径, 如 <code>conf/jmx.keystore</code>
<code>--key-password=</code>	密钥密码
<code>--key-store-password=</code>	密钥库密码
<code>--key-store-type=</code>	密钥库类型, JKS 或 PKCS12
<code>--trust-store-file=</code>	信任库路径, 如 <code>conf/jmx.truststore</code>
<code>--trust-store-password=</code>	信任库密码
<code>--trust-store-type=</code>	信任库类型, JKS 或 PKCS12

<code>--enabled-cipher-suites=</code>	密码套件，多个以英文逗号分隔，可用 <code>aascli cert cipher-suites</code> 查询候选项
<code>--key-store-name=</code>	密钥库名称（证书自定义关闭时使用），可用 <code>aascli cert list</code> 查询候选项
<code>--trust-store-name=</code>	信任库名称（证书自定义关闭时使用）
<code>--key-store-entry=</code>	密钥库条目别名，可用 <code>aascli cert get</code> 查询候选项

10.2.6.2 示例

示例1 查看 JMX 配置

```
admin@localhost> jmx get
启用 JMX：关闭
```

示例2 启用 JMX

```
admin@localhost> jmx update --enabled=true
JMX 配置保存成功，重启服务器后生效。
```

示例3 恢复 JMX 默认配置

```
admin@localhost> jmx reset
```

10.2.7 证书管理 (cert)

`cert` 子命令用于管理证书库（密钥库/信任库），包括证书库的创建、导入、下载、删除及证书条目管理。

10.2.7.1 概要

```
Usage: cert [-h] [COMMAND]
```

证书管理

`-h, --help` 显示帮助信息

Commands:

<code>list</code>	列出证书库
<code>get</code>	查看证书库详情及条目列表
<code>add</code>	新建证书库
<code>update</code>	更新证书库
<code>delete</code>	删除证书库
<code>delete-entry</code>	删除证书条目
<code>import</code>	从证书库文件导入条目

```
import-pem    导入 PEM 格式证书或私钥
download      下载证书库文件
cipher-suites 查询支持的密码套件
```

`cert add` 选项:

选项	说明
<code>--name=</code>	证书库名称（必填），最长 64 字符，仅支持字母、数字、下划线、短横线、点号，全局唯一
<code>--type=</code>	证书库类型，JKS 或 PKCS12，默认 JKS
<code>--store-type=</code>	库类型，keystore（密钥库）或 truststore（信任库），默认 keystore
<code>--password=</code>	证书库密码，8-32 字符，省略时交互式输入

`cert import` 选项:

选项	说明
	目标证书库名称
<code>--file=</code>	证书库文件路径，支持 .jks/.p12/.pfx/.pkcs12，大小不超过 10MB
<code>--source-password=</code>	源证书库密码，省略时交互式输入
<code>--aliases=</code>	要导入的别名，多个以英文逗号分隔，省略则导入全部
<code>--target-alias=</code>	导入后在目标证书库中的别名，留空则使用源别名
<code>--entry-password=</code>	条目保护密码，仅 PKCS12 类型的密钥库可用，留空则使用目标证书库密码

`cert import-pem` 选项:

选项	说明
	目标证书库名称
<code>--file=</code>	PEM 文件路径，内容为证书或私钥（含 BEGIN/END 标记）
<code>--alias=</code>	导入后在目标证书库中的别名
<code>--entry-password=</code>	条目保护密码，仅 PKCS12 类型的密钥库可用，留空则使用目标证书库密码

10.2.7.2 示例

示例1 新建证书库

```
admin@localhost> cert add --name=test-store --password=Test12345
```

证书库创建成功, ID: 2

示例2 列出证书库

```
admin@localhost> cert list
```

名称	用途	类型	路径
test-store	keystore	JKS	D:\AAMS-V10.0.8\store\ssl\test-store.jks

第 1 / 1 页, 共 1 条

示例3 从证书库文件导入条目

```
admin@localhost> cert import test-store --file=D:/aas-source.jks --source-password=Source123
```

证书条目导入成功。

示例4 查看证书库详情及条目列表

```
admin@localhost> cert get test-store
```

名称: test-store
 路径: D:\AAMS-V10.0.8\store\ssl\test-store.jks
 用途: 密钥库
 类型: JKS

证书条目:

别名	条目类型	算法	有效期起	有效期止
demo-cert	私钥	SHA256WITHRSA	2026-09-27	2027-09-27

条目详情:
 demo-cert:

```
Subject: CN=demo,OU=test,O=test,L=bj,ST=bj,C=CN
Issuer: CN=demo,OU=test,O=test,L=bj,ST=bj,C=CN
```

示例5 下载证书库文件

```
admin@localhost> cert download test-store --file=D:/aas-test-store.jks
证书库文件已下载到: D:\aas-test-store.jks
```

--file 省略时保存为当前目录下的 <证书库名称>.<jks|p12>。

示例6 删除证书条目

```
admin@localhost> cert delete-entry test-store demo-cert -f
证书条目删除成功。
```

示例7 删除证书库

```
admin@localhost> cert delete test-store -f
证书库删除成功。
```

示例8 查询支持的密码套件

配置 HTTP 通道或 JMX 的 SSL 密码套件时，可先查询当前 JDK 支持的密码套件：

```
admin@localhost> cert cipher-suites
TLS_AES_256_GCM_SHA384
TLS_AES_128_GCM_SHA256
TLS_CHACHA20_POLY1305_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
... (省略) ...
TLS_EMPTY_RENEGOTIATION_INFO_SCSV
共 23 个密码套件。
```

10.3 资源管理

10.3.1 数据源管理 (datasource)

`datasource` 子命令用于管理 JDBC 数据源，包括数据源的创建、查看、更新、启停、连通性测试与删除。

10.3.1.1 概要

Usage: datasource [-h] [COMMAND]

数据源管理

-h, --help 显示帮助信息

Commands:

list 列出数据源
 get 查看数据源详情
 add 创建数据源
 update 更新数据源
 delete 删除数据源
 toggle 启用或禁用数据源
 ping 测试数据源连通性

datasource add 主要选项（ update 时名称由参数指定，其余选项相同）：

选项	说明
--name=	数据源名称（必填），支持字母、数字、点号、斜杠、中横线、下横线，最长128字符
--db-type=	数据库驱动供应商（必填），可选：Oracle、MySQL8、MySQL5、PostgreSQL、DM（达梦）、Kingbase（人大金仓）、GaussDB（高斯）、HighGo（瀚高）、Vastbase（海量）、Other（其他）
--resource-type=	资源类型（必填），可选：javax.sql.DataSource、javax.sql.XADataSource、javax.sql.ConnectionPoolDataSource（默认 javax.sql.DataSource）
--driver-class=	数据库驱动类名（必填），最长128字符
--driver-path=	数据库驱动 jar 在服务器上的绝对路径，如 /opt/drivers/mysql-connector.jar
--status=	创建后是否立即启用（默认 true）
--class-level=	密级标识（2=秘密、3=机密、4=绝密，默认 2）
--host=	数据库主机名或 IP，最长255字符
--port=	数据库端口（必填，1-65535）
--database-name=	数据库名或 Schema，最长64字符
--username=	连接用户名（必填），最长255字符
--password=	连接密码（必填），最长255字符
--url=	JDBC URL（必填），必须以 jdbc: 开头，最长500字符

--custom-props=	自定义连接属性，格式：key1=value1,key2=value2
--initial-connections=	初始连接数（默认 10，范围 0-10000）
--max-connections=	最大连接数（默认 20，范围 1-10000）
--min-idle=	最小空闲连接数（默认 4，范围 0-10000）
--max-idle=	最大空闲连接数（默认 10，范围 1-10000）
--max-wait-ms=	最大等待时间 ms，-1 表示无限等待（默认 15000）
--idle-timeout-ms=	空闲超时时间 ms（默认 60000，范围 1-86400000）
--idle-check-period-ms=	空闲检测周期 ms，0 表示禁用检测（默认 60000）
--max-lifetime-ms=	最大存活时间 ms，0 表示不限制（默认 900000）
--validation-query=	验证 SQL，建议使用只读 SELECT 语句（最长255字符，禁止危险操作关键字）
--test-on-borrow=	获取连接时校验（默认 true）
--test-on-return=	归还连接时校验（默认 false）
--test-while-idle=	空闲状态时校验（默认 false）
--test-on-connect=	建立连接时校验（默认 false）
--validation-interval-ms=	连接校验间隔 ms（默认 30000）
--connection-auto-reclaim=	连接自动回收（默认 false）
--leak-recover=	泄漏连接回收（默认 false）
--leak-detection-threshold-seconds=	泄漏判定超时秒（默认 30，范围 1-1800）
--auto-commit=	自动提交事务（默认 true）
--log-abandoned=	记录泄漏日志（默认 false）
--sql-trace-enabled=	慢语句跟踪（默认 false）
--slow-sql-ms=	慢 SQL 阈值 ms（默认 2000，范围 1-300000）

```
--failed-sql-alert-
enabled=
```

记录失败 SQL (默认 false)

另有 `--max-check-connections`、`--remove-abandoned-on-maintenance`、`--abandon-when-percentage-full`、`--ha-probe-enabled` 等选项, 可通过 `datasource add --help` 查看完整说明。

10.3.1.2 示例

示例1 创建数据源

```
admin@localhost> datasource add --name=test-ds --db-type=Other --
resource-type=javax.sql.DataSource --driver-class=org.h2.Driver --
driver-path=D:/aas-h2.jar --port=9092 --username=[YOURUSER] --
password=[YOURPASSWORD] --url=jdbc:h2:mem:testdb
数据源创建成功, ID: 1
```

示例2 列出数据源

```
admin@localhost> datasource list
| 数据源名 | 资源类型 | 密级 | 驱动类名 | 状态 | 连通性 |
+-----+-----+-----+-----+-----+-----+
| test-ds | javax.sql.DataSource | 秘密 | org.h2.Driver | 已启用 | 已连通 |
第 1 / 1 页, 共 1 条
```

示例3 查看数据源详情

```
admin@localhost> datasource get test-ds
【基本配置】
ID: 1
名称: test-ds
数据库类型: 其他
资源类型: javax.sql.DataSource
驱动类名: org.h2.Driver
驱动路径: D:/aas-h2.jar
状态: 开启
密级: 秘密
```

主机名：

端口：9092

数据库名：

用户名：sa

密码：*****

URL：jdbc:h2:mem:testdb

自定义属性：

【连接池配置】

初始连接数：10

最大连接数：20

最小空闲连接数：4

最大空闲连接数：10

最大等待时间：15000 毫秒

空闲超时时间：60000 毫秒

空闲检测周期：60000 毫秒

最大存活时间：900000 毫秒

单次检测连接数：5

【连接验证配置】

验证 SQL：

校验超时时间：3 秒

获取连接时校验：开启

归还连接时校验：关闭

空闲状态时校验：关闭

建立连接时校验：关闭

连接校验间隔：30000 毫秒

【高级配置】

连接自动回收：关闭

泄漏连接回收：关闭

维护时回收泄漏连接：关闭

泄漏判定超时：30 秒

连接满阈值比例：10 %

自动提交事务：开启

记录泄漏日志：关闭

【监控与诊断】

慢语句跟踪：关闭

慢 SQL 阈值：2000 毫秒

记录失败 SQL：关闭

连接唤醒池：关闭

示例4 测试数据源连通性

```
admin@localhost> datasource ping test-ds
```

连接成功：连接成功，耗时 1ms

示例5 禁用/启用数据源

```
admin@localhost> datasource toggle test-ds --status=false
```

数据源已关闭。

```
admin@localhost> datasource toggle test-ds --status=true
```

数据源已开启。

示例6 删除数据源

处于启用状态的数据源不允许直接删除，需先禁用：

```
admin@localhost> datasource delete test-ds -f
```

数据源操作失败：数据源已启用，请先禁用后再删除

```
admin@localhost> datasource toggle test-ds --status=false
```

数据源已关闭。

```
admin@localhost> datasource delete test-ds -f
```

数据源删除成功。

10.4 监控管理

10.4.1 监控配置管理 (monitor-config)

`monitor-config` 子命令用于查看、更新、重置服务器监控采集配置。

10.4.1.1 概要

Usage: monitor-config [-h] [COMMAND]

监控配置管理

-h, --help 显示帮助信息

Commands:

get 查看监控配置
update 更新监控配置
reset 恢复监控配置默认值

monitor-config update 主要选项:

选项	说明
--monitor-enabled=	监视服务总开关 (true/false)
--mbean-monitor-enabled=	MBean 监控开关 (true/false)
--collect-interval-seconds=	监控采集周期 (秒), 范围 10~3600, 默认 60
--jvm-monitor-enabled=	JVM 监控开关 (true/false)
--thread-pool-monitor-enabled=	线程池监控开关 (true/false)
--datasource-monitor-enabled=	数据源监控开关 (true/false)
--http-monitor-enabled=	HTTP 通道监控开关 (true/false)
--session-monitor-enabled=	应用会话监控开关 (true/false)
--system-resource-monitor-enabled=	系统资源监控开关 (true/false)
--fullgc-monitor-enabled=	FullGC 监控开关 (true/false)
--cpu-monitor-enabled=	CPU 监控开关 (true/false)
--memory-monitor-enabled=	内存监控开关 (true/false)
--max-data-points=	监控数据总量, 范围 1~2147483647
--replay-sample-interval=	回放采样间隔 (秒), 范围 1~86400

10.4.1.2 示例

示例1 查看监控配置

```
admin@localhost> monitor-config get
```

监控配置：

监视服务：关闭

监视 MBean：关闭

监控采集周期（秒）：60

JVM 监控：关闭

线程池监控：关闭

数据源监控：关闭

HTTP 通道监控：关闭

应用会话监控：关闭

系统资源监控：关闭

FullGC 监控：关闭

CPU 监控：关闭

内存监控：关闭

监控数据总量：11000

回放采样间隔（秒）：30

示例2 开启监控并调整采集周期

```
admin@localhost> monitor-config update --monitor-enabled=true --
collect-interval-seconds=120
```

监控配置已更新。

示例3 恢复监控配置默认值

```
admin@localhost> monitor-config reset
```

监控配置已恢复默认值。

10.4.2 告警配置管理 (alert-config)

`alert-config` 子命令用于查看、更新、重置告警配置，包括告警开关、通知方式（邮件/SNMP/告警平台）及各项阈值。

10.4.2.1 概要

```
Usage: alert-config [-h] [COMMAND]
```

告警配置管理

`-h, --help` 显示帮助信息

Commands:

```

get      查看告警配置
update   更新告警配置
reset    恢复告警配置默认值

```

`alert-config update` 主要选项:

分类	选项	说明
告警开关	--enabled=	启用告警 (true/false)
	--interval-minutes=	告警间隔 (分钟), 范围 1~527040
邮件通知	--email-enabled=	启用邮件通知 (true/false)
	--email-recipients=	接收邮件, 多个邮箱用英文分号隔开
	--email-smtp-host=	发送邮件服务器地址, 如 smtp.example.com
	--email-smtp-ssl=	启用邮件 SSL (true/false)
	--email-smtp-port=	发送邮件服务器端口, 范围 1~65535
	--email-smtp-user=	发送邮箱地址
	--email-smtp-password=	发送邮箱密码, 留空表示保留原值
SNMP 通知	--snmp-enabled=	启用 SNMP 通知 (true/false)
	--snmp-version=	SNMP 版本: v2c 或 v3
	--snmp-community=	SNMP 团体名 (v2c)
	--snmp-address=	SNMP 通知目标地址, 如 172.24.4.108:162
	--snmp-security=	SNMP 安全级别 (v3): 无、认证不加密、认证加密
	--snmp-username=	SNMP 用户名 (v3)
	--snmp-auth-method=	SNMP 认证方式 (v3): MD5 或 SHA
	--snmp-auth-password=	SNMP 认证密码 (v3), 留空表示保留原值
	--snmp-cipher=	SNMP 加密算法 (v3): DES、3DES、AES128、AES192、AES256

	--snmp-priv-password=	SNMP 加密密码 (v3) , 留空表示保留原值
告警平台	--platform-enabled=	启用告警平台通知 (true/false)
	--platform-level=	告警级别, 如 警告、严重
	--platform-url=	告警平台地址, 须为 http:// 或 https:// 开头
阈值	--cpu-threshold= / --cpu-consecutive=	CPU 使用率告警阈值 / 连续发生次数 (不小于 1)
	--memory-threshold= / --memory-consecutive=	内存使用率告警阈值 / 连续发生次数
	--fullgc-count-threshold= / --fullgc-count-consecutive=	FullGC 次数告警阈值 / 连续发生次数 (固定为 1)
	--fullgc-duration-threshold= / --fullgc-duration-consecutive=	FullGC 时间告警阈值 (毫秒) / 连续发生次数
	--http-timeout-threshold= / --http-timeout-consecutive=	HTTP 超时请求告警阈值 / 连续发生次数
	--http-waiting-threshold= / --http-waiting-consecutive=	HTTP 等待任务数告警阈值 / 连续发生次数
	--http-error-threshold= / --http-error-consecutive=	HTTP 错误数告警阈值 / 连续发生次数
	--jdbc-waiting-threshold= / --jdbc-waiting-consecutive=	JDBC 等待请求数告警阈值 / 连续发生次数
	--jdbc-leak-threshold= / --jdbc-leak-consecutive=	JDBC 泄露连接数告警阈值 / 连续发生次数

10.4.2.2 示例

示例1 查看告警配置

```
admin@localhost> alert-config get
```

告警配置:

告警开关: 关闭

告警间隔 (分钟) : 30

邮件通知

邮件通知开关: 关闭

接收邮件:

发送邮件服务器:

邮件 SSL: 开启

发送邮件服务器端口:

发送邮箱:

发送邮箱密码:

SNMP 通知

SNMP 通知开关: 关闭

SNMP 版本: v2c

SNMP 团体名: public

SNMP 地址:

SNMP 安全级别: 无

SNMP 用户名:

SNMP 认证方式: SHA

SNMP 认证密码:

SNMP 加密算法: AES128

SNMP 加密密码:

SNMP engineID: 0x8000137001AC1806A0FB9B64B9

告警平台通知

告警平台通知开关: 关闭

告警级别: 警告

告警平台地址:

告警指标

指标	阈值	单位	连续次数
CPU使用率	80.0	%	3
内存使用率	80.0	%	3
FullGC次数	5.0	次/分钟	1
FullGC时间	1000.0	毫秒	2
HTTP超时请求	10.0	-	2
HTTP等待任务数	10.0	-	3
HTTP错误数	5.0	-	2

JDBC等待请求数	10.0	-	3	
JDBC泄露连接数	1.0	-	2	

示例2 开启告警并配置邮件通知

```
admin@localhost> alert-config update --enabled=true --email-enabled=true --email-recipients=ops@example.com --email-smtp-host=smtp.example.com --email-smtp-port=465 --email-smtp-user=alerts@example.com --email-smtp-password=*****
```

告警配置已更新。

示例3 调整 CPU 告警阈值

```
admin@localhost> alert-config update --cpu-threshold=90 --cpu-consecutive=5
```

告警配置已更新。

示例4 恢复告警配置默认值

```
admin@localhost> alert-config reset
```

告警配置已恢复默认值。

10.4.3 SNMP 代理配置管理 (snmp-config)

`snmp-config` 子命令用于查看、更新、重置 AAS 内置 SNMP 代理配置，供上层网管系统通过 SNMP 方式采集监控数据。

10.4.3.1 概要

```
Usage: snmp-config [-h] [COMMAND]
```

SNMP 代理配置管理

`-h, --help` 显示帮助信息

Commands:

<code>get</code>	查看	SNMP 代理配置
<code>update</code>	更新	SNMP 代理配置
<code>reset</code>	恢复	SNMP 配置默认值

`snmp-config update` 选项:

选项	说明
--enabled=	启用 SNMP 服务 (true/false)
--host=	绑定地址, 如 0.0.0.0、192.168.1.1, 最长 128 个字符
--port=	绑定端口, 范围 1~65535, 修改时服务端会做端口占用探测
--protocol=	传输协议: udp 或 tcp
--version=	SNMP 版本: v2c 或 v3
--community=	SNMP 团体名 (v2c)
--security=	SNMP 安全级别 (v3) : 无、认证不加密、认证加密
--username=	SNMP 用户名 (v3)
--auth-method=	SNMP 认证方式 (v3) : MD5 或 SHA
--auth-password=	SNMP 认证密码 (v3) , 留空表示保留原值
--cipher=	SNMP 加密算法 (v3) : DES、3DES、AES128、AES192、AES256
--priv-password=	SNMP 加密密码 (v3) , 留空表示保留原值

10.4.3.2 示例

示例1 查看 SNMP 代理配置

```
admin@localhost> snmp-config get
SNMP 代理配置:
SNMP 服务开关: 关闭
绑定地址: 0.0.0.0
绑定端口: 161
传输协议: udp
SNMP 版本: v2c
SNMP 团体名: public
```

示例2 启用 SNMP 代理 (v2c)

```
admin@localhost> snmp-config update --enabled=true --host=0.0.0.0 --
port=161 --protocol=udp --version=v2c --community=public
SNMP 配置已更新。
```

示例3 恢复 SNMP 配置默认值

```
admin@localhost> snmp-config reset
SNMP 配置已恢复默认值。
```

10.4.4 一键巡检管理 (inspection)

`inspection` 子命令用于一键巡检服务器的运行状态，生成巡检报告，并可查看、下载、删除报告及查看巡检系统快照。

10.4.4.1 概要

```
Usage: inspection [-h] [COMMAND]
一键巡检管理
    -h, --help    显示帮助信息
Commands:
    list          列出巡检报告
    get           查看巡检报告详情
    run           发起一键巡检
    status        查询巡检报告执行状态
    summary       查看巡检报告摘要
    download      下载巡检报告压缩包（仅巡检成功的报告）
    delete       删除巡检报告
    rerun         重新巡检（仅已完成或异常的报告）
    snapshot      查看或下载巡检系统快照（jstack、lsof、netstat、top, lsof 仅 Linux）
    has-running   查询是否存在正在执行的巡检任务
```

`inspection list` 过滤选项:

选项	说明
--name=	按报告名称过滤
--status=	按状态过滤：RUNNING、COMPLETED、TIMEOUT、ABNORMAL（异常）
--start-time=	开始时间，格式：yyyy-MM-dd HH:mm:ss
--end-time=	结束时间，格式：yyyy-MM-dd HH:mm:ss

`inspection run` 选项:

选项	说明
--timeout-seconds=	巡检超时时间（秒），范围 1~172800，默认 600

`inspection snapshot` 参数：

参数	说明
	报告 ID，可用 <code>aascli inspection list</code> 查询
	快照类型：jstack、lsof、netstat、top（lsof 仅 Linux）
--file=	保存路径，缺省时直接输出快照内容

10.4.4.2 示例

示例1 查询是否存在正在执行的巡检任务

```
admin@localhost> inspection has-running
不存在正在执行的巡检任务
```

示例2 发起一键巡检

巡检在后台异步执行，提交后可通过报告 ID 查询进度：

```
admin@localhost> inspection run
巡检任务已提交（ID: 1, 名称: InspectionReport_20260927144009），正在后台执行
可使用 aascli inspection status 1 查询执行状态
```

示例3 查询巡检执行状态

```
admin@localhost> inspection status 1
正在巡检

admin@localhost> inspection status 1
巡检成功
```

示例4 列出巡检报告

```
admin@localhost> inspection list
| ID | 巡检结果 | 状态 | 超时时间（秒） | 日期 |
|
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| 1 | InspectionReport_20260927144009 | 巡检成功 | 600 |
2026-09-27 14:40:09 |
第 1 / 1 页, 共 1 条
```

示例5 查看巡检报告摘要

```
admin@localhost> inspection summary 1
CPU 负载 80% 以上的实例数: 0
内存使用率 80% 以上的实例数: 0
堆使用率 80% 以上的实例数: 0
硬盘使用率 80% 以上的实例数: 1
```

示例6 下载巡检报告压缩包

```
admin@localhost> inspection download 1 --file=D:/aas-inspection-
report.zip
下载成功: D:\aas-inspection-report.zip
```

示例7 重新巡检

仅已完成或异常的报告可重新巡检:

```
admin@localhost> inspection rerun 1
```

示例8 查看巡检系统快照

```
admin@localhost> inspection snapshot 1 jstack
```

示例9 删除巡检报告

```
admin@localhost> inspection delete 1 -f
巡检报告已删除。
```

10.4.5 监控快照管理 (snapshot)

`snapshot` 子命令用于管理监控快照，支持手动创建、下载、删除快照，以及配置快照自动生成策略。

10.4.5.1 概要

Usage: snapshot [-h] [COMMAND]

监控快照管理（手动创建、下载、删除、自动配置）

-h, --help 显示帮助信息

Commands:

list 列出监控快照
 get 按名称精确查询快照详情
 create 手动创建监控快照（异步生成）
 delete 删除监控快照
 download 下载监控快照压缩包（仅生成完成的快照）
 config 快照自动生成配置

snapshot list 过滤选项:

选项	说明
--name=	按快照名称过滤
--status=	按状态过滤: pending、running、done、failed
--start-time=	开始时间, 格式: yyyy-MM-dd HH:mm:ss
--end-time=	结束时间, 格式: yyyy-MM-dd HH:mm:ss

snapshot create 选项:

选项	说明
--contents=<a,b,...>	快照内容（必填），逗号分隔: accessLog、gcLog、systemConfig、systemLog、hostMonitor、instanceMetrics、coreDump、threadDump、heapDump
--thread-stack-count=	线程栈抓取次数, 范围 1~10（含 threadDump 时建议指定）
--heapdump-mode=	Heapdump 模式: all 或 live（含 heapDump 时建议指定）

snapshot config 子命令:

Usage: snapshot config [-h] [COMMAND]

快照自动生成配置

-h, --help 显示帮助信息

Commands:

```

get      查看快照自动生成配置
update   更新快照自动生成配置
reset    恢复快照自动生成配置默认值

```

snapshot config update 主要选项：

分类	选项	说明
基本设置	--auto-enabled=	启用自动快照 (true/false)
	--max-snapshots=	最多保存快照数, 范围 1~1000
	--interval-minutes=	生成间隔 (分钟), 范围 1~1440
快照内容	--content-access-log=	采集访问日志 (true/false)
	--content-gc-log=	采集 GC 日志 (true/false)
	--content-system-config=	采集系统配置 (true/false)
	--content-system-log=	采集系统日志 (true/false)
	--content-host-monitor=	采集主机监控数据 (true/false)
	--content-instance-metrics=	采集实例监视量 (true/false)
	--content-core-dump=	采集系统 Coredump (true/false)
	--content-thread-dump=	采集线程栈 (true/false)
	--thread-stack-count=	线程栈抓取次数, 范围 1~10
	--content-heap-dump=	采集 Heapdump (true/false)
	--heapdump-mode=	Heapdump 模式: all 或 live
触发规则	--rule-cpu-threshold= / --rule-cpu-occurrences=	CPU 使用率阈值 (%) / 连续发生次数
	--rule-memory-threshold= / --rule-memory-occurrences=	内存使用率阈值 (%) / 连续发生次数
	--rule-fullgc-count-threshold= / --rule-fullgc-count-occurrences=	FullGC 次数阈值 / 连续发生次数

	--rule-fullgc-time-threshold= / --rule-fullgc-time-occurrences=	FullGC 时间阈值（毫秒） / 连续发生次数
	--rule-http-timeout-threshold= / --rule-http-timeout-occurrences=	HTTP 慢请求数阈值 / 连续发生次数
	--rule-http-wait-threshold= / --rule-http-wait-occurrences=	HTTP 等待任务数阈值 / 连续发生次数
	--rule-http-error-threshold= / --rule-http-error-occurrences=	HTTP 错误数阈值 / 连续发生次数
	--rule-jdbc-wait-threshold= / --rule-jdbc-wait-occurrences=	JDBC 等待请求数阈值 / 连续发生次数
	--rule-jdbc-leak-threshold= / --rule-jdbc-leak-occurrences=	JDBC 疑似泄露连接数阈值 / 连续发生次数

10.4.5.2 示例

示例1 手动创建监控快照

快照为异步生成，提交后可通过列表查看进度：

```
admin@localhost> snapshot create --contents=systemLog,threadDump --
thread-stack-count=1
快照任务已提交，正在后台生成。
可使用 aascli snapshot list 查看生成进度。
```

示例2 列出监控快照

```
admin@localhost> snapshot list
| ID | 名称 | 类型 | 状态 | 内容 |
| 创建时间 | | | | |
+----+-----+-----+-----+-----+
-----+-----+
| 1 | snapshot-20260927-144158-manual | 手动 | 已完成 |
systemLog,threadDump | 2026-09-27 14:41:58 |
第 1 / 1 页, 共 1 条
```

示例3 按名称精确查询快照详情

```
admin@localhost> snapshot get snapshot-20260927-144158-manual
```

示例4 下载监控快照压缩包

```
admin@localhost> snapshot download 1 --file=D:/aas-snapshot.zip  
快照下载成功: D:\aas-snapshot.zip
```

示例5 删除监控快照

```
admin@localhost> snapshot delete 1 -f  
快照已删除。
```

示例6 查看快照自动生成配置

```
admin@localhost> snapshot config get  
快照自动生成配置:
```

自动快照

自动快照开关: 关闭

最多保存快照数: 10

生成间隔 (分钟) : 5

快照内容

访问日志: 开启

GC 日志: 关闭

系统配置: 关闭

系统日志: 开启

主机监控数据: 关闭

实例监视量: 关闭

系统 Coredump: 关闭

线程栈: 开启

线程栈抓取次数: 3

Heapdump: 开启

Heapdump 模式: live

触发规则

CPU 使用率阈值 (%) : 80.0

```

CPU 连续发生次数: 3
内存使用率阈值 (%) : 80.0
内存连续发生次数: 3
FullGC 次数阈值: 5
FullGC 次数连续发生次数: 1
FullGC 时间阈值 (毫秒) : 1000
FullGC 时间连续发生次数: 2
HTTP 慢请求数阈值: 1
HTTP 慢请求数连续发生次数: 2
HTTP 等待任务数阈值: 1
HTTP 等待任务数连续发生次数: 3
HTTP 错误数阈值: 3
HTTP 错误数连续发生次数: 2
JDBC 等待请求数阈值: 1
JDBC 等待请求数连续发生次数: 3
JDBC 疑似泄露连接数阈值: 1
JDBC 疑似泄露连接数连续发生次数: 2

```

示例7 更新快照自动生成配置

```

admin@localhost> snapshot config update --auto-enabled=true --
interval-minutes=30 --max-snapshots=20

```

示例8 恢复快照自动生成配置默认值

```

admin@localhost> snapshot config reset

```

10.5 日志服务

10.5.1 系统日志配置 (log-config system)

`log-config system` 子命令用于查看、更新服务器系统日志配置。

10.5.1.1 概要

```
Usage: log-config system [-h] [COMMAND]
```

系统日志配置

`-h, --help` 显示帮助信息

Commands:

get 查看系统日志配置
update 更新系统日志配置

log-config system update 主要选项:

分类	选项	说明
基本配置	--global-level=	全局日志级别，默认 INFO，可选：SEVERE/WARNING/INFO/CONFIG/FINE/FINER/FINEST/ALL
	--class-loggers=	自定义类级别配置，为指定包或类设置独立日志级别，多个以英文逗号分隔，格式：package.Class=LEVEL
	--console-enabled=	是否在控制台输出日志信息，默认 true
	--file-enabled=	是否将日志信息写入文件，默认 true
文件配置	--file-prefix=	日志文件名前缀标识，默认 apusic.，最长 100 字符
	--file-suffix=	日志文件名后缀，默认 .log，最长 100 字符
	--encoding=	日志文件字符编码，默认 UTF-8，可选：UTF-8/GBK
	--file-timestamp-format=	日志文件名中的日期格式，默认 yyyy-MM-dd
格式配置	--console-pattern=	控制台日志输出格式模板，如 "%d{yyyy-MM-dd HH:mm:ss.SSS} %level [%t] %logger %msg%n"
	--file-pattern=	文件日志输出格式模板，格式同上
轮转配置	--retention-count=	按天保留日志文件，超过保留天数的日志将自动删除，默认 30，0 或负数表示不限制
	--max-size=	单个日志文件达到该大小时自动轮转，单位 MB，默认 100，0 或负数表示不限制
	--rotate-by-date=	是否每天生成一个新的日志文件，默认 true
压缩配置	--compress-enabled=	是否开启日志自动压缩归档功能，默认 false
	--compress-time=	每天执行日志压缩的时间点，默认 02:00，格式 HH:mm

`--compress-dir=`

压缩文件的存储路径，默认 `${apusic.base}/logs/archive`，支持变量如 `${apusic.base}`

10.5.1.2 示例

示例1 查看系统日志配置

```
admin@localhost> log-config system get
```

【基本配置】

全局日志级别：FINE

自定义类级别配置：

控制台日志输出：开启

日志文件输出：开启

【文件配置】

日志文件前缀：apusic.

日志文件后缀：.log

日志文件编码：UTF-8

日志文件名日期格式：yyyy-MM-dd

【格式配置】

控制台日志格式：`%d{yyyy-MM-dd HH:mm:ss.SSS} %level [%t] %logger %msg%n`

文件日志格式：`%d{yyyy-MM-dd HH:mm:ss.SSS} %level [%t] %logger %msg%n`

【轮转配置】

保留天数：

按大小轮转：0 MB

按天轮转：开启

【压缩配置】

日志压缩：关闭

示例2 调整全局日志级别

```
admin@localhost> log-config system update --global-level=INFO
```

日志配置更新成功。

示例3 为指定包设置独立日志级别

```
admin@localhost> log-config system update --class-
loggers=com.example.biz=DEBUG,com.example.dao=INFO
日志配置更新成功。
```

10.5.2 访问日志配置 (log-config access)

`log-config access` 子命令用于查看、更新 HTTP 访问日志 (Access Log) 配置。

10.5.2.1 概要

```
Usage: log-config access [-h] [COMMAND]
```

访问日志配置

`-h, --help` 显示帮助信息

Commands:

`get` 查看访问日志配置

`update` 更新访问日志配置

`log-config access update` 主要选项:

分类	选项	说明
基础配置	<code>--enabled=</code>	启用访问日志 (true/false)
	<code>--virtual-hosts=</code>	应用虚拟主机, 多个以英文逗号分隔, 可用 <code>aascli host list</code> 查询候选项
	<code>--directory=</code>	日志文件路径, 支持变量如 <code>\${apusic.base}</code>
	<code>--encoding=</code>	日志文件编码, 可选: UTF-8/GBK, 默认 UTF-8
	<code>--buffer-enabled=</code>	写入缓冲区 (true/false)
	<code>--buffer-size=</code>	缓冲区大小, 单位字节, 默认 32768
	<code>--file-date-format=</code>	日志文件名日期格式, 如 <code>.yyyy-MM-dd</code>
格式配置	<code>--max-days=</code>	保留天数, 默认 30, 0 或负数表示不限制
	<code>--custom-format-enabled=</code>	启用自定义格式 (true/false)

	--log-content=	日志内容，多个以英文逗号分隔，可选：time、remoteAddr、method、protocol、status、uri、bytesSent、sessionId、threadName、serverName、localPort
	--pattern=	自定义访问日志的内容记录格式，使用占位符如 %a(远程IP)、%t(时间) 等，如 "%h %l %u %t \"%r\" %s %b"
	--date-format=	日期格式，控制时间字段输出格式（%{xxx}t 中的 xxx），默认 dd/MM/yyyy:HH:mm:ss
	--request-headers=	记录请求头，多个以英文逗号分隔
	--response-headers=	记录响应头，多个以英文逗号分隔
	--cookies=	记录 Cookie，多个以英文逗号分隔
	--session-attrs=	记录 Session，多个以英文逗号分隔
轮转与压缩	--rotate-by-date=	按天轮转（true/false）
	--max-size=	按大小轮转，单位 MB，0 表示不限制
	--compress-enabled=	日志压缩（true/false）
	--compress-time=	日志压缩时间，格式 HH:mm，默认 00:00
	--compress-dir=	日志压缩路径，支持变量如 \${apusic.base}/logs/archive

10.5.2.2 示例

示例1 启用访问日志

```
admin@localhost> log-config access update --enabled=true
```

日志配置更新成功。

示例2 查看访问日志配置

```
admin@localhost> log-config access get
```

【基础配置】

```

启用访问日志： 开启
应用虚拟主机： localhost
日志文件路径： ${apusic.base}/logs/localhost_access_log
日志文件编码： UTF-8
写入缓冲区： 开启
缓冲区大小： 32768
日志文件名日期格式： .yyyy-MM-dd
保留天数： 30

```

【日志格式配置】

```

启用自定义格式：
日志内容： time, remoteAddr, method, uri, status
日期格式： dd/MM/yyyy:HH:mm:ss
记录请求头：
记录响应头：
记录 Cookie：
记录 Session：

```

【日志轮转配置】

```

按天轮转： 开启
按大小轮转： 30 MB

```

示例3 关闭访问日志

```

admin@localhost> log-config access update --enabled=false
日志配置更新成功。

```

10.6 用户管理

10.6.1 用户管理 (user)

`user` 子命令用于管控台用户管理，包括用户的创建、查看、删除及权限查看。用户的角色分配、权限配置由安全管理员负责，系统管理员可查看用户权限配置。

10.6.1.1 概要

```

Usage:  user [-h] [COMMAND]
用户管理

```

```
-h, --help    显示帮助信息

Commands:
  list        列出用户
  get         查看用户详情
  add         创建用户
  delete      删除用户
  permission  用户权限配置
```

user add 选项:

选项	说明
--username=	用户名（必填）。长度 1-64 字符，仅支持字母、数字、下划线、短横线
--password=	登录密码。脚本场景可指定，未指定时交互式输入并确认
--status=	用户状态：0=禁用，1=正常，2=锁定。默认：1
--email=	邮箱地址。有效邮箱格式，最长 255 字符
--phone=	手机号码。中国手机号 1[3-9]XXXXXXXXXX
--description=	说明。最长 255 字符

10.6.1.2 示例

示例1 创建用户

```
admin@localhost> user add --username=testuser --password=Test@12345 --
description=CLI测试用户
用户创建成功。
```

注意：新创建的用户尚未分配角色，此时登录会提示 登录失败：用户未授权，请联系管理员 。需由安全管理员为用户分配角色后，用户方可登录使用。

示例2 列出用户

```
admin@localhost> user list
| 用户名 | 状态 | 角色 | 密级 | 过期时间 | 特权标识 |
+-----+-----+-----+-----+-----+-----+
| testuser | 正常 |  | 秘密 | 永久 | 否 |
| admin | 正常 | sysadmin | 绝密 | 永久 | 是 |
```

secure	正常	security	绝密	永久	是	
audit	正常	auditor	绝密	永久	是	

第 1 / 1 页, 共 4 条

示例3 查看用户详情

```
admin@localhost> user get testuser
```

ID: 4
 用户名: testuser
 状态: 正常
 密级: 秘密
 特权标识: 否
 角色:
 邮箱:
 手机号:
 OTP: 否
 过期时间: 永久
 创建时间: 2026-09-27 14:39:50
 描述: CLI测试用户

示例4 查看用户权限配置

```
admin@localhost> user permission get testuser
```

用户权限配置:
 用户名: testuser
 状态: 正常
 权限分配角色:
 过期时间: 永久
 允许访问 IP:
 开始访问时间:
 结束访问时间:
 数据密级: 秘密
 手机号:
 邮箱:
 描述: CLI测试用户

示例5 删除用户

```
admin@localhost> user delete testuser -f
```

用户删除成功。

10.7 安全服务

10.7.1 安全域管理 (security-domain)

`security-domain` 子命令用于管理安全域 (Realm)，为应用提供认证与授权数据来源，支持数据源、用户数据库、JAAS、JNDI (LDAP 等) 四种类型。

10.7.1.1 概要

```
Usage: security-domain [-h] [COMMAND]
```

安全域管理

`-h, --help` 显示帮助信息

Commands:

```
list      列出安全域
get       查看安全域详情
add       创建安全域
update    更新安全域
delete    删除安全域
```

`security-domain add` 基本选项 (`update` 时名称由参数指定，类型不提供则保持不变)：

选项	说明
<code>--name=</code>	安全域名称 (必填)，支持字母、数字、下划线、中横线，最长128字符
<code>--type=</code>	安全域类型 (必填)，可选：DataSourceRealm (数据源)、UserDatabaseRealm (用户数据库)、JAASRealm (JAAS)、JNDIRealm (JNDI)
<code>--description=</code>	安全域描述，最长500字符，不能包含 < 和 >
<code>--password-hash-algorithm=</code>	密码比较算法，可选空值 (明文) 或 MD5、SHA-1、SHA-256、SHA-512
<code>--lock-enabled=</code>	是否启用登录失败锁定 (true/false，默认 false)
<code>--max-failed-attempts=</code>	允许错误次数，启用锁定时连续失败多少次后锁定账户 (默认 5)
<code>--lock-timeout=</code>	锁定超时时间 (秒)，账户被锁定后多久自动解锁 (默认 300)

各类型专属选项：

类型	主要选项
DataSourceRealm	--data-source-jndi (数据源 JNDI 名称, 必填)、--user-table、--username-field、--password-field、--role-table、--role-field
UserDatabaseRealm	--resource-name (apusic.conf 中已配置的 UserDatabase 资源名, 必填)
JAASRealm	--user-principal-class (必填)、--role-principal-class (必填)、--jaas-module (LoginModule 配置, 格式: loginModuleClass@controlFlag@options, 多个模块用分号分隔)
JNDIRealm	--context-factory、--jndi-connection-url (必填, 如 ldap://localhost:389)、--connection-username、--connection-password、--user-search-base、--user-search-filter (如 (uid={0}))、--role-search-base、--role-search-filter (如 (member={0}))、--user-subtree、--role-subtree 等

各类型选项的完整说明可通过 `security-domain add --help` 查看。

10.7.1.2 示例

示例1 创建用户数据库类型安全域

```
admin@localhost> security-domain add --name=test-realm --
type=UserDatabaseRealm --resource-name=UserDatabase
安全域创建成功。
```

示例2 列出安全域

```
admin@localhost> security-domain list
| 安全域名称 | 类型 | 绑定应用 | 失败锁定 |
+-----+-----+-----+-----+
| test-realm | UserDatabaseRealm | -- | 关闭 |
第 1 / 1 页, 共 1 条
```

示例3 查看安全域详情

```
admin@localhost> security-domain get test-realm
【基本配置】
ID: 1
安全域名称: test-realm
```

安全域类型: UserDatabaseRealm

描述:

密码比较算法:

【锁定机制】

启用锁定: 关闭

【UserDatabase 配置】

资源名字: UserDatabase

示例4 更新安全域

```
admin@localhost> security-domain update test-realm --lock-enabled=true
--max-failed-attempts=5 --lock-timeout=300
```

示例5 删除安全域

```
admin@localhost> security-domain delete test-realm -f
安全域删除成功。
```

示例6 应用绑定安全域

创建安全域后, 可通过 `app update` 为应用绑定:

```
admin@localhost> app update testapp --security-domain=test-realm --
reload
应用配置更新成功。
应用重新加载成功。
```

10.8 系统与账户

10.8.1 用户设置 (usersettings)

`usersettings` 子命令用于查看或修改当前登录用户的个人设置, 包括命令行分页、语言、历史记录条数及个人信息。

10.8.1.1 概要

```
Usage: usersettings [-hl] [--no-paging] [--history-size=<historySize>]
[--lang=<language>] [--page-size=<pageSize>]
[COMMAND]
```

用户设置

-h, --help	显示帮助信息
--history-size=<historySize>	历史命令记录条数
-l, --list	查看当前设置
--lang=<language>	语言: en / zh / zh_TW
--no-paging	关闭分页输出
--page-size=<pageSize>	每页记录数

Commands:

profile 查看或修改个人信息

```
Usage: usersettings profile [-h] [--description=<description>] [--
email=<email>] [--phone=<phone>]
```

查看或修改个人信息

--description=<description>	说明
--email=<email>	邮箱地址
-h, --help	显示帮助信息
--phone=<phone>	手机号码

10.8.1.2 选项

```
-l, --list
```

查看当前设置。

```
--lang=<language>
```

设置命令行界面语言: en / zh / zh_TW。

```
--page-size=<pageSize>
```

设置列表查询每页显示的记录数。

```
--no-paging
```

关闭命令行分页输出，列表一次输出全部记录。

```
--history-size=<historySize>
```

设置历史命令记录条数。

10.8.1.3 示例

示例1 查看当前设置

```
admin@localhost> usersettings -l
```

当前配置：

分页输出：开启

每页条数：20

语言：zh

历史命令记录条数：1,000

示例2 修改每页记录数与历史记录条数

```
admin@localhost> usersettings --page-size=50 --history-size=2000
```

示例3 查看个人信息

```
admin@localhost> usersettings profile
```

用户名：admin

手机：-

邮箱：-

说明：-

示例4 修改个人信息

```
admin@localhost> usersettings profile --email=admin@example.com --  
phone=13800000000
```

10.8.2 修改当前登录用户的密码 (changepassword)

`changepassword` 子命令用于修改当前登录用户自己的密码。

10.8.2.1 概要

```
Usage: changepassword [-h] [--new-password=<newPassword>] [--old-
password=<oldPassword>]
```

修改当前登录用户的密码

-h, --help

显示帮助信息

--new-password=<newPassword>

新密码;未提供时交互式输入

--old-password=<oldPassword>

旧密码;未提供时交互式输入

10.8.2.2 选项

```
--old-password=<oldPassword>
```

当前登录用户的旧密码。未提供时交互式输入（输入不回显）。

```
--new-password=<newPassword>
```

新密码。未提供时交互式输入并二次确认（输入不回显）。新密码需满足密码复杂度要求。

10.8.2.3 示例

示例1 交互式修改密码

```
admin@localhost> changepassword
```

旧密码:

新密码:

确认新密码:

密码修改成功, 请重新 login 登录。

密码修改成功后当前会话失效, 需重新执行 `login` 登录。

示例2 指定旧密码与新密码

```
admin@localhost> changepassword --old-password=Old@12345 --new-
password=New@67890
```

密码修改成功, 请重新 login 登录。

11 安全管理员功能模块

安全管理员使用内置账号 `secure`（角色 `security`）登录 `aascli`，负责角色管理、用户授权与安全策略配置。安全管理员登录后可见的命令如下：

```
secure@localhost> help
用户管理：
    user          用户管理

角色管理：
    role          角色管理

安全服务：
    security-config 安全配置管理

系统与账户：
    usersettings  用户设置
    changepassword 修改当前登录用户的密码
    logout        退出登录
    help          显示可用命令
```

`usersettings`、`changepassword`、`logout`、`help` 为通用账户命令，用法与系统管理员一致，参见"系统与账户"章节，此处不再重复。

安全管理员与系统管理员的用户管理职责分工如下：

操作	系统管理员（admin）	安全管理员（secure）
创建用户（user add）	支持	不支持
列出/查看用户	支持	支持
删除用户（user delete）	支持	支持
重置用户密码（user reset-password）	不支持	支持
分配/解除用户角色（user permission update）	不支持（仅可查看）	支持

11.1 用户管理（user）

`user` 子命令用于用户管理。安全管理员可查看、删除用户，重置用户密码，并配置用户权限（角色绑定、访问控制等）。

11.1.1 概要

```
Usage:  user [-h] [COMMAND]
用户管理
    -h, --help      显示帮助信息
Commands:
    list            列出用户
    get             查看用户详情
    delete         删除用户
    reset-password  重置用户密码
    permission      用户权限配置
```

11.1.2 列出用户 (user list)

用法与系统管理员一致，展示用户名、状态、角色、密级等信息。

11.1.2.1 示例

示例1 列出用户

```
secure@localhost> user list
| 用户名   | 状态 | 角色       | 密级 | 过期时间 | 特权标识 |
+-----+-----+-----+-----+-----+-----+
| admin    | 正常 | sysadmin  | 绝密 | 永久     | 是       |
| secure   | 正常 | security  | 绝密 | 永久     | 是       |
| audit    | 正常 | auditor   | 绝密 | 永久     | 是       |
第 1 / 1 页, 共 3 条
```

11.1.3 查看用户详情 (user get)

用法与系统管理员一致。

11.1.3.1 示例

示例1 查看用户详情

```
secure@localhost> user get admin
ID: 1
```

```

用户名: admin
状态: 正常
密级: 绝密
特权标识: 是
角色: sysadmin
邮箱:
手机号:
OTP: 否
过期时间: 永久
创建时间: 2026-09-27 11:23:52
描述:

```

11.1.4 重置用户密码 (user reset-password)

`user reset-password` 子命令用于重置指定用户的密码，用户忘记密码或账号密码泄露时使用。

11.1.4.1 概要

```
Usage: user reset-password <ID> [--password=<password>] [-h]
```

重置用户密码

<code><id></code>	用户 ID
<code>--password=<password></code>	新密码。脚本场景可指定，未指定时交互式输入并确认
<code>--help</code>	显示帮助信息

11.1.4.2 选项

```
<ID>
```

用户 ID，可通过 `user get <用户名>` 查询。注意此处参数为用户 ID 而非用户名。

```
--password=<password>
```

新密码，需符合当前密码策略。未指定时交互式输入并二次确认（输入不回显）。

11.1.4.3 示例

示例1 重置用户密码

```
secure@localhost> user reset-password 7 --password=New@67890
```

密码重置成功。

11.1.5 删除用户 (user delete)

用法与系统管理员一致，删除前建议先解除用户角色绑定。

11.1.5.1 示例

示例1 删除用户

```
secure@localhost> user delete testuser2 -f
```

用户删除成功。

11.1.6 用户权限配置 (user permission)

`user permission` 子命令用于配置用户权限，包括角色绑定、用户状态、账号有效期、访问 IP 白名单、访问时段、数据密级等，是安全管理员的核心操作。

11.1.6.1 概要

```
Usage: user permission [-h] [COMMAND]
```

用户权限配置

`-h, --help` 显示帮助信息

Commands:

`get` 查看用户权限配置

`update` 更新用户权限配置

```
Usage: user permission update <USERNAME> [--status=<status>] [--role=
<code>|--unbind-role] [--expire-date=<date>]
[--ip-whitelist=<ips>] [--start-time=<time>] [--end-time=<time>] [--
class-level=<level>] [--phone=<phone>]
[--email=<email>] [--description=<desc>] [-h]
```

更新用户权限配置

`<username>`

用户名，可用 `aascli user list` 查询

`--status=<status>`

用户状态：0=禁用，1=正常，2=锁定

`--role=<role>`

权限分配角色编码，可通过 ``aascli role`

`list`` 查看可用角色编码，或通过 ``aascli user`

`permission get <USERNAME>`` 查看

当前用户的角色编码

<code>--unbind-role</code>	解除用户当前绑定的角色，与 <code>--role</code> 互斥
<code>--expire-date=<expireDate></code> 为空表示永久有效	过期时间，格式 <code>yyyy-MM-dd HH:mm:ss</code> ，
<code>--ip-whitelist=<ipWhitelist></code> 为不限制	允许访问 IP，多个用英文逗号分隔，为空表示不限制
<code>--start-time=<startTime></code> <code>HH:mm:ss</code>	开始访问时间，格式 <code>yyyy-MM-dd</code>
<code>--end-time=<endTime></code> <code>HH:mm:ss</code>	结束访问时间，格式 <code>yyyy-MM-dd</code>
<code>--class-level=<classLevel></code>	数据密级：2=秘密，3=机密，4=绝密
<code>--phone=<phone></code> 9]XXXXXXXXXX	手机号码。取值范围：中国手机号 1[3-
<code>--email=<email></code> 255 字符	邮箱地址。取值范围：有效邮箱格式，最长
<code>--description=<description></code>	说明。取值范围：最长 255 字符
<code>--help</code>	显示帮助信息

11.1.6.2 选项

```
--status=<status>
```

用户状态：0=禁用，1=正常，2=锁定。禁用后用户无法登录。

```
--role=<role>
```

权限分配的角色编码，可通过 `role list` 查看可用角色编码。用户只有绑定了角色才能登录管控台或 aascli。

```
--unbind-role
```

解除用户当前绑定的角色，与 `--role` 互斥。解绑后用户无法登录。

```
--expire-date=<expireDate>
```

账号过期时间，格式 `yyyy-MM-dd HH:mm:ss`，为空表示永久有效。过期后账号无法登录。含空格的日期时间需使用引号包裹。

```
--ip-whitelist=<ipWhitelist>
```

允许访问的 IP 白名单，多个用英文逗号分隔，为空表示不限制。仅支持 IPv4、IPv6 地址或 * 通配符（如 192.168.1.*），不支持 CIDR 格式（如 192.168.1.0/24）。

```
--start-time=<startTime> / --end-time=<endTime>
```

允许访问的开始/结束时间，格式 yyyy-MM-dd HH:mm:ss，用于限制用户仅在指定时段内访问。

```
--class-level=<classLevel>
```

用户数据密级：2=秘密，3=机密，4=绝密。

```
--phone=<phone> / --email=<email> / --description=<description>
```

用户联系方式与说明信息，与 `usersettings profile` 的区别在于可维护其他用户的信息。

11.1.6.3 示例

示例1 为用户分配角色并设置账号有效期与数据密级

系统管理员创建用户后，需由安全管理员分配角色，用户方可登录：

```
secure@localhost> user permission update testuser2 --role=test-role --
expire-date="2027-12-31 23:59:59" --class-level=3
用户权限配置已保存。
```

```
secure@localhost> user permission get testuser2
```

用户权限配置：

用户名：testuser2

状态：正常

权限分配角色：test-role

过期时间：2027-12-31 23:59:59

允许访问 IP：

开始访问时间：

结束访问时间：

数据密级：机密

手机号：

邮箱：

描述：

示例2 设置用户访问 IP 白名单

仅支持 IPv4、IPv6 地址或 * 通配符：

```
secure@localhost> user permission update testuser2 --ip-
whitelist=192.168.1.*,10.0.0.8
用户权限配置已保存。
```

传入 CIDR 格式会被拒绝并提示：

```
secure@localhost> user permission update testuser2 --ip-
whitelist=192.168.1.0/24,10.0.0.8
用户操作失败：允许访问IP格式非法，只允许IPv4、IPv6地址或*通配符
```

示例3 禁用/启用用户

```
secure@localhost> user permission update testuser2 --status=0
用户权限配置已保存。

secure@localhost> user permission update testuser2 --status=1
用户权限配置已保存。
```

示例4 解除用户角色绑定

```
secure@localhost> user permission update testuser2 --unbind-role
用户权限配置已保存。
```

11.2 角色管理 (role)

`role` 子命令用于角色管理，包括角色的创建、查看、更新、删除及可用权限资源查询。角色是权限的集合，通过 `user permission update --role` 绑定到用户。

11.2.1 概要

```
Usage:  role [-h] [COMMAND]
```

角色管理

```
-h, --help    显示帮助信息
```

```
Commands:
```

list	列出角色
get	查看角色详情
add	创建角色
update	更新角色
delete	删除角色
permissions	列出可用权限资源

11.2.2 列出角色 (role list)

11.2.2.1 概要

```
Usage: role list [--page=<page>] [--output=<format>] [--no-paging] [-h]
```

列出角色

--page=<page>	页码
--help	显示帮助信息
--no-paging	本次命令关闭分页输出
--output=<output>	输出格式: table 或 json

11.2.2.2 示例

示例1 列出角色

```
secure@localhost> role list
| 角色名 | 关联用户 | 特权标识 | 说明 |
+-----+-----+-----+-----+
--+
| sysadmin | admin | 是 | System maintenance and user creation |
| security | secure | 是 | Security policy and authorization |
| auditor | audit | 是 | Operation supervision and audit |
第 1 / 1 页, 共 3 条
```

sysadmin、security、auditor 为系统内置特权角色，不可修改、不可删除。

11.2.3 查看角色详情 (role get)

11.2.3.1 概要

```
Usage: role get <ROLE_CODE> [--output=<format>] [-h]
```

查看角色详情

<code><roleCode></code>	角色编码, 可用 <code>aascli role list</code> 查询
<code>--output=<output></code>	输出格式: json
<code>--help</code>	显示帮助信息

11.2.3.2 示例

示例1 查看角色详情

```
secure@localhost> role get test-role
ID: 4
角色编码: test-role
角色名称: test-role
说明: CLI-test-role
内置标识: 否
数据密级: 公开
权限列表: dashboard:view, app_list:view
```

11.2.4 列出可用权限资源 (role permissions)

`role permissions` 子命令用于列出系统中全部可用权限资源及其编码, 创建或更新角色时 `--permissions` 参数取值即来自该列表。

11.2.4.1 概要

```
Usage: role permissions [--output=<format>] [-h]
```

列出可用权限资源

<code>--help</code>	显示帮助信息
<code>--output=<output></code>	输出格式: json

11.2.4.2 示例

示例1 列出可用权限资源

输出为 json 格式的权限资源树, 每个资源包含 `code` (权限编码)、`name` (名称) 及 `children` (子操作权限, 如 view 查看、create 新增、update 更新、delete 删除) :

```
secure@localhost> role permissions --output=json
[ {
  "code" : "dashboard",
  "name" : "首页概览",
  "resourceId" : 1,
  "children" : [ {
    "code" : "dashboard:view",
    "name" : "查看",
    "resourceId" : null,
    "children" : [ ]
  } ]
}, {
  "code" : "user_mgmt",
  "name" : "用户管理",
  "resourceId" : 40,
  "children" : [ {
    "code" : "user_mgmt:view",
    "name" : "查看",
    ...
  } ]
}, ... ]
```

顶层权限资源编码包括：

权限编码	名称
dashboard	首页概览
user_mgmt	用户管理
app_mgmt	应用管理
role_mgmt	角色管理
config_mgmt	配置管理
monitor_mgmt	监控管理
resource_mgmt	资源管理
log_service	日志服务

security_service	安全服务
system_mgmt	系统管理
audit_log	审计日志
security_config	安全配置
ops_mgmt	运维巡检

11.2.5 创建角色 (role add)

11.2.5.1 概要

```
Usage: role add --name=<name> --permissions=<codes> [--description=
<desc>] [-h]
```

创建角色

```
--name=<name>                角色名，必填。取值范围：字母、数字、下划
线、短横线，最长 64 字符
--permissions=<permissions>  权限资源编码，多个用英文逗号分隔，必填。
可通过 `aascli role permissions` 查看可用编
码
--description=<description>  说明。取值范围：最长 255 字符
--help                        显示帮助信息
```

11.2.5.2 选项

```
--name=<name>
```

角色名（即角色编码），字母、数字、下划线、短横线，最长 64 字符。

```
--permissions=<permissions>
```

权限资源编码，多个用英文逗号分隔，必填。顶层资源编码（如 app_mgmt）与具体操作编码（如 app_list:view）均可使用。

```
--description=<description>
```

角色说明，最长 255 字符。

11.2.5.3 示例

示例1 创建角色

```
secure@localhost> role add --name=test-role --
permissions=dashboard:view,app_list:view --description=CLI-test-role
角色创建成功。
```

11.2.6 更新角色 (role update)

11.2.6.1 概要

```
Usage: role update <ROLE_CODE> [--permissions=<codes>] [--description=
<desc>] [-h]
```

更新角色

<roleCode>	角色编码
--permissions=<permissions>	权限资源编码，多个用英文逗号分隔。可通过 `aascli role permissions` 查看可用编码
--description=<description>	说明。取值范围：最长 255 字符
--help	显示帮助信息

11.2.6.2 示例

示例1 更新角色权限与说明

更新角色时必须携带 `--permissions`，且权限列表为全量覆盖式更新（未保留的权限将被移除）：

```
secure@localhost> role update test-role --permissions=dashboard:view -
-description=updated-desc
角色更新成功。
```

仅指定 `--description` 不指定 `--permissions` 时会提示：

```
secure@localhost> role update test-role --description=updated-desc
角色操作失败：请至少为角色分配一个权限资源
```

11.2.7 删除角色 (role delete)

11.2.7.1 概要

```
Usage: role delete <ROLE_CODE> [-f] [-h]
删除角色
```

<roleCode>	角色编码
-f, --force	不提示确认
--help	显示帮助信息

11.2.7.2 示例

示例1 删除角色

```
secure@localhost> role delete test-role -f
```

角色删除成功。

示例2 删除内置角色被拒绝

```
secure@localhost> role delete sysadmin -f
```

内置角色不能删除。

11.3 安全配置管理 (security-config)

`security-config` 子命令用于安全策略配置，包括密码策略、审计日志配置、业务配置备份、邮件验证与动态码验证。

11.3.1 概要

```
Usage: security-config [-h] [COMMAND]
```

安全配置管理

-h, --help 显示帮助信息

Commands:

password-policy	密码策略配置
audit-log	审计日志配置
backup	业务配置备份管理
mail	邮件验证配置
otp	动态码验证配置

11.3.2 密码策略配置 (security-config password-policy)

`password-policy` 子命令用于查看、更新管控台账号的密码策略。

11.3.2.1 概要

Usage: security-config password-policy [-h] [COMMAND]

密码策略配置

-h, --help 显示帮助信息

Commands:

get 查看密码策略配置

update 更新密码策略配置

security-config password-policy update 选项:

选项	说明
--password-min-length=	密码最小长度，取值范围 8~32 位，默认 8
--password-valid-days=	密码强制修改周期（天），取值范围 1~90，默认 30
--password-max-failed-attempts=	连续登录失败次数上限，取值范围 1~5 次，默认 5
--password-complexity=	密码复杂度，normal=普通（字母、数字、特殊符号至少 2 种），complex=复杂（需同时包含 3 种），默认 normal
--password-history-limit=	禁止重复使用的历史密码数量，取值范围 1~10 次，默认 5
--password-lock-duration=	账号锁定时长（分钟），取值范围 1~1440，默认 15
--weak-passwords=	弱密码列表，多个用英文逗号分隔，最多 255 个字符，默认空

11.3.2.2 示例

示例1 查看密码策略配置

secure@localhost> security-config password-policy get

密码策略配置:

密码长度: 8 位

密码有效天数: 30 天

密码重试次数: 5 次

密码复杂度: normal

最近密码重复限制: 5 次

用户锁定时间: 15 分钟

弱密码设置:

示例2 提高密码强度要求

```
secure@localhost> security-config password-policy update --password-
min-length=10 --password-complexity=complex
```

安全配置已保存。

```
secure@localhost> security-config password-policy get
```

密码策略配置：

密码长度：10 位

密码有效天数：30 天

密码重试次数：5 次

密码复杂度：complex

最近密码重复限制：5 次

用户锁定时间：15 分钟

弱密码设置：

11.3.3 审计日志配置 (security-config audit-log)

`audit-log` 子命令用于查看、更新审计日志（登录日志、操作日志）的保留策略与定时备份策略。审计日志的查询与导出由审计管理员负责。

11.3.3.1 概要

```
Usage: security-config audit-log [-h] [COMMAND]
```

审计日志配置

`-h, --help` 显示帮助信息

Commands:

`get` 查看审计日志配置

`update` 更新审计日志配置

`security-config audit-log update` 选项：

选项	说明
<code>--login-log-max-count=</code>	登录日志数量上限（条），范围 1000~100000，默认 10000
<code>--login-log-retention=</code>	登录日志保留时间（天），范围 30~365，默认 180
<code>--operation-log-max-count=</code>	操作日志数量上限（条），范围 1000~100000，默认 10000

<code>--operation-log-retention=</code>	操作日志保留时间（天），范围 30~365，默认 180
<code>--audit-backup-period-days=</code>	审计日志定时备份周期（天/次），范围 1~365，默认 30
<code>--audit-backup-retention-days=</code>	审计日志定时备份保留时间（天），-1 表示永久保留，默认 -1

11.3.3.2 示例

示例1 查看审计日志配置

```
secure@localhost> security-config audit-log get
```

审计日志配置：

```

登录日志数量上限：10000 条
登录日志保留时间：180 天
操作日志数量上限：10000 条
操作日志保留时间：180 天
定时备份周期：30 天/次
定时备份保留时间：永久

```

示例2 调整日志保留时间

```
secure@localhost> security-config audit-log update --login-log-retention=90 --operation-log-retention=90
```

安全配置已保存。

11.3.4 邮件验证配置 (security-config mail)

`mail` 子命令用于查看、更新登录邮件验证配置。开启后，用户登录需输入邮箱验证码。

11.3.4.1 概要

```
Usage: security-config mail [-h] [COMMAND]
```

邮件验证配置

```
-h, --help 显示帮助信息
```

Commands:

```

get      查看邮件验证配置
update   更新邮件验证配置

```

`security-config mail update` 选项：

选项	说明
--smtp-verify-enabled=	邮件验证开关：true 或 false，默认 false。开启后登录需输入邮箱验证码，并自动关闭动态码验证
--smtp-host=	SMTP 服务器地址，如 smtp.example.com，默认空
--smtp-ssl=	SMTP SSL 开关：true 或 false，默认 false
--smtp-port=	SMTP 端口，范围 1~65535，默认 25（SSL 时通常为 465）
--smtp-user=	发送邮箱账号，默认空
--smtp-password=	发送邮箱密码或客户端授权码，默认空
--smtp-license-email=	接收授权提醒邮箱，多个用英文分号分隔，默认空

11.3.4.2 示例

示例1 查看邮件验证配置

```
secure@localhost> security-config mail get
```

邮件验证配置：

邮件验证开关：false

发送邮件服务器：

开启SSL：false

发送端口：25

发送邮箱：

发送邮箱密码：

接收授权提醒邮箱：

示例2 配置并开启邮件验证

开启邮件验证前，需先配置可用的 SMTP 服务，否则所有用户将无法登录：

```
secure@localhost> security-config mail update --smtp-
host=smtp.example.com --smtp-ssl=true --smtp-port=465 --smtp-
user=alerts@example.com --smtp-password=***** --smtp-verify-
enabled=true
```

安全配置已保存。

11.3.5 动态码验证配置 (security-config otp)

`otp` 子命令用于查看、更新登录动态码（OTP）验证配置。开启后，用户登录需同时输入密码与 Authenticator 动态验证码。

11.3.5.1 概要

```
Usage: security-config otp [-h] [COMMAND]
```

动态码验证配置

`-h, --help` 显示帮助信息

Commands:

`get` 查看动态码验证配置

`update` 更新动态码验证配置

```
Usage: security-config otp update --enabled=<true|false> [-h]
```

更新动态码验证配置

`--enabled=<true|false>` 动态码验证开关: `true` 或 `false`。开启后登录需同时输入密码与 Authenticator 动态验证码，并自

动关闭邮件验证

`--help` 显示帮助信息

11.3.5.2 示例

示例1 查看动态码验证配置

```
secure@localhost> security-config otp get
```

动态码验证配置:

OTP验证开关: `false`

示例2 开启/关闭动态码验证

动态码验证与邮件验证互斥，开启其中一个会自动关闭另一个：

```
secure@localhost> security-config otp update --enabled=true
安全配置已保存。
```

```
secure@localhost> security-config otp update --enabled=false
安全配置已保存。
```

11.3.6 业务配置备份管理 (security-config backup)

`backup` 子命令用于业务配置的备份与恢复，支持定时备份策略配置、手动立即备份、备份查看、恢复与删除。

11.3.6.1 概要

Usage: `security-config backup [-h] [COMMAND]`

业务配置备份管理

`-h, --help` 显示帮助信息

Commands:

<code>get</code>	查看业务配置备份设置
<code>update</code>	更新业务配置备份设置
<code>backup-now</code>	立即执行业务配置备份
<code>list</code>	列出业务配置备份
<code>restore</code>	恢复业务配置备份
<code>delete</code>	删除业务配置备份

`security-config backup update` 选项:

选项	说明
<code>--backup-enabled=</code>	定时备份开关: true 或 false, 默认 false
<code>--backup-path=</code>	共享备份目录, 支持系统变量, 默认 <code>\${apusic.home}/store/config_bak</code>
<code>--backup-period-months=</code>	定时备份周期 (月/次), 范围 1~12, 默认 1

`security-config backup restore` 选项:

选项	说明
	备份文件名, 可通过 <code>security-config backup list</code> 查询
<code>-f, --force</code>	不提示恢复确认
<code>-r, --restart</code>	恢复成功后立即重启, 不再提示确认

11.3.6.2 示例

示例1 查看业务配置备份设置

```
secure@localhost> security-config backup get
```

业务配置备份设置:

```
定时备份开关: false
共享备份目录: ${apusic.home}/store/config_bak
定时备份周期: 1 月/次
```

示例2 开启定时备份

```
secure@localhost> security-config backup update --backup-enabled=true
--backup-period-months=1
安全配置已保存。
```

示例3 立即执行业务配置备份

```
secure@localhost> security-config backup backup-now
备份成功：业务配置备份成功
```

示例4 列出业务配置备份

```
secure@localhost> security-config backup list
| 备份文件名 | 备份类型 | 备份时间 |
| 文件大小 |
+-----+-----+-----+
-----+-----+
| config_backup_manual_20260927_172533.zip | 手动备份 | 2026-09-27
17:25:33 | 36.33 KB |
第 1 / 1 页, 共 1 条
```

示例5 恢复业务配置备份

恢复会以备份文件内容覆盖当前业务配置，恢复后需重启服务器生效：

```
secure@localhost> security-config backup restore
config_backup_manual_20260927_172533.zip
```

示例6 删除业务配置备份

```
secure@localhost> security-config backup delete
config_backup_manual_20260927_172533.zip -f
备份已删除。
```

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

