AAMS V10

功能测试用例

|  |  |
| --- | --- |
| 编制：周惠媚 | 生效日期：2020-1-3 |
| 审核：王晶 | 批准：韦永森 |

深圳市金蝶天燕中间件股份有限公司对本文件资料享受著作权及其它专属权利，未经书面许可，不得将该等文件资料（其全部或任何部分）披露予任何第三方，或进行修改后使用。

|  |  |  |  |
| --- | --- | --- | --- |
| 文件更改摘要： | | | |
| 日期 | 版本号 | 修订说明 | 修订人 |
| 2020-1-3 | V1.0 | 新建 | 周惠媚 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 产品包目录检查 2](#_Toc14354)

[2. 产品安装与卸载 3](#_Toc954)

[2.1 产品安装 3](#_Toc3686)

[2.2 产品卸载 3](#_Toc5671)

[3. 启用和停止 5](#_Toc30091)

[3.1 产品启动 5](#_Toc27030)

[3.2 产品停止 5](#_Toc13372)

[4. 应用程序 7](#_Toc1960)

[4.1 部署应用程序 7](#_Toc15420)

[4.2 指定路径部署 7](#_Toc30505)

[5. 配置文件修改 9](#_Toc19669)

[5.1 设置JVM内存 9](#_Toc18214)

[5.2 修改apusic.conf 9](#_Toc31272)

[6. 与Springboot集成使用 11](#_Toc32632)

[6.1 AAMS核心jar包的注册 11](#_Toc18421)

[6.2 Starter的注册 13](#_Toc2787)

[6.3 使用AAMS作为Springboot应用的运行容器 13](#_Toc15316)

[7. 使用Apollo集中配置 16](#_Toc3842)

[8. 使用ETCD集中配置 17](#_Toc10732)

[9. 国密配置 19](#_Toc2179)

[9.1 单向认证 19](#_Toc22699)

[9.2 双向认证 20](#_Toc30360)

[10. 操作系统适配 23](#_Toc15127)

[10.1 Windows适配 23](#_Toc30329)

[10.2 Linux适配 23](#_Toc16033)

# 产品包目录检查

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 产品包目录检查 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试产品包目录检查 | | |
| **前置条件** | | 准备安装平台，如龙芯服务器、飞腾服务器等  准备zip格式的应用服务器安装文件  设置好系统的JDK环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 把zip格式的安装文件拷贝到硬盘的对应目录下，如/home目录下 | | 拷贝成功 | |
| 2 | 解压zip格式的文件 | | 执行成功，解压后的文件带有bin/、conf/、lib/、logs/、plugins/、webapps/、work/等 | |
| 3 | 分别检查各目录 | | 文件正常，无缺失、乱码等情况 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 产品安装与卸载

## 产品安装

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS-V10.1安装 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器安装 | | |
| **前置条件** | | 准备安装平台，如龙芯服务器、飞腾服务器等  准备zip格式的应用服务器安装文件  设置好系统的JDK环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 获取AAMS-V10.1.zip | | 成功获取 | |
| 2 | 解压zip格式的文件 | | 执行成功，解压后的文件带有bin、conf等目录 ，AAMS安装成功 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## 产品卸载

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS-V10.1卸载 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器安装和卸载 | | |
| **前置条件** | | 已安装AAMS | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 到安装目录下直接删除整个AAMS安装目录 | | 删除成功，将会从服务器中卸载AAMS | |
| 2 | 检查是否有剩余文件 | | 无遗留文件 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 启用和停止

## 产品启动

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS启动 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器启用 | | |
| **前置条件** | | 已成功安装Apusic应用服务器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 到安装目录下的bin目录，运行apusic.sh run  命令，启动应用服务器 | | 开始启动，加载完成后，应用服务器启动成功 | |
| 2 | 访问[http://ip:6888](https://ip:6848) | | 出现应用服务器首页，应用服务器可正常访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## 产品停止

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS停止 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器停止 | | |
| **前置条件** | | 已成功安装并运行Apusic应用服务器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 到安装目录下的bin目录，退出运行，停止应用服务器 | | 应用服务器成功停止运行 | |
| 2 | 访问[http://ip:6888](https://ip:6848) | | 无法访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS停止 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器停止 | | |
| **前置条件** | | 已成功安装并运行Apusic应用服务器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 到安装目录下的bin目录，执行stopapusic.sh，停止应用服务器 | | 应用服务器成功停止运行 | |
| 2 | 访问[http://ip:6888](https://ip:6848) | | 无法访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS停止 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器停止 | | |
| **前置条件** | | 已成功安装并运行Apusic应用服务器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 查看运行进程号，杀掉进程 | | 成功杀掉进行，应用服务器成功停止运行 | |
| 2 | 访问[http://ip:6888](https://ip:6848) | | 无法访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 应用程序

## 部署应用程序

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 部署应用程序测试 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器部署应用程序功能 | | |
| **前置条件** | | 已成功安装Apusic应用服务器  准备可用的应用程序test1.war包 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 把应用test1.war拷贝到 webapps 目录下 | | 成功自动部署应用程序 | |
| 2 | 接上一步，浏览器访问http://ip:6888/test1 | | 成功访问应用程序 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## 指定路径部署

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 指定路径部署 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器指定部署应用程序功能 | | |
| **前置条件** | | 已成功安装Apusic应用服务器  准备可用的应用程序test1.war包 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 把应用test1.war解压到某一目录下，如/opt/test1 | | 成功解压 | |
| 2 | 编辑apusic.conf，在host下新增<application path="/test1" docBase="D:\work\_test\test1" useHttpOnly="true"/>  保存 | | 成功编辑 | |
| 3 | 启动应用服务器 | | 成功启动应用服务器并部署应用程序 | |
| 4 | 接上一步，浏览器访问http://ip:6888/test1 | | 成功访问应用程序 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 配置文件修改

## 设置JVM内存

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 设置JVM内存 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器设置JVM内存功能 | | |
| **前置条件** | | 已分别在Windows/Linux成功安装Apusic应用服务器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | Windows 下修改 apusic.bat 文件，找到 MEMORY\_JVMOPTS 进行设置 | | 成功设置 | |
| 2 | 接上一步，启动应用服务器 | | 成功启动并成功修改JVM内存 | |
| 3 | Linux 下修改 apusic.sh 文件，找到 MEMORY\_JVMOPTS 进行设置 | | 成功设置 | |
| 4 | 接上一步，启动应用服务器 | | 成功启动并成功修改JVM内存 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## 修改apusic.conf

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 修改apusic.conf | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器修改配置文件功能 | | |
| **前置条件** | | 已成功安装Apusic应用服务器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 进入../conf/ | | 目录下有apusic.conf | |
| 2 | 接上一步，打开apusic.conf | | 成功打开 | |
| 3 | 接上一步，修改apusic.conf，如将endpoint下的port修改为5888，保存 | | 成功修改 | |
| 4 | 接上一步，启动应用服务器 | | 成功启动并成功修改配置文件 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 与Springboot集成使用

## AAMS核心jar包的注册

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | AAMS核心jar包的注册 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试AAMS核心jar包的注册 | | |
| **前置条件** | | 已成功安装Apusic应用服务器  获取AAMS-10.1.0 SP1-embed.zip  准备maven环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 解压AAMS-10.1.0 SP1-embed.zip到${ AAMS\_HOME }目录中 | | 成功解压，显示有${ AAMS\_HOME }\AAMS-embed\ | |
| 2 | 接上一步，安装各个软件包到本地库的命令如下（在当前目录打开终端，输入如下的命令）：  mvn install:install-file -Dfile=AAMS-embed-core.jar -DgroupId=com.apusic.ams.embed -DartifactId=AAMS-embed-core -Dversion=10.1 -Dpackaging=jar  mvn install:install-file -Dfile=AAMS-embed-el.jar -DgroupId=com.apusic.ams.embed -DartifactId=AAMS-embed-el -Dversion=10.1 -Dpackaging=jar  mvn install:install-file -Dfile=AAMS-embed-websocket.jar -DgroupId=com.apusic.ams.embed -DartifactId=AAMS-embed-websocket -Dversion=10.1 -Dpackaging=jar  mvn install:install-file -Dfile=AAMS-embed-jasper.jar -DgroupId=com.apusic.ams.embed -DartifactId=AAMS-embed-jasper -Dversion=10.1 -Dpackaging=jar  mvn install:install-file -Dfile=AAMS-dbcp.jar -DgroupId=com.apusic.ams.embed -DartifactId=AAMS-dbcp -Dversion=10.1 -Dpackaging=jar  mvn install:install-file -Dfile=AAMS-util.jar -DgroupId=com.apusic.ams.embed -DartifactId=AAMS-util -Dversion=10.1 -Dpackaging=jar  mvn install:install-file -Dfile=ecj-4.12.jar -DgroupId=com.apusic.ams.embed -DartifactId=ecj -Dversion=4.12 -Dpackaging=jar | | 成功安装 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## Starter的注册

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | Starter的注册 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试支持spring2.X的Starter的注册 | | |
| **前置条件** | | 已成功安装Apusic应用服务器  获取aams-spring-boot-starter.zip  准备maven环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 解压aams-spring-boot-starter.zip到${ AMS\_HOME }中 | | 成功解压，显示有${ AAMS\_HOME }\aams-spring-boot-starter\ | |
| 2 | 接上一步，mvn install:install-file -Dfile=aams-spring-boot-starter-2.1.7.RELEASE.jar -DgroupId=com.apusic -DartifactId=aams-spring-boot-starter -Dversion=2.1.7.RELEASE -Dpackaging=jar -DpomFile=pom.xml | | 成功安装 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## 使用AAMS作为Springboot应用的运行容器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 使用AAMS作为Springboot应用的运行容器 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试使用AAMS作为Springboot应用的运行容器 | | |
| **前置条件** | | 已成功安装Apusic应用服务器  准备应用示例chapter-1-spring-boot-quickstart  准备maven环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 在pom.xml中需要排除tomcat的starter：  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  <exclusions>  <exclusion>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-tomcat</artifactId>  </exclusion>  </exclusions>  </dependency> | | 成功设置 | |
| 2 | 接上一步，增加aams的starter依赖：  <dependency>  <groupId>com.apusic</groupId>  <artifactId>aams-spring-boot-starter</artifactId>  <version>2.1.7.RELEASE</version>  </dependency> | | 成功添加 | |
| 3 | 接上一步，进入到工程目录下，并打开终端，并输入mvn package命令进行构建 | | 成功构建 | |
| 4 | 接上一步，切换到构建后的target目录，在该目录下打开终端，并运行  java –jar chapter-1-spring-boot-quickstart-1.0.jar | | 成功启动 | |
| 5 | 在浏览器中输入：[http://localhost:6888/hello](http://localhost:8080/hello) | | 成功访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 使用Apollo集中配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 使用Apollo集中配置 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器使用Apollo软件进行集中配置 | | |
| **前置条件** | |  | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 在apusic.conf配置文件增加plugins的配置：  <plugins>  <plugin appId="10086" className="com.apusic.ams.config.listener.ApolloPluginLoader" cluster="sz-test" metaUrl="http://172.20.129.231:8080" namespace="test-namespace"/>  </plugins> | | 成功配置 | |
| 2 | 进入Apollo，设置key为apusic\_config，value为要改变的值进行修改 | | 成功设置 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 使用ETCD集中配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 使用ETCD集中配置 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器使用ETCD进行集中配置 | | |
| **前置条件** | |  | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 在apusic.conf配置文件增加plugins的配置：  要支持ETCD2，则需要指定实现为：  <plugins>  <plugin className="com.apusic.ams.config.listener.EtcdPluginLoader" clusterId="cluster" host="172.18.100.74" port="2379" namespace="test"/>  </plugins>  要支持ETCD3则需要指定为：  <plugins>  <plugin className="com.apusic.ams.config.listener.EtcdV3PluginLoader" clusterId="cluster" host="172.18.100.74" port="2379" namespace="test"/>  </plugins> | | 成功配置 | |
| 2 | 进入Apollo，设置key为apusic\_config，value为要改变的值进行修改 | | 成功设置 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 国密配置

## 单向认证

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 单向认证 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器支持国密单向认证 | | |
| **前置条件** | | 准备JDK8版本  准备360国密浏览器 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 修改apusic.conf文件：  <endpoint name="https" port="6887" maxThreads="150" SSLEnabled="true">  <SSLHostConfig sslProtocol="SMv1.1" protocols="SMv1.1" >  <Certificate certificateKeystoreFile="conf/test016.pfx" certificateKeystorePassword="123456" certificateKeyPassword="123456"  type="EC" certificateKeystoreType="PKCS12"/>  </SSLHostConfig>  </endpoint> | | 成功修改 | |
| 2 | 接上一步，增加endpoint到server：  <server name="ams-server" services="ams-service" endpoints="http,https" executors="http-thread-pool"/> | | 成功添加 | |
| 3 | 接上一步，启动AAMS | |  | |
|  | 开启国密浏览器的支持国密功能 | | 成功开启 | |
| 4 | 在C:\Users\用户名\AppData\Roaming\360se6\User Data\Default\gmssl(该路径中的360se6目录可能有所变化如360se或360ent等)下创建ctl目录，然后在ctl目录下创建ctl.dat文件，并将根证书内容拷入该文件，然后重启浏览器 | | 成功设置 | |
| 5 | 接上一步，国密浏览器访问应用程序 https://ip:6888/ | | 成功访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## 双向认证

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 双向认证 | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器支持国密双向认证 | | |
| **前置条件** | | 准备JDK8版本  准备360国密浏览器 | | |
| **测试执行步骤** | | | | |
| 序号 | 测试输入 | | 预期结果 | |
| 1 | 修改apusic.conf文件：  <endpoint name="https" port="6887" maxThreads="150" SSLEnabled="true">  <SSLHostConfig sslProtocol="SMv1.1" protocols="SMv1.1" truststoreFile="conf/truststore.jks" truststorePassword="123456" certificateVerification="want" truststoreType="JKS">  <Certificate certificateKeystoreFile="conf/keystore.p12" certificateKeystorePassword="123456" certificateKeyPassword="123456" type="EC" certificateKeystoreType="PKCS12"/>  </endpoint> | | 成功修改 | |
| 2 | 接上一步，增加endpoint到server：  <server name="ams-server" services="ams-service" endpoints="http,https" executors="http-thread-pool"/> | | 成功添加 | |
| 3 | 接上一步，启动AAMS | |  | |
| 4 | 开启国密浏览器的支持国密功能 | | 成功开启 | |
| 5 | 在C:\Users\用户名\AppData\Roaming\360se6\User Data\Default\gmssl(该路径中的360se6目录可能有所变化如360se或360ent等)下创建ctl目录，然后在ctl目录下创建ctl.dat文件，并将根证书内容拷入该文件，然后重启浏览器 | | 成功设置 | |
| 6 | 配置客户端证书 | | 成功配置 | |
| 7 | 接上一步，国密浏览器访问应用程序 https://ip:6888/ | | 成功访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

# 操作系统适配

## Windows适配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 操作系统适配测试-Windows | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器与操作系统兼容 | | |
| **前置条件** | | 准备安装平台win10（172.20.140.15）  准备安装文件  已各有准备JDK环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 把zip格式的安装文件拷贝到硬盘的对应目录下 | | 拷贝成功 | |
| 2 | 解压zip格式的文件 | | 执行成功，解压后的文件带有bin、conf、lib、logs/、plugins、webapps、work等 | |
| 3 | 到安装目录下的bin目录，运行apusic.sh run  命令，启动应用服务器 | | 开始启动，加载完成后，应用服务器启动成功 | |
| 4 | 访问[http://ip:6888](https://ip:6848) | | 出现应用服务器首页，应用服务器可正常访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |

## Linux适配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | | 操作系统适配测试-Linux | **测试用例标识** |  |
| **测试用例说明（方法、目的）** | | 测试应用服务器与操作系统兼容 | | |
| **前置条件** | | 准备安装平台X86（172.20.140.41）、银河麒麟（172.20.140.77）、天玥龙芯（172.20.140.78）  准备安装文件  已各有准备JDK环境 | | |
| **测试执行步骤** | | | | |
| **序号** | **测试输入** | | **预期结果** | |
| 1 | 把zip格式的安装文件拷贝到硬盘的对应目录下，如/home目录下 | | 拷贝成功 | |
| 2 | 解压zip格式的文件 | | 执行成功，解压后的文件带有bin/、conf/、lib/、logs/、plugins/、webapps/、work/等 | |
| 3 | 到安装目录下的bin目录，运行apusic.sh run  命令，启动应用服务器 | | 开始启动，加载完成后，应用服务器启动成功 | |
| 4 | 访问[http://ip:6888](https://ip:6848) | | 出现应用服务器首页，应用服务器可正常访问 | |
| 测试结果 | | |  | |
| 参测人员签字 | | |  | |