



APUSIC
固若长城
睿比世界

安装手册

金蝶Apusic分布式配置中心V1.0.243

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

版权声明

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

免责声明

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

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

商标声明

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

目录

- .1 概述
 - .1.1 产品简介
 - .1.2 范围和读者
 - .1.3 约定和术语
- .2 系统环境要求
 - .2.1 环境要求
- .3 安装前准备
 - .3.1 安装JDK
 - .3.2 产品安装包
 - .3.2.1 授权文件
 - .3.2.2 获取特征码
 - .3.2.3 集中授权
 - .3.3 安装数据库 (单机模式下可选)
 - .3.4 负载均衡服务器(集群模式下要求)
 - .3.5 Visual C++ 2015 (Windows系列要求)
- .4 安装
 - .4.1 单机部署模式
 - .4.1.1 解压安装包
 - .4.1.2 开启数据库配置 (可选)
 - .4.2 集群部署模式
 - .4.2.1 解压安装包
 - .4.2.2 开启数据库配置 (可选)
 - .4.2.3 开启集群配置
 - .4.2.4 开启负载均衡配置
- .5 启动
 - .5.1 单机部署模式
 - .5.2 集群部署模式
- .6 停止
- .7 卸载
- .8 云部署
 - .8.1 build docker镜像
 - .8.1.1 build adcc镜像
 - .8.1.2 build mysql镜像
 - .8.2 配置参数
- .9 其他操作
 - .9.1 访问控制台
 - .9.2 开启鉴权
 - .9.3 修改JWT令牌的密钥
 - .9.4 修改密码

1 概述

1.1 产品简介

金蝶Apusic分布式配置中心软件for nacos (Apusic Distributed Config Center, 简称: ADCC for nacos) 是一款易于配置管理和服务管理平台, 旨在帮助构建云原生应用。它提供了服务注册与发现功能, 让服务提供者能便捷地将自身信息注册, 服务消费者可轻松获取服务列表, 保障服务间通信顺畅; 配置管理功能支持集中化管理应用配置, 实时推送配置变更, 确保应用使用最新配置; 还具备服务管理能力, 涵盖服务健康检查、流量管理等, 助力优化服务性能。ADCC for nacos支持多种主流语言客户端, 拥有高可用集群架构, 广泛应用于微服务架构、云原生等场景, 大幅提升了应用开发与运维的效率。

1.2 范围和读者

本手册介绍ADCC产品安装相关的内容, 主要适用于实施人员, 维护人员等。

1.3 约定和术语

ADCC: 金蝶Apusic分布式配置中心软件 (Apusic Distributed Config Center, 简称: ADCC)

ADCC_HOME: 金蝶Apusic分布式配置中心软件安装目录

2 系统环境要求

ADCC产品支持Windows Server、Linux（包括采用龙芯和飞腾等国产芯片的Linux服务器）、Unix等多个操作系统平台的安装部署。

2.1 环境要求

表格1-1 软件及操作系统环境要求

组件	要求
操作系统	Windows系列; Linux Red Hat 5.2或以上; 国产操作系统如银河麒麟系列、中标麒麟系列、普华、中科红旗、深度等
CPU	X86、鲲鹏、飞腾、龙芯、申威等
内存	4G或以上
硬盘	可用空间100G或以上
数据库	达梦、人大金仓、瀚高、MySQL等
浏览器	FireFox 70及以上、Chrome 60及以上、IE 11及以上

3 安装前准备

3.1 安装JDK

ADCC运行环境依赖于JAVA运行环境，在运行ADCC之前，需要具备。

目前支持的JDK厂商版本包括：Oracle JDK 8+、Open JDK 8+、IBM JDK 8+。

3.2 产品安装包

ADCC产品安装包：`ADCC-V1.0.XXX-Nacos-XXXXXXXXX.tar.gz` 或 `ADCC-V1.0.XXX-Nacos-XXXXXXXXX.zip`，如 `ADCC-V1.0.243-Nacos-20241218.tar.gz`；安装包适用于windows/linux等操作系统。

ADCC产品安装包目录结构和端口使用情况如下：

表2-1 ADCC产品包主要目录结构说明

目录	内容
bin	ADCC的脚本目录
conf	ADCC的配置文件目录
target	ADCC的程序包放置目录

表2-2 ADCC端口使用说明

端口	说明
8848	主端口，客户端、控制台及OpenAPI所使用的HTTP端口
9848	客户端gRPC请求服务端端口，用于客户端向服务端发起连接和请求
9849	服务端gRPC请求服务端端口，用于服务间同步等
7848	Jraft请求服务端端口，用于处理服务端间的Raft相关请求

授权认证

ADCC for nacos启动需要依赖授权文件进行校验，一般情况下产品自带有效授权文件，如果授权文件将要过期或者已经过期，可联系金蝶天燕对接人员申领授权文件（`license.xml`），与此同时需要将产品特征码发给对接人员。

3.2.1 授权文件

获取到授权文件后，命名为 `license.xml`，放置ADCC安装目录下即可。

```
[root@linux-2-41 adcc]# ll
total 32
drwxr-xr-x 2 root root  99 May 21 18:40 bin
drwxr-xr-x 2 root root 310 Feb 20 13:45 conf
drwxr-xr-x 7 root root  83 May 21 18:41 data
-rw-r--r-- 1 root root 16583 Feb  5 17:23 LICENSE
-rw-r--r-- 1 root root 2755 May 24 10:28 license.xml
drwxr-xr-x 2 root root 4096 May 28 17:57 logs
-rw-r--r-- 1 root root 1305 Feb  5 17:23 NOTICE
drwxr-xr-x 3 root root  40 May 21 18:40 target
drwxr-xr-x 3 root root  17 May 21 18:40 work
[root@linux-2-41 adcc]#
```

3.2.2 获取特征码

在 `/${ADCC_HOME}/bin`，执行 `startup.sh -ac host`，`host`取值为ip地址或者网卡名称，类似如下：

```
./startup.sh -ac 172.20.140.17
```

打印特征码信息，类似如下，Auth Code=特征码内容：

```
ADCC JMX enabled by default
Auth Code=SZTY-777387783
```

获取特征码后再提供特征码申请授权文件。

3.2.3 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。

在系统环境配置环境变量，或在ADCC安装目录根目录 \${ADCC_HOME} 下创建 `acls.properties` 文件，添加以下参数：

```
apusic_acls_enable=true
apusic_acls_authUrls=172.24.4.166:6886
apusic_acls_ns=apusic
apusic_acls_tenant=ApusicTest
```

连接参数说明：

参数名	参数值说明
<code>apusic_acls_enable</code>	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；
<code>apusic_acls_authUrls</code>	授权中心的地址，可设置多个授权地址，格式为ip1:port1,ip2:port2，如果一个授权地址链接失败，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；
<code>apusic_acls_ns</code>	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。
<code>apusic_acls_tenant</code>	设置该实例所属的租户名称，可选参数。

ADCC启动时将会自动连接到Apusic授权中心。

3.3 安装数据库（单机模式下可选）

ADCC支持单机和集群部署模式。在单机模式下，ADCC提供内置数据源来实现数据的存储，可选择不使用外置数据源，但不方便观察数据存储的基本情况。

当前支持MySQL5.6.5+、达梦8、人大金仓V8和瀚高数据库安全版v4.5.8作为外置数据源来实现数据的存储。

3.4 负载均衡服务器(集群模式下要求)

集群模式通过配置负载均衡服务器作为统一的访问入口，将请求分摊到多个ADCC服务器上。

当前支持Nginx作为负载均衡服务器，版本要求：1.18.0+。使用nginx请求时，需要配置成TCP转发，不能配置http2转发，否则连接会被nginx断开。因此，在安装Nginx时需要启用stream模块，通过 `nginx -v` 查看是否包含 `--with-stream` 编译参数，包含则表示支持TCP转发。

```
# /usr/local/nginx/sbin/nginx -V
nginx version: nginx/1.18.0
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
configure arguments: --prefix=/usr/local/nginx --with-stream --with-pcre=/opt/nginx/pcre-8.34
```

3.5 Visual C++ 2015（Windows系列要求）

ADCC在Windows平台下使用时，需要依赖Microsoft Visual C++ 2015，要求版本在14.25.28508.3以上。

4 安装

4.1 单机部署模式

4.1.1 解压安装包

将安装包 `ADCC-V1.0.243-Nacos-20241218.tar.gz` 拷贝至指定目录, 如 `/opt`, 执行解压命令, 目录 `/opt/adcc/` 作为 `${ADCC_HOME}`。

```
tar -zxvf ADCC-V1.0.243-Nacos-20241218.tar.gz
```

4.1.2 开启数据库配置 (可选)

ADCC支持MySQL5.6.5+、达梦8、人大金仓V8和瀚高数据库安全版v4.5.8作为外置数据源来实现数据的存储, 可根据实际需要任选一种数据库类型进行配置, 具体配置如下所示:

1) 配置mysql数据库:

- 初始化数据库

创建并初始化数据库adcc, 数据库初始化文件: `${ADCC_HOME}/config/mysql-schema.sql`。具体操作如下

```
#创建数据库adcc
mysql -unext -pxxx mysql -Ne "CREATE DATABASE adcc DEFAULT CHARSET utf8 COLLATE utf8_general_ci;"
#初始化数据库adcc
mysql -uroot -pxxx adcc < ADCC_HOME/config/mysql-schema.sql
```

- 开启数据源连接

修改配置文件 `${ADCC_HOME}/config/application.properties`。修改部分内容如下:

```
spring.sql.init.platform=mysql

### Count of DB:
db.num=1

### Connect URL of DB:
db.url.0=jdbc:mysql://172.24.4.159:3306/adcc?
characterEncoding=utf8&connectTimeout=20000&socketTimeout=60000&autoReconnect=true&useUnicode=true&us
### 指定数据库连接的用户名和密码
db.user.0=xxxx
db.password.0=xxxxxxx
```

- 放置JDBC驱动包

将mysql数据库的JDBC驱动包放置在 `${ADCC_HOME}/target/ext` 目录下。

2) 配置达梦数据库:

- 初始化数据库

创建并初始化数据库adcc, 数据库初始化文件: `${ADCC_HOME}/config/dm-schema.sql`。

- 开启数据源连接

修改配置文件 `${ADCC_HOME}/config/application.properties`。修改部分内容如下:

```

spring.sql.init.platform=dm

### Count of DB:
db.num=1

#db.jdbcDriverName=xxx
db.jdbcDriverName=dm.jdbc.driver.DmDriver

### Connect URL of DB:
db.url.0=jdbc:dm://172.24.4.159:5236/ADCC
### 指定数据库连接的用户名和密码
db.user.0=xxxx
db.password.0=xxxxxxx

```

- 放置JDBC驱动包

将达梦数据库的JDBC驱动包放置在 `${ADCC_HOME}/target/ext` 目录下。

3) 配置人大金仓数据库:

- 初始化数据库

创建并初始化数据库adcc, 数据库初始化文件: `${ADCC_HOME}/config/kingbase-schema.sql`。

- 开启数据源连接

修改配置文件 `${ADCC_HOME}/config/application.properties`。修改部分内容如下:

```

spring.sql.init.platform=kingbase

### Count of DB:
db.num=1

#db.jdbcDriverName=xxx
db.jdbcDriverName=com.kingbase8.Driver

### Connect URL of DB:
db.url.0=jdbc:kingbase8://172.24.4.159:54321/adcc
### 指定数据库连接的用户名和密码
db.user.0=xxxx
db.password.0=xxxxxxx

```

- 放置JDBC驱动包

将人大金仓数据库的JDBC驱动包放置在 `${ADCC_HOME}/target/ext` 目录下。

4) 配置瀚高数据库:

- 初始化数据库

创建并初始化数据库adcc, 数据库初始化文件: `${ADCC_HOME}/config/hgdb-schema.sql`。

- 开启数据源连接

修改配置文件 `${ADCC_HOME}/config/application.properties`。修改部分内容如下:

```

spring.sql.init.platform=hgdb

### Count of DB:
db.num=1

#db.jdbcDriverName=xxx
db.jdbcDriverName=com.highgo.jdbc.Driver

### Connect URL of DB:
db.url.0=jdbc:highgo://172.24.4.159:5866/adcc
### 指定数据库连接的用户名和密码
db.user.0=xxxx
db.password.0=xxxxxxx

```

- 放置JDBC驱动包

将瀚高数据库的JDBC驱动包放置在 `${ADCC_HOME}/target/ext` 目录下。

5) 配置gbase数据库:

- 初始化数据库

创建并初始化数据库adcc, 数据库初始化文件: `${ADCC_HOME}/config/gbase-schema.sql`。

【注意】创建数据库时需加上dbcompatibility参数, 且值不能为A。例如: `create database adcc dbcompatibility 'PG';`

- 开启数据源连接

修改配置文件 `${ADCC_HOME}/config/application.properties`。修改部分内容如下:

```

spring.sql.init.platform=gbase

### Count of DB:
db.num=1

#db.jdbcDriverName=xxx
db.jdbcDriverName=com.gbase.Driver

### Connect URL of DB:
db.url.0=jdbc:gbase://172.24.1.116:15400/adcc
### 指定数据库连接的用户名和密码
db.user.0=xxxx
db.password.0=xxxxxxx

```

- 放置JDBC驱动包

将gbase数据库的JDBC驱动包放置在 `${ADCC_HOME}/target/ext` 目录下。

4.2 集群部署模式

集群部署模式下需要3个或3个以上ADCC节点、负载均衡服务器和数据库才能构成集群。

4.2.1 解压安装包

将安装包 `ADCC-V1.0.243-Nacos-20241218.tar.gz` 拷贝至指定目录, 如 `/opt`, 执行解压命令, 目录 `/opt/adcc/` 作为 `${ADCC_HOME}`。

```
tar -zxvf ADCC-V1.0.243-Nacos-20241218.tar.gz
```

每个节点都需要安装ADCC。

4.2.2 开启数据库配置 (可选)

每个ADCC节点服务器上，配置同一个数据库，可使用主备模式或高可用数据库。具体配置如“单机部署模式”的“开启数据库配置”所示。

4.2.3 开启集群配置

每个ADCC节点服务器的配置文件目录conf下，拷贝 `cluster.conf.example` 为 `cluster.conf`：

操作示例如下：

```
cd $ADCC_HOME/config/
cp cluster.conf.example cluster.conf
```

将各个节点的IP和端口填写到 `cluster.conf`，每行配置成 `ip:port`。（配置3个或3个以上节点）

```
# ip:port
172.24.4.152:8848
172.24.4.154:8848
172.24.4.163:8848
```

4.2.4 开启负载均衡配置

以Nginx为例，配置负载均衡。

修改配置文件，`/${NGINX_HOME}/conf/nginx.conf`，修改其内容如下所示：

```
worker_processes 1;

events {
    worker_connections 1024;
}

#http转发
http {
    include mime.types;
    default_type application/octet-stream;

    sendfile on;

    keepalive_timeout 65;

    #adcc集群各节点ip:port
    upstream adcc-cluster {
        server 172.24.4.163:8848;
        server 172.24.4.152:8848;
        server 172.24.4.154:8848;
    }

    server {
        listen 8848;
        #负载均衡IP
        server_name 172.24.4.171;
```

```
location / {
    #root    html;
    #index  index.html index.htm;
    proxy_pass http://adcc-cluster;
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root    html;
}

}

}

#TCP转发
stream {
    upstream adcc-tcp{
        server 172.24.4.163:9848;
        server 172.24.4.152:9848;
        server 172.24.4.154:9848;
    }
    server {
        listen 9848;
        proxy_pass    adcc-tcp;
    }
}
}
```

5 启动

5.1 单机部署模式

以Linux为例，在 `${ADCC_HOME}/bin` 目录下，运行 `startup.sh -m standalone` 进行启动。

```
cd ADCC_HOME/bin
./startup.sh -m standalone
```

查看进程；或打开浏览器，输入 <http://ip:port/adcc/index.html> 进行访问；默认端口为8848 (http)；如果访问成功，则说明ADCC安装成功。

5.2 集群部署模式

以Linux为例，在各个节点 `${ADCC_HOME}/bin` 目录下，运行 `startup.sh` 进行启动。

```
cd ${ADCC_HOME}/bin
./startup.sh
```

启动负载均衡器Nginx。

查看进程；或打开浏览器，输入 <http://ip:port/adcc/index.html> 进行访问；其中ip为负载均衡器的IP，端口为负载均衡器的端口；如果访问成功，则说明ADCC安装成功。

6 停止

在 `$(ADCC_HOME)/bin` 目录下, 运行 `shutdown.sh` 停止ADCC。

```
cd ADCC_HOME/bin  
./shutdown.sh
```

7 卸载

删除ADCC安装部署目录。如安装部署目录 `${ADCC_HOME}` 为 `/opt/adcc`，将该目录删除即可。

8 云部署

ADCC支持云部署。

以下说明是一些基本的使用，实际使用的时候应该根据情况进行调整。

8.1 build docker镜像

8.1.1 build adcc镜像

1. 联系销售人员获取adcc产品包。
2. 确定license授权方式，adcc支持两种license授权方式：
 - 根据ip授权：传统的授权方式，一个ip地址一个授权文件。从销售人员处获取该授权文件，授权文件名为 `license.xml`。
 - 集中授权：不需要授权文件，但需要先搭建apusic授权中心，该中心的搭建可以咨询技术支持人员。然后修改Dockerfile，将 `APUSIC_ACLS_ENABLE` 设置为 `true`，添加 `APUSIC_ACLS_URLS` 参数并指向授权中心地址，注释或者删掉 `ADD license.xml license.xml` 那一行。

```

7 CLASSPATH="./home/apusic/adcc/conf:$CLASSPATH" \
8 CLUSTER_CONF="/home/apusic/adcc/conf/cluster.conf" \
9 FUNCTION_MODE="all" \
10 JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk" \
11 NACOS_USER="nacos" \
12 JAVA="/usr/lib/jvm/java-1.8.0-openjdk/bin/java" \
13 JVM_XMS="1g" \
14 JVM_XMX="1g" \
15 JVM_XMN="512m" \
16 JVM_MS="128m" \
17 JVM_MMS="320m" \
18 ADCC_DEBUG="n" \
19 ACCESSLOG_ENABLED="false" \
20 TIME_ZONE="Asia/Shanghai" \
21 APUSIC_ACLS_ENABLE="true"
22 APUSIC_ACLS_URLS=授权中心地址
23
24 ARG ADCC_VERSION=1.0.230
25 ARG HOT_FIX_FLAG=""
26
27 WORKDIR $BASE_DIR
28
29 RUN curl http://mirrors.aliyun.com/repo/Centos-7.repo > /etc/yum.repos.d/CentOS-Base.repo
30 RUN yum clean all && yum makecache
31
32 RUN set -x \
33     && yum update -y \
34     && yum install -y java-1.8.0-openjdk java-1.8.0-openjdk-devel iputils nc vim libcurl \
35     && yum clean all
36
37 COPY adcc-server-1.0.230.tar.gz /home/apusic
38 RUN tar -xzf /home/apusic/adcc-server-1.0.230.tar.gz -C /home/apusic \
39     && rm -rf /home/apusic/adcc-server-1.0.230.tar.gz \
40     && ln -snf /usr/share/zoneinfo/$TIME_ZONE /etc/localtime && echo $TIME_ZONE > /etc/timezone
41
42 #use license.xml
43 #ADD license.xml license.xml
44

```

3. 添加数据驱动。adcc自带mysql数据库驱动，如果需要使用其他数据库，则需要将驱动拷贝到 `images/adcc` 目录下，并在Dockerfile中添加一条指令，例如：

```
ADD kingbase8-8.2.0.jar target/ext/kingbase8-8.2.0.jar
```

4. 创建Dockerfile。下例为在X86操作系统中构建的Dockerfile，安装包名称需要根据实际情况修改。

```

FROM harbor.apusic.com/public/centos:7.5

# set environment
ENV MODE="standalone" \
    PREFER_HOST_MODE="ip" \
    BASE_DIR="/home/apusic/adcc" \
    CLASSPATH="./home/apusic/adcc/conf:$CLASSPATH" \

```

```

CLUSTER_CONF="/home/apusic/adcc/conf/cluster.conf" \
FUNCTION_MODE="all" \
JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk" \
NACOS_USER="nacos" \
JAVA="/usr/lib/jvm/java-1.8.0-openjdk/bin/java" \
JVM_XMS="1g" \
JVM_XMX="1g" \
JVM_XMN="512m" \
JVM_MS="128m" \
JVM_MMS="320m" \
ADCC_DEBUG="n" \
ACCESSLOG_ENABLED="false" \
TIME_ZONE="Asia/Shanghai" \
APUSIC_ACLS_ENABLE="false"

ARG ADCC_VERSION=1.0.243
ARG HOT_FIX_FLAG=""

WORKDIR $BASE_DIR

RUN curl http://mirrors.aliyun.com/repo/Centos-7.repo > /etc/yum.repos.d/CentOS-Base.repo
RUN yum clean all && yum makecache

RUN set -x \
    && yum update -y \
    && yum install -y java-1.8.0-openjdk java-1.8.0-openjdk-devel iputils nc vim libcurl \
    && yum clean all

COPY adcc-server-1.0.tar.gz /home/apusic
RUN tar -xzvf /home/apusic/adcc-server-1.0.tar.gz -C /home/apusic \
    && rm -rf /home/apusic/adcc-server-1.0.tar.gz \
    && ln -snf /usr/share/zoneinfo/$TIME_ZONE /etc/localtime && echo $TIME_ZONE > /etc/timezone

#use license.xml
ADD license.xml license.xml

#add database driver
#ADD xxx.jar target/ext/xxx.jar

ADD application.properties conf/application.properties

# set startup log dir
RUN mkdir -p logs \
    && touch logs/start.out \
    && ln -sf /dev/stdout start.out \
    && ln -sf /dev/stderr start.out
RUN chmod +x bin/docker-startup.sh
RUN chmod -c 755 plugins/peer-finder/*.sh plugins/peer-finder/peer-finder

EXPOSE 8848
ENTRYPOINT ["bin/docker-startup.sh"]

```

5. 然后执行 docker build 命令:

```
docker build -t adcc/adcc-server:1.0.243 .
```

6. 构建成功后, 通过 `docker images` 命令可以查看构建好的镜像。

7. 关于Dockerfile文件中环境变量的说明

- `BASE_DIR` 变量暂时不支持修改;
- 其他变量除了license相关的外也没必要直接修改Dockerfile文件, 而是在k8s或docker-compose的yaml文件中去指定, 这样可以保证构建的镜像具备一定的通用性。

8.1.2 build mysql镜像

如果需要使用mysql数据库, 可以进入 `images/mysql` 目录进行构建, 并根据需要修改Dockerfile; 也可以使用自己的mysql镜像, 或者搭建外部mysql服务器。

8.2 配置参数

参数	描述	参考值
SPRING_DATASOURCE_PLATFORM	数据库类型	embedded (默认值, 使用内置的derby), mysql, dm (达梦数据库), kingbase (人大金仓数据库), hgdb (瀚高数据库)
DB_URL	数据库连接URL	无
DB_USER	数据库连接用户名	无
DB_PASSWORD	数据库连接密码	无
DB_DRIVER_CLASSNAME	数据库连接驱动类	embedded和mysql数据库不需要填写, 其他数据库需要填写
ACCESSLOG_ENABLED	访问日志	默认值false
ADCC_AUTH_ENABLE	是否开启认证	默认false
ADCC_AUTH_IDENTITY_KEY	认证identity key	ADCC_AUTH_ENABLE为true时需要
ADCC_AUTH_IDENTITY_VALUE	认证identity value	ADCC_AUTH_ENABLE为true时需要
ADCC_AUTH_TOKEN	认证 token	ADCC_AUTH_ENABLE为true时需要
MODE	adcc启动模式	支持standalone和cluster
ADCC_SERVERS	ADCC集群实例列表	adcc集群模式需要设置, 格式为IP:端口, 多个实例用空格分隔。使用k8s-operator创建集群时不要设置。
JVM_XMS、JVM_XMX、JVM_XMN、JVM_MS、JVM_MMS	对应jvm相应参数	默认值分别为: 1g、1g、512m、128m、320m
APUSIC_ACLS_ENABLE	是否开启license集中授权	默认值false
APUSIC_ACLS_URLS	license集中授权服务器地址	APUSIC_ACLS_ENABLE为true时需要配置
APUSIC_ACLS_TENANT	租户	APUSIC_ACLS_ENABLE为true时才需要配置, 默认public
APUSIC_ACLS_NS	命名空间	APUSIC_ACLS_ENABLE为true时才需要配置, 默认public

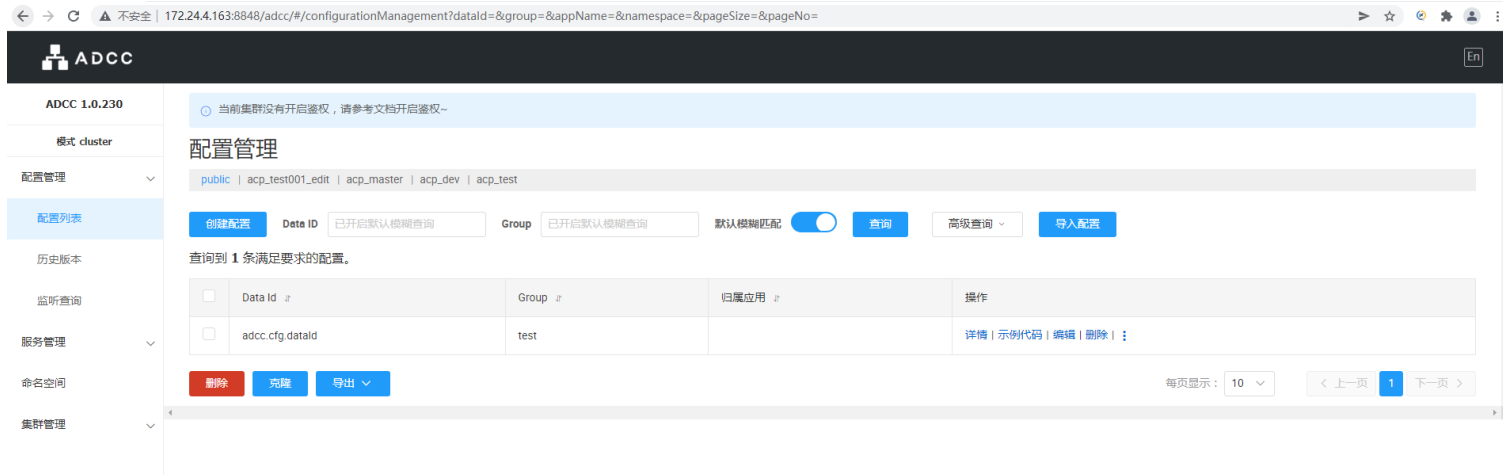
上述参数有的是在启动脚本 `docker-startup.sh` 中定义的, 有的是在adcc配置文件 `conf/application.properties` 中定义的。对于

`conf/application.properties` 中的相关参数, 上述表格如果无法满足需求, 可以自定义一份 `application.properties`, 替换 `images/adcc` 下的同名文件, 然后再开始构建。

9 其他操作

9.1 访问控制台

ADCC默认不开启鉴权，浏览器访问<http://IP:8848/adcc/>，无需登录即可访问控制台，进行ADCC的配置和服务管理。



9.2 开启鉴权

ADCC默认不开启鉴权，若开启鉴权，需要在使用用户名和密码登录之后，才能正常使用ADCC。默认的用户名为adcc，密码为apusic@2024。

操作示例如下，修改配置文件\${ADCC_HOME}/config/application.properties中的部分内容：

```
#兼容nacos鉴权
nacos.core.auth.system.type=nacos
#开启鉴权设置为true，关闭鉴权设置为false
nacos.core.auth.enabled=true

#设置apusic鉴权标签
nacos.core.auth.server.identity.key=apusic-key
nacos.core.auth.server.identity.value=apusic-value
#设置JWT令牌的密钥
nacos.core.auth.plugin.nacos.token.secret.key=U2VjcmV0S2V5YXB1c2l1jMTIzNDU2NzhhcHVzaWxMxMjM0NTY3ODkwMTIzNDU2Nz
```

【注意】鉴权开关是修改之后立马生效的，不需要重启ADCC服务端。

9.3 修改JWT令牌的密钥

ADCC提供生成JWT令牌的密钥的功能，进入ADCC的脚本目录 \${ADCC_HOME}/bin，运行 `startup.sh -b64` 生成密钥。

```
cd ADCC_HOME/bin

#SecretKey0123456789012345678901234567890123456789012345678901234567890123456789可自定义，其为SecretKey开头的Base64编码字符串，且原始密钥长度不得低于32字符，生成的密钥可替换application.properties中配置的默认密钥
nacos.core.auth.plugin.nacos.token.secret.key

./startup.sh -b64 SecretKey012345678901234567890123456789012345678901234567890123456789
```

9.4 修改密码

ADCC除了在控制台中提供修改默认密码的功能，也支持通过命令行的方式将明文密码生成加密字符串，以便直接初始化密码。

进入ADCC的脚本目录 `${ADCC_HOME}/bin` , 运行 `startup.sh -pw` 将明文密码转换成加密密码。

```
cd ADCC_HOME/bin

#passwordvalue为明文密码

./startup.sh -pw passwordvalue
```

在生成加密密码后, 可在启动初始化前修改 `${ADCC_HOME}/config/derby-schema.sql` 和 `derby-schema.sql` 下的sql语句, 具体操作如下:

```
INSERT INTO users (username, password, enabled) VALUES ('adcc', ''$2a$10$VGJgmj0Jo3SjX9z6fwVmNeFWYZi0yZF2j3FzKNez27N1pj7kSY4Ou'', TRUE);
INSERT INTO roles (username, role) VALUES ('adcc', 'ROLE_ADMIN');
```

其中adcc是用户名, 横线的是密码, 替换成生成的加密字符串。

全国统一服务热线
4008-555-800



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

Apusic
金蝶天燕

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

