



APUSIC  
固若长城  
睿比世界

# 用户手册

金蝶 Apusic 迁移工具

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

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

## 免责声明

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

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

## 商标声明

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

# 目录

- 1 前言
  - 1.1 面向对象
- 2 版本说明
- 3 产品简介
  - 3.1 概述
  - 3.2 平台环境
  - 3.3 系统环境
  - 3.4 术语
- 4 产品清单
- 5 使用向导
  - 5.1 安装
  - 5.2 配置说明
  - 5.3 命令说明
    - 5.3.1 migrate analyse
    - 5.3.2 migrate install
    - 5.3.3 migrate rollback
  - 5.4 卸载
- 6 独立应用迁移
  - 6.1 应用分析
  - 6.2 应用部署
  - 6.3 AAS备份恢复
  - 6.4 非本机迁移
  - 6.5 迁移格式
- 7 WebLogic 应用迁移
  - 7.1 WebLogic管控台
  - 7.2 数据源迁移
  - 7.3 迁移分析
  - 7.4 迁移至AASV10
  - 7.5 迁移至AASV9
  - 7.6 EJB迁移
    - 7.6.1 配置迁移分析目录
    - 7.6.2 迁移分析

- 7.6.3 安装至AASV10
- 7.6.4 安装至AASV9
- 7.7 迁移回退
- 7.8 问题解答
  - 7.8.1 数据源AES解密异常
  - 7.8.2 安装重名
- 8 Tomcat迁移
  - 8.1 支持迁移的功能
  - 8.2 Tomcat 中的示例应用和示例配置
  - 8.3 迁移至AAS10
    - 8.3.1 安装迁移工具包
    - 8.3.2 配置分析
    - 8.3.3 生成分析文件
    - 8.3.4 安装解析好的配置至AAS10
    - 8.3.5 重启AAS10 查看应用是否迁移成功
    - 8.3.6 查看已部署的应用程序
    - 8.3.7 查看已迁移的 JDBC Resources
    - 8.3.8 查看已迁移的JavaMail Sessions
    - 8.3.9 回滚迁移至AAS10的功能
  - 8.4 迁移至AAS9
    - 8.4.1 配置解析以及安装
    - 8.4.2 重启AAS9,查看应用是否迁移成功
    - 8.4.3 查看已部署的应用程序
    - 8.4.4 查看已迁移的 JDBC Resources
    - 8.4.5 查看已迁移的JavaMail Sessions
    - 8.4.6 回滚迁移至AAS9的功能
  - 8.5 类冲突检测
  - 8.6 Jar安全漏洞检测
  - 8.7 厂商依赖检测
  - 8.8 常见问题解答
- 9 WebSphere 迁移至AAS
  - 9.1 管控台配置数据源
    - 9.1.1 数据源迁移
    - 9.1.2 迁移分析

- 9.1.3 迁移至AASV10
- 9.2 EJB迁移
  - 9.2.1 迁移分析
  - 9.2.2 迁移至AASV10
  - 9.2.3 迁移至AASV9
- 9.3 迁移回退
- 10 AASV9 迁移至AASV10

# 1 前言

本文档为金蝶 Apusic 迁移工具使用说明，详细介绍金蝶 Apusic 迁移工具相关配置方法。

## 1.1 面向对象

本手册主要面向对象为使用金蝶 Apusic 迁移工具进行资源迁移的相关管理人员和运维人员。

## 2 版本说明

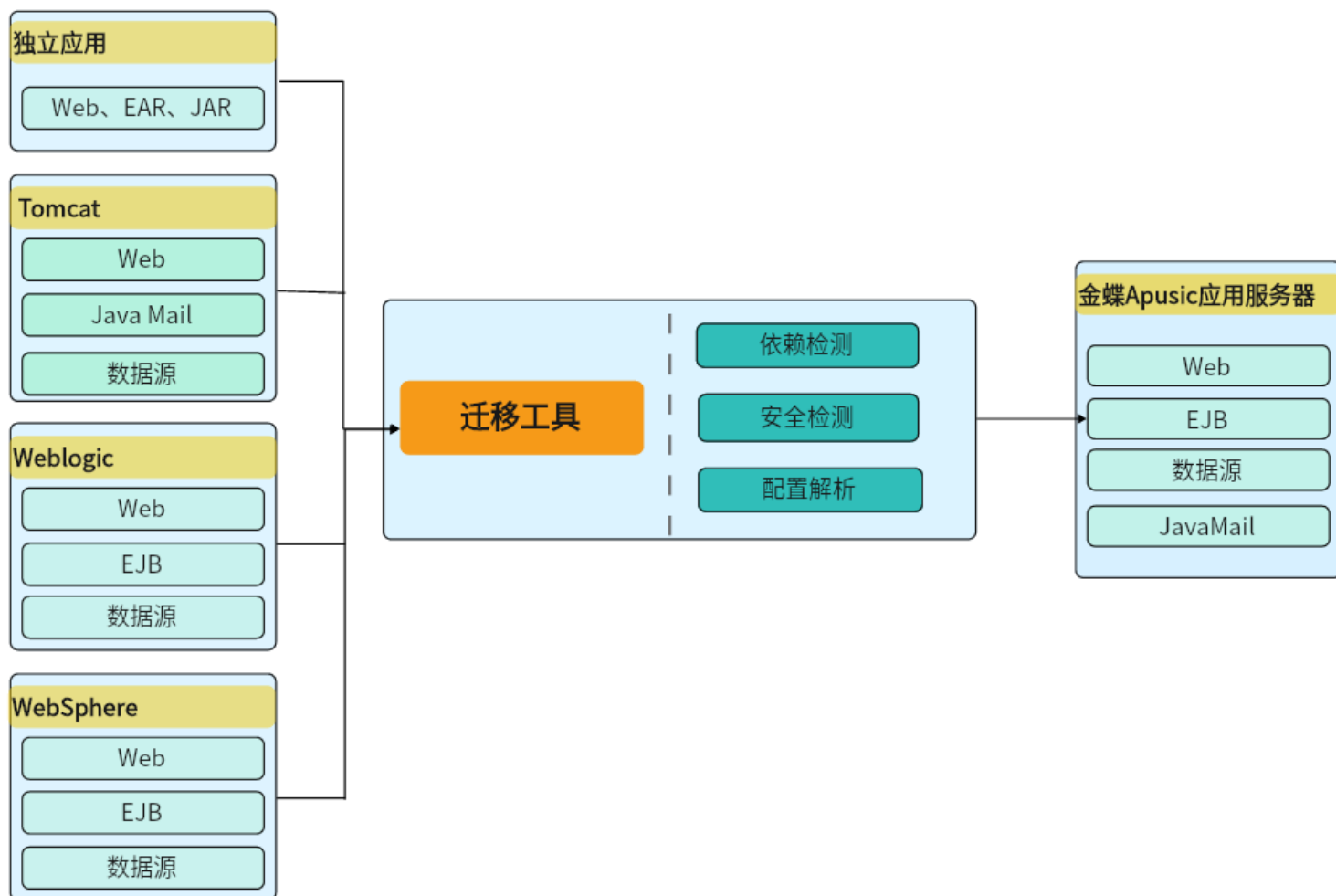
本文档为金蝶 Apusic 迁移工具使用手册，根据工具版本更新情况累积修订本文档内容。

日期	手册版本	适用产品	更新说明
2021年12月	V01E01F01	Migration-AAS V1.0	介绍使用说明

## 3 产品简介

### 3.1 概述

金蝶Apusic迁移工具版本1.0（简称“Migration-AAS V1.0”）使用Java技术开发，实现应用/配置分析、依赖检测、迁移到AAS。该工具整体由解析工具和安装工具组成，解析分析工具对目标服务器的WEB、EJB、数据源解析分析，依赖检测；安装工具把解析分析工具的输出安装到AAS服务器。



### 3.2 平台环境

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

### 3.3 系统环境

系统组件	系统要求
Java环境	JDK8
内存	至少需要 512MB 的内存
磁盘空间	至少需要 1024MB 的磁盘空间

### 3.4 术语

Migration-AAS: 金蝶Apusic迁移工具

AAS: 金蝶Apusic应用服务器

## 4 产品清单

下表说明产品资源，其中 `${Migration-AAS_HOME}` 表示金蝶Apusic迁移工具安装目录。

目录资源	说明
<code>\${Migration-AAS_HOME}/applications</code>	需要安装的独立应用程序包，如jar、war、ear包
<code>\${Migration-AAS_HOME}/bin</code>	迁移工具启动脚本（Linux平台是migrate，Windows平台是migrate.cmd）
<code>\${Migration-AAS_HOME}/classes</code>	迁移工具Java字节码文件目录
<code>\${Migration-AAS_HOME}/common</code>	迁移工具公共资源目录
<code>\${Migration-AAS_HOME}/conf</code>	迁移工具配置conf目录
<code>\${Migration-AAS_HOME}/lib</code>	迁移工具依赖jar目录
<code>\${Migration-AAS_HOME}/log</code>	工具log文件目录
<code>\${Migration-AAS_HOME}/out</code>	运行迁移工具后保存迁移的应用、配置、jar包安全分析结果等
<code>\${Migration-AAS_HOME}/temp</code>	运行迁移工具后迁移数据源、war、ejb应用生产的临时目录
<code>\${Migration-AAS_HOME}/backup</code>	迁移应用、配置到AAS应用服务器时生成的备份AAS相关应用、配置的备份目录

## 5 使用向导

下面介绍 金蝶Apusic迁移工具作为一个完整的产品进行使用, `${Migration-AAS_HOME}` 表示金蝶Apusic迁移工具安装目录。

### 5.1 安装

金蝶Apusic迁移工具支持在Linux、Windows等操作系统上安装, 需要准备JDK8环境。

Windows、Linux系统下压缩包直接解压到相关目录即可, 以Linux操作系统为例。

终端解压安装包: `unzip migration.zip` 解压成功后进入安装目录, 其目录结构如下:

```
[root@localhost migration]# ls -l
total 0
drwxr-xr-x. 2 root root  6 Jul 16 16:23 applications
drwxr-xr-x. 2 root root 40 Jul 16 09:52 bin
drwxr-xr-x. 3 root root 17 Jul 16 09:52 classes
drwxr-xr-x. 3 root root 30 Jul 16 09:52 common
drwxr-xr-x. 2 root root 110 Jul 16 09:52 conf
drwxr-xr-x. 2 root root  45 Jul 16 09:52 lib
```

### 5.2 配置说明

迁移工具配置文件为 `conf/conf.properties`, 内部配置说明如下:

参数	说明	默认值
server_home	待迁移应用服务器的安装目录。若迁移独立应用只需把独立应用放置到applications目录即可, 不用配置该项	D:/develop/wls1036
install_home	迁移安装的AASV9/AASV10服务器安装目录	D:/develop/aas/AAS-V10
wls.domainPath	WebLogic 域路径名,不指定将	D:/develop/wls1036/user_projects/domains/base_domain

	解析所有域， 从Weblogic迁移应用或其他资源时才需要配置	
security.checker.update.enable	安全检测更新，启用安全检测更新则在安全检测时会联网校验	false



## 5.3 命令说明

进入安装目录并运行对应平台下的migrate，如Linux执行 `sh bin/migrate`，效果如下。

```
[root@localhost migration]# sh bin/migrate
[WARN] 使用说明:
migrate [options]

选项:
help          打印帮助命令说明
analyse      应用配置解析
install      应用配置迁移至Apusic
rollback     还原迁移
dependent-testing 厂商依赖检测
security-scan jar安全漏洞检测
class-conflict 类冲突检测

[root@localhost migration]#
```

### 5.3.1 migrate analyse

对待迁移服务器或独立应用目录(applications)的应用进行分析，生成迁移应用、迁移配置、依赖检测和安全检测至out目录。

### 5.3.2 migrate install

迁移生成的out目录配置、应用到AAS服务器，`conf/conf.properties` 配置文件需要配置 `install_home` 配置项指定AAS服务器安装目录。安装时还会生成backup目录，其内保存AAS服务器对应的备份文件，在迁移失败时使用保存的backup文件中的备份文件恢复AAS服务器配置和应用信息。

### 5.3.3 migrate rollback

AAS配置、应用回退会使用backup目录中离当前最近的备份恢复AAS服务器。

## 5.4 卸载

进入金蝶Apusic迁移工具安装目录，将整个目录删除即可卸载该工具。

## 6 独立应用迁移

金蝶Apusic迁移工具支持将应用包从工具对应目录中迁移到金蝶Apusic应用服务器。

### 6.1 应用分析

把独立应用war、jar、ear包放置于applications目录，如 web.war 。

```
[root@localhost migration]# ls applications/
web.war
[root@localhost migration]#
```

配置 conf/conf.properties 的 install\_home 配置项为金蝶Apusic应用服务器安装目录，配置项 server\_home 不用配置。

```
#Apusic服务器安装目录
install_home=/opt/aas/ApusicAS
```

执行 sh bin/migrate analyse 生成输出配置、应用目录out。

```
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-07-20 15:07:32 WARN ParserFactory:61 - 未找到server_home服务器配置信息或配置有误
2021-07-20 15:07:32 INFO AppsParser:29 - 解析applications目录中的独立应用
2021-07-20 15:07:32 INFO JarSecurity:19 - class文件安全检测
[INFO]
Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.
```

out目录结构如下:

- 20210720163731: 按时间顺序生成的虚拟域目录，如果解析的为WebLogic则为WebLogic的域名称
- applications: 待迁移的应用程序包目录
- class-conflict: 类冲突检测的输出目录
- config: 迁移应用解析后的配置文件目录
- dependency-check: 安全检测输出目录
- manufacturer-dependent-testing: 厂商依赖检测输出目录

```

out
├── domains
│   └── 20210720163731
│       ├── applications
│       │   └── web.war
│       ├── class-conflict
│       │   └── class-conflict.json
│       ├── config
│       │   └── domain.xml
│       ├── dependency-check
│       │   └── dependency-check-report.html
│       └── manufacturer-dependent-testing
│           └── manufacturer-dependent-testing.json

```

## 6.2 应用部署

执行 `sh bin/migrate install` 将应用部署到金蝶Apusic应用服务器。

```

[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-20 15:51:34 INFO  AnsApp:94 - 执行AAS本地迁移工具
2021-07-20 15:51:34 INFO  ConflictClass:43 - 类冲突检测
2021-07-20 15:51:37 INFO  AasV10Parser:39 - 开始应用迁移
@@@@应用迁移成功, 请重启AAS 服务器!@@@@
[root@localhost migration]#

```

执行完成后，迁移工具安装目录有 `/backup` 目录，backup目录结构按时间命名文件夹保存AAS服务器相关备份配置、应用。

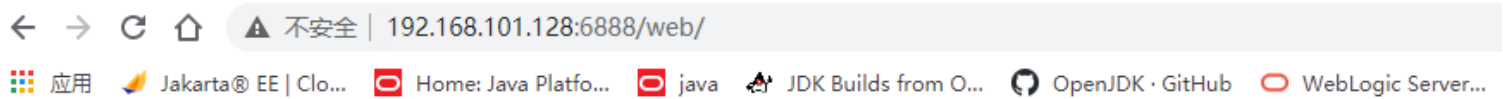
```

[root@localhost migration]# tree -C -L 1 backup/
backup/
├── 20210716195816
└── 20210719150621

2 directories, 0 files
[root@localhost migration]#

```

部署完成后，启动金蝶Apusic应用服务器并访问迁移的应用。如果金蝶Apusic应用服务器处于运行状态，需要重启。



## header

```
javax.servlet.jsp.el.ImplicitObjectELResolver$ImplicitObjects$7@735efb02
javax.servlet.jsp.el.ImplicitObjectELResolver$ImplicitObjects$8@110b3020
```

**server local addr: 192.168.101.128**

## session

```
javax.servlet.jsp.el.ImplicitObjectELResolver$ImplicitObjects$3@7ec77601
org.apache.catalina.session.StandardSessionFacade@1d9de49e
```

## 6.3 AAS备份恢复

AAS配置/应用回退: 执行 `sh bin/migrate rollback` 会使用backup目录中离当前最近的备份恢复AAS服务器。

```
[root@localhost migration]# sh bin/migrate rollback
[INFO] 设置路径/opt/migration/migration/
2021-07-20 16:57:23 INFO AasV10Parser:127 - =====从备份文件backup/20210719150621恢复=====
2021-07-20 16:57:23 INFO AasV10Parser:133 - =====备份文件恢复成功=====
[root@localhost migration]#
```

重启AAS服务器并登录查看管理控制台已无迁移的应用。

## 6.4 非本机迁移

迁移工具支持非本机迁移(即AAS服务器安装目录与分析输出不在同一台机器), 只需要把分析输出out目录拷贝至AAS目标机器迁移工具安装目录。

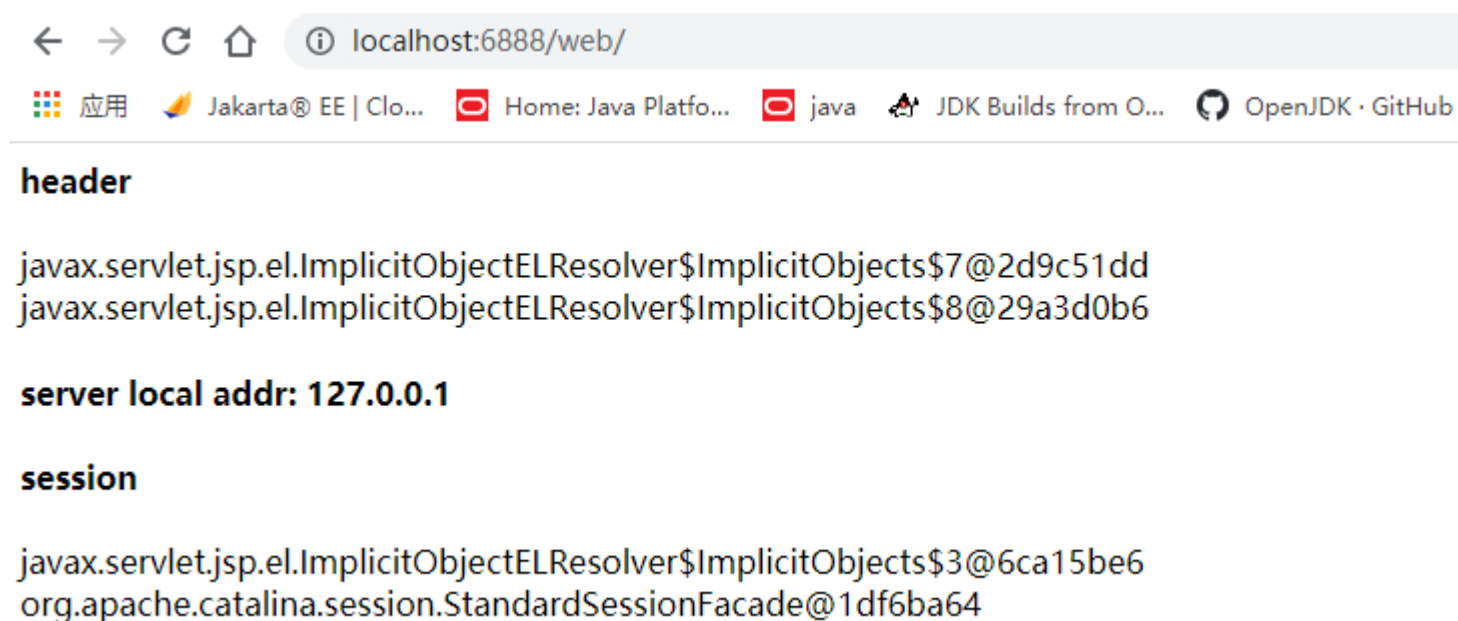
此电脑 > 桌面 > tmp > migration

名称	修改日期	类型	大小
applications	2021/7/20 9:11	文件夹	
bin	2021/7/20 17:11	文件夹	
classes	2021/7/20 17:11	文件夹	
common	2021/7/20 17:11	文件夹	
conf	2021/7/20 17:12	文件夹	
lib	2021/7/20 17:12	文件夹	
out	2021/7/20 17:12	文件夹	

执行应用迁移。

```
C:\Users\kingdee.DESKTOP-RCPNUKG\Desktop\tmp\migration>bin\migrate.cmd install
JAVA_HOME: D:\develop\Java\jdk1.8.0_281
[INFO] 设置路径C:\Users\kingdee.DESKTOP-RCPNUKG\Desktop\tmp\migration\bin\..\
2021-07-20 17:12:57 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-20 17:12:57 INFO ConflictClass:43 - 类冲突检测
2021-07-20 17:13:01 INFO AasV10Parser:39 - 开始应用迁移
@@@@应用迁移成功, 请重启AAS 服务器!@@@@
C:\Users\kingdee.DESKTOP-RCPNUKG\Desktop\tmp\migration>
```

启动AAS服务器并登录管控台。访问非本机迁移的应用。



The screenshot shows a web browser window with the address bar set to `localhost:6888/web/`. The browser tabs include '应用', 'Jakarta® EE | Clo...', 'Home: Java Platfo...', 'java', 'JDK Builds from O...', and 'OpenJDK · GitHub'. The page content displays the following information:

```
header
javax.servlet.jsp.el.ImplicitObjectELResolver$ImplicitObjects$7@2d9c51dd
javax.servlet.jsp.el.ImplicitObjectELResolver$ImplicitObjects$8@29a3d0b6

server local addr: 127.0.0.1

session
javax.servlet.jsp.el.ImplicitObjectELResolver$ImplicitObjects$3@6ca15be6
org.apache.catalina.session.StandardSessionFacade@1df6ba64
```

## 6.5 迁移格式

迁移工具对配置项 `server_home/` 独立应用的解析输出配置为AASV10格式的配置文件，在安装到目标AAS服务器时会自动判断目标平台，生成最终对应平台的配置文件信息，最终的配置、应用在temp目录。

如对独立应用web.war生成的信息如下，out目录子目录 `config/domain.xml` 就是AASV10格式配置文件。

```
[root@localhost migration]# tree -C out
out
├── domains
│   └── 20210721142006
│       ├── applications
│       │   └── web.war
│       ├── config
│       │   └── domain.xml
│       ├── dependency-check
│       │   └── dependency-check-report.html
│       └── manufacturer-dependent-testing
│           └── manufacturer-dependent-testing.json
6 directories, 4 files
[root@localhost migration]#
```

```
[root@localhost migration]# cat out/domains/20210721142006/config/domain.xml
<?xml version="1.0" encoding="UTF-8"?>
<domain>
  <resources></resources>
  <servers>
    <server name="server">
      <application-ref ref="web" virtual-servers="server"/>
    </server>
  </servers>
  <configs>
    <config name="server-config"></config>
  </configs>
  <applications>
    <application name="web" context-root="/web" location="${com.apusic.aas.instanceRootURI}/applications/web" object-type="user">
      <property name="archiveType" value="war"/>
      <property name="appLocation" value="${com.apusic.aas.instanceRootURI}/applications/__internal/web/web.war"/>
      <property name="defaultAppName" value="web"/>
      <property name="secretLevel" value="SECERT"/>
      <module name="web">
        <engine sniffer="security"/>
        <engine sniffer="web"/>
      </module>
    </application>
  </applications>
</domain>
[root@localhost migration]#
```

如迁移配置项为AASV9, `install_home=/opt/aas/aasv9` , 则运行 `sh bin/migrate install` 迁移应用至AASV9过程中会在temp目录下生成最终AASV9的信息输出。

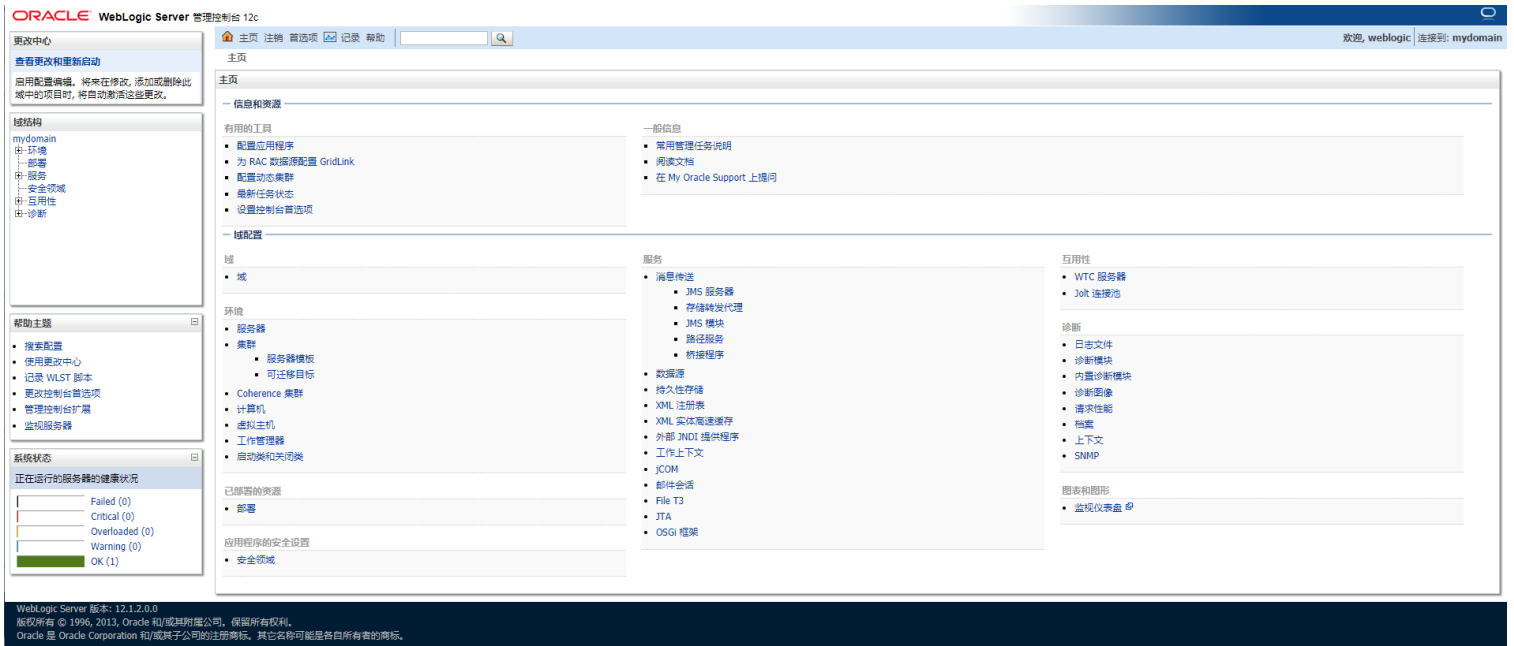
```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-21 14:28:36 INFO  AnsApp:94 - 执行AAS本地迁移工具
2021-07-21 14:28:36 INFO  ConflictClass:43 - 类冲突检测
2021-07-21 14:28:38 INFO  AasV9Parser:32 - 开始应用迁移
2021-07-21 14:28:38 INFO  AasV9Parser:108 - @@@@应用迁移成功, 请重启AAS 服务器!@@@@
[root@localhost migration]# tree -C temp
temp
├── config
│   ├── datasources.xml
│   ├── jndi_resource.xml
│   └── server.xml
├── upload
│   └── web
│       └── web.war
3 directories, 4 files
[root@localhost migration]#
```

## 7 WebLogic 应用迁移

WebLogic 是美国 Oracle 公司的一款基于 JAVAEE 架构的中间件产品，用于开发、部署、管理Web应用、EAR应用、EJB应用。

### 7.1 WebLogic管控台

浏览器打开：<http://localhost:7001/console> 并登录



### 7.2 数据源迁移

金蝶迁移工具可将WebLogic的数据源信息迁移到AAS。通过WebLogic 管控台配置数据源。如已配置有数据源信息，可忽略。



配置 `conf/conf.properties` 的配置项 `server_home` 为WebLogic安装目录。

```
[root@localhost migration]# cat conf/conf.properties
#####注意：windows系统路径需要把\转换为/

#=====Web/EJB/DataSource transfer config
#WebLogic/Tomcat/AASV9 安装目录
server_home=/opt/weblogic/wls12130

#Apusic服务器安装目录
#install_home=D:/develop/aas/AAS-V10

#WebLogic 域路径名,不指定将解析所有域
#wls.domainPath=D:/develop/wls1036/user_projects/domains/base_domain

#安全检测更新,启用安全检测更新则在安全检测时会联网校验
security.checker.update.enable=false
```

## 7.3 迁移分析

执行 `sh migrate analyse` 生成分析输出信息，数据源配置的JDBC驱动会被自动扫描并保存到out相关子目录lib下，在迁移安装时会被拷贝到AAS。

WebLogic 迁移数据源会执行WebLogic的相关脚本以解密数据源AES密文，执行脚本时如出现如下错误需要配置

MW\_HOME 环境变量为WebLogic安装目录，可在终端中零时配置环境变量 `export MW_HOME=xxxxxxx`。

```
[root@localhost migration]# unset MW_HOME
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-07-19 15:41:59 INFO AppsParser:29 - 解析applications目录中的独立应用
/opt/weblogic/wls12130/wlserver/server/bin/setWLSEnv.sh: line 41: /wlserver/common/bin/commEnv.sh: No such file or directory
The WebLogic Server wasn't found in directory /wlserver/server.
Please edit the startWebLogic.sh script so that the WL_HOME
variable points to the WebLogic installation directory.
Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/security/internal/SerializedSystemIni
    at com.apusic.util.AesDecrypt.main(AesDecrypt.java:17)
Caused by: java.lang.ClassNotFoundException: weblogic.security.internal.SerializedSystemIni
    at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:355)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
    ... 1 more
2021-07-19 15:42:00 ERROR WlsParser:125 - 数据源密码AES解密失败
```

```
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-07-16 18:27:02 WARN ParserFactory:32 - 未找到install_home服务器配置信息
2021-07-16 18:27:02 INFO AppsParser:29 - 解析applications目录中的独立应用
CLASSPATH=/opt/tools/jdk1.8.0_281/lib/tools.jar:/opt/weblogic/wls12130/wlserver/server/lib/weblogic_sp.jar:/opt/weblogic/wls12130/wlserver/server/lib/weblogic.jar:/opt/weblog
ic/wls12130/oracle_common/modules/net.sf.antcontrib_1.1.0.0_1-0b3/lib/ant-contrib.jar:/opt/weblogic/wls12130/wlserver/modules/features/oracle.wls.common.nodemanager_2.0.0.0.j
ar:
PATH=/opt/weblogic/wls12130/wlserver/server/bin:/opt/weblogic/wls12130/oracle_common/modules/org.apache.ant_1.9.2/bin:/opt/tools/jdk1.8.0_281/jre/bin:/opt/tools/jdk1.8.0_281/
bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/opt/tools/jdk1.8.0_281/bin:/opt/GmSSL/bin:/root/bin:/opt/weblogic/wls12130/oracle_common/modules/org.apache.maven_3.0.5
/bin
Your environment has been set.
DECRYPTION_FLAG=root
2021-07-16 18:27:04 INFO JarSecurity:19 - class文件安全检测
2021-07-16 18:27:04 INFO VendorClass:28 - 厂商依赖检测
```

执行成功后生成输出信息目录。

```
[root@localhost migration]# ls
applications bin classes common conf lib logs out
[root@localhost migration]#
```

## 7.4 迁移至AASV10

执行迁移安装时会自动把JDBC驱动包拷贝到AAS服务器。 `conf/conf.properties` 配置 `install_home` 为 AASV10安装目录。

```
[root@localhost migration]# cat conf/conf.properties
#####注意：windows系统路径需要把\转换为/

#=====Web/EJB/DataSource transfer config
#WebLogic/Tomcat/AASV9 安装目录
server_home=/opt/weblogic/wls12130

#Apusic服务器安装目录
install_home=/opt/ApusicAS
```

执行 `sh bin/migrate install` 迁移数据源至AASV10。

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-16 19:58:13 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-16 19:58:13 INFO ConflictClass:43 - 类冲突检测
2021-07-16 19:58:16 INFO AasV10Parser:39 - 开始应用迁移
2021-07-16 19:58:16 INFO AasV10Parser:101 - mysql-connector-java-commercial-5.1.22-bin.jar文件已经存在
@@@应用迁移成功，请重启AAS服务器!@@@
[root@localhost migration]#
```

重启AAS服务器并登录管控台，测试数据源迁移效果如下图。

### JDBC 连接池

为了存储、组织和检索数据，大多数应用程序都采用了关系数据库。Java EE 应用程序通过 JDBC API 访问关系数据库。应用程序必须获得一个连接之后才可以访问数据库。

刷新时间间隔 (秒) :

连接池 (3)

选择	连接池名称	资源类型	类名称	状态
<input type="checkbox"/>	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	未连通
<input type="checkbox"/>	__TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource	已连通
<input type="checkbox"/>	jdbcds0	java.sql.Driver	com.mysql.jdbc.Driver	已连通

## JDBC 资源

JDBC 资源为应用程序提供连接数据库的方法。

资源 (3)

选择	JNDI 名称	Logical JNDI Name	已启用	连接池	说明
<input type="checkbox"/>	jdbc/TimerPool		✓	TimerPool	
<input type="checkbox"/>	jdbc/default	java:comp/DefaultDataSource	✓	DerbyPool	
<input type="checkbox"/>	jbcds0		✓	jbcds0	

AAS管控制台测试数据源迁移效果。

一般信息 高级 其他属性

ping成功

### 编辑 JDBC 连接池

修改现有的 JDBC 连接池。连接池是用于特定数据库的一组可重复使用的连接。

加载默认值 刷新 ping

保存 返回

执行成功

### 编辑 JDBC 资源

编辑现有的 JDBC 数据源。

加载默认值

JNDI 名称: jdbcds0

密码: 秘密

连接池名称: jdbcds0  
使用 JDBC 连接池页面创建新池

SQL 语句: show databases; 执行

保存 返回

## 7.5 迁移至AASV9

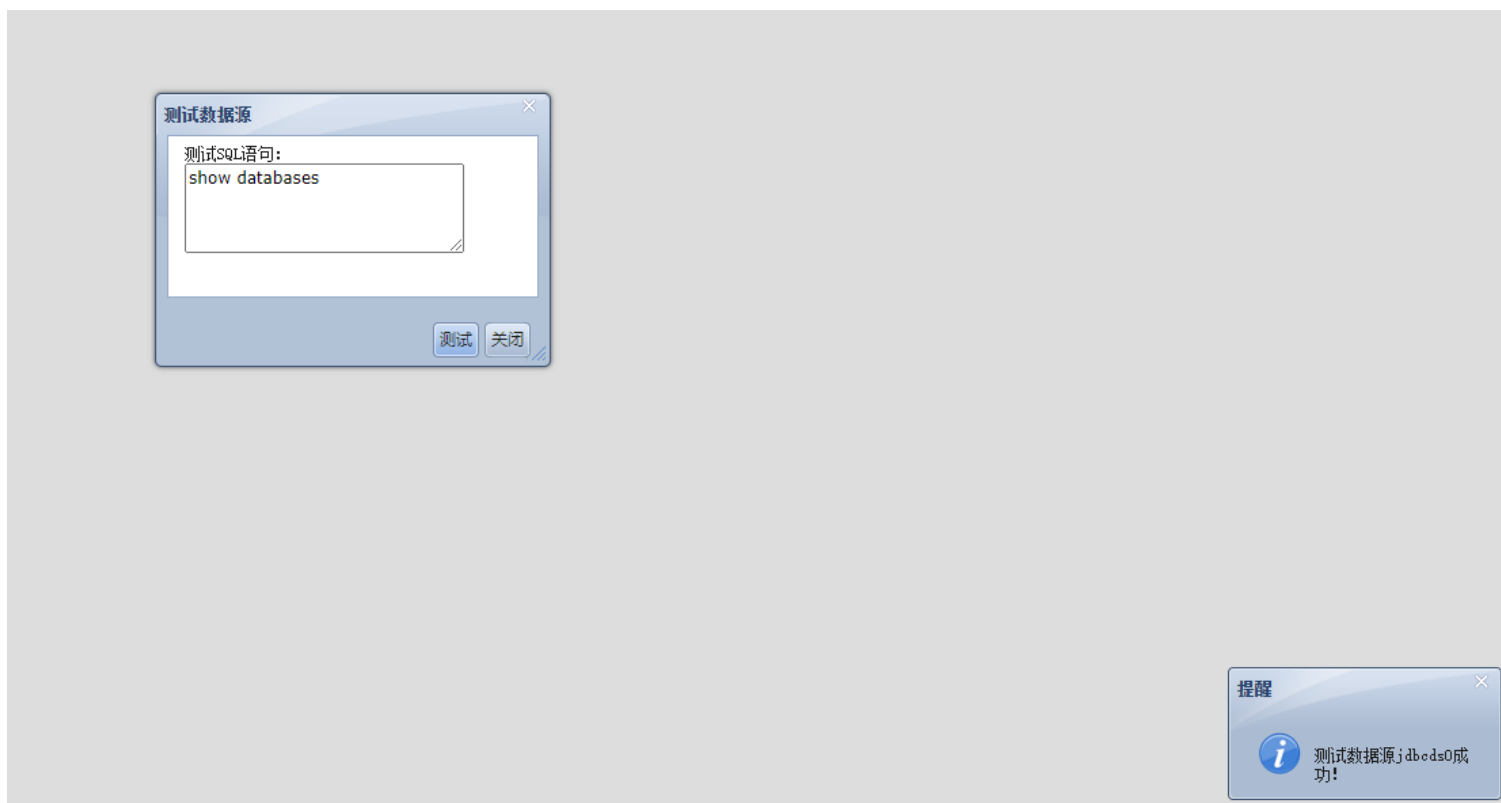
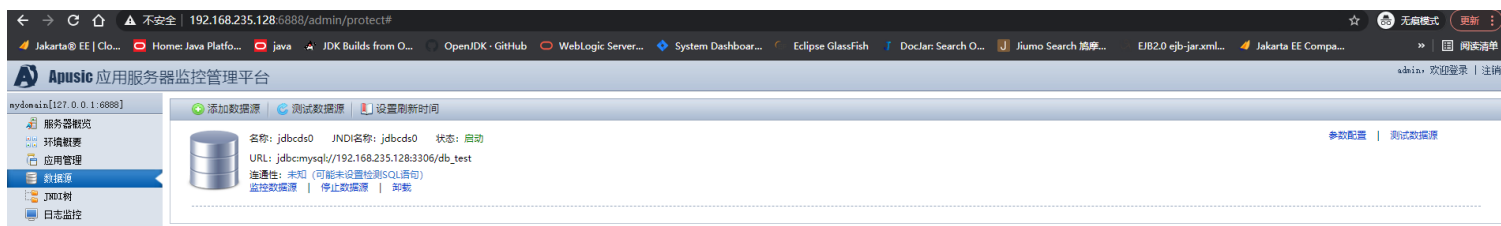
配置 `install_home` 配置项为AASV9安装目录。

```
#Apusic服务器安装目录
#install_home=/opt/aas/ApusicAS
install_home=/opt/aas/aasv9
```

执行 `sh bin/migrate install` 迁移数据源至AASV9。

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-23 10:36:43 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-23 10:36:43 INFO ConflictClass:43 - 类冲突检测
2021-07-23 10:36:44 INFO AasV9Parser:33 - 开始应用迁移
2021-07-23 10:36:44 INFO AasV9Parser:104 - 复制依赖jar包到AAS服务器
2021-07-23 10:36:44 INFO AasV9Parser:110 - @@@@应用迁移成功，请重启AAS 服务器!@@@@
```

重启AAS服务器并登录管控台，测试数据源迁移效果如下图。



## 7.6 EJB迁移

之前将WebLogic 中的EJB应用迁移到AAS。如已部署应用程序包 `ejb2-web.ear`。



## 7.6.1 配置迁移分析目录

配置 `conf/conf.properties` 的配置项 `server_home` 为WebLogic安装目录。

```
#=====Web/EJB/DataSource transfer config
#WebLogic/Tomcat/AASV9 安装目录
server_home=/opt/weblogic/wls12130/
```

## 7.6.2 迁移分析

执行 `sh bin/migrate analyse` 生成迁移输出信息。

```
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-07-19 14:11:48 WARN ParserFactory:32 - 未找到install_home服务器配置信息
2021-07-19 14:11:48 INFO AppsParser:29 - 解析applications目录中的独立应用
CLASSPATH=/opt/tools/jdk1.8.0_281/lib/tools.jar:/opt/weblogic/wls12130/wlserver/server/lib/weblogic_sp.jar:/opt/weblogic/wls12130/wlserver/server/lib/weblogic.jar:/opt/weblogic/wls12130/oracle_common/modules/net.sf.antcontrib_1.1.0.0_1-0b3/lib/ant-contrib.jar:/opt/weblogic/wls12130/wlserver/modules/features/oracle.wls.common.nodemanager_2.0.0.0.jar:
PATH=/opt/weblogic/wls12130/wlserver/server/bin:/opt/weblogic/wls12130/oracle_common/modules/org.apache.ant_1.9.2/bin:/opt/tools/jdk1.8.0_281/jre/bin:/opt/tools/jdk1.8.0_281/bin:/usr/local/sbin:/usr/local/bin:/usr/bin:/opt/tools/jdk1.8.0_281/bin:/opt/GmSSL/bin:/root/bin:/opt/weblogic/wls12130/oracle_common/modules/org.apache.maven_3.0.5/bin
Your environment has been set.
DECRYPTION_FLAG=root
2021-07-19 14:11:51 INFO JarSecurity:19 - class文件安全检测
[INFO]
Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.
About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
False Positives: https://jeremylong.github.io/DependencyCheck/general/suppression.html
Sponsor: https://github.com/sponsors/jeremylong
[INFO] Analysis Started
[INFO] Finished Archive Analyzer (0 seconds)
[INFO] Finished File Name Analyzer (0 seconds)
[INFO] Finished Jar Analyzer (0 seconds)
[INFO] Finished Central Analyzer (0 seconds)
[INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
[INFO] Created CPE Index (6 seconds)
[INFO] Finished CPE Analyzer (7 seconds)
[INFO] Finished False Positive Analyzer (0 seconds)
[INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] Finished Sonatype OSS Index Analyzer (0 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Analysis Complete (8 seconds)
[INFO] Writing report to: /opt/migration/migration/out/domains/mydomain/dependency-check/dependency-check-report.html
[INFO] 输出jar安全检测报告(dependency-check-report.html)至out/domains/mydomain/dependency-check成功
2021-07-19 14:12:04 INFO VendorClass:28 - 厂商依赖检测
[root@localhost migration]#
```

## 7.6.3 安装至AASV10

修改配置项 `install_home` 为AASV10安装目录。

```
#Apusic服务器安装目录
install_home=/opt/aas/ApusicAS
```

执行 `sh bin/migrate install` 迁移应用/配置到AASV10。

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-19 14:27:10 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-19 14:27:10 INFO ConflictClass:43 - 类冲突检测
2021-07-19 14:27:18 INFO AasV10Parser:39 - 开始应用迁移
已有name=jdbcds0的jdbc-connection-pool节点
请选择操作? [1:跳过,2:覆盖,-1:退出] 2
已有jndi-name=jdbcds0的jdbc-resource节点
请选择操作? [1:跳过,2:覆盖,-1:退出] 2
已有resource-ref对资源jdbcds0的引用
请选择操作? [1:跳过,2:覆盖,-1:退出] 2
2021-07-19 14:27:27 INFO AasV10Parser:101 - mysql-connector-java-commercial-5.1.22-bin.jar文件已经存在
@@@应用迁移成功,请重启AAS服务器!@@@
[root@localhost migration]#
```

重启AAS并登录管理控制台查看迁移的应用。

#### 应用程序

应用程序可以是企业应用程序, Web 应用程序或各种类型的模块。通过单击重新加载链接重新启动应用程序或模块,此操作仅适用于启用了应用程序或模块的目标。

已部署的应用程序 (1)

选择	名称	部署顺序	已启用	引擎	操作
<input type="checkbox"/>	ejb2-web	100	✓	ear, web, ejb	<a href="#">重新部署</a>   <a href="#">重新加载</a>

← → ↻ 🏠 ⚠ 不安全 | 192.168.235.128:6888/web/ejb

应用 Jakarta® EE | Clo... Home: Java Platfo... java JDK Builds from O... OpenJDK · GitHub WebLogic Server...

```
hello ejb:1627012545
HelloBean hello: helloBean      class com.test.beans.session.HelloBean
StatefullHelloBean hello: statefulHelloBean      class com.test.beans.session.StatefullHelloBean
```

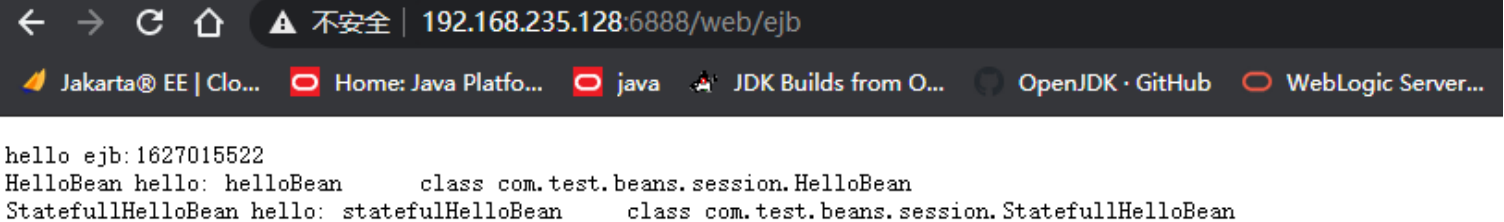
## 7.6.4 安装至AASV9

配置AASV9安装目录。

```
#Apusic服务器安装目录
#install_home=/opt/aas/ApusicAS
install_home=/opt/aas/aasv9
```

执行 `sh bin/migrate install` 迁移应用/配置到AASV9。

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-23 12:02:38 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-23 12:02:38 INFO ConflictClass:43 - 类冲突检测
2021-07-23 12:02:41 INFO AasV9Parser:33 - 开始应用迁移
2021-07-23 12:02:42 INFO AasV9Parser:110 - @@@@应用迁移成功, 请重启AAS 服务器!@@@@
[root@localhost migration]#
```



```
hello ejb:1627015522
HelloBean hello: helloBean      class com.test.beans.session.HelloBean
StatefullHelloBean hello: statefulHelloBean      class com.test.beans.session.StatefullHelloBean
```

## 7.7 迁移回退

迁移过程中会备份相关应用/配置到迁移工具目录backup目录。

执行 `sh bin/migrate rollback` 会使用backup目录下离当前时刻最近的备份文件恢复AAS相关配置，配置回退后重启AAS服务器即可。

```
[root@localhost migration]# sh bin/migrate rollback
[INFO] 设置路径/opt/migration/migration/
2021-07-19 16:01:29 INFO AasV10Parser:127 - =====从备份文件backup/20210719152408恢复=====
2021-07-19 16:01:29 INFO AasV10Parser:133 - =====备份文件恢复成功=====
[root@localhost migration]#
```

## 7.8 问题解答

### 7.8.1 数据源AES解密异常

WebLogic 迁移数据源会执行WebLogic的相关脚本以解密数据源AES密文，执行脚本时如出现如下错误需要配置

`MW_HOME` 环境变量为WebLogic安装目录，可在终端中零时配置环境变量 `export MW_HOME=xxxxxxx`。

```
[root@localhost migration]# unset MW_HOME
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-07-19 15:41:59 INFO AppsParser:29 - 解析applications目录中的独立应用
/opt/weblogic/wls12130/wlserver/server/bin/setWLSEnv.sh: line 41: /wlserver/common/bin/commEnv.sh: No such file or directory
The WebLogic Server wasn't found in directory /wlserver/server.
Please edit the startWebLogic.sh script so that the WL_HOME
variable points to the WebLogic installation directory.
Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/security/internal/SerializedSystemIni
    at com.apusic.util.AesDecrypt.main(AesDecrypt.java:17)
Caused by: java.lang.ClassNotFoundException: weblogic.security.internal.SerializedSystemIni
    at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:355)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
    ... 1 more
2021-07-19 15:42:00 ERROR WlsParser:125 - 数据源密码AES解密失败
```

## 7.8.2 安装重名

迁移应用/配置到AAS服务器时会和AAS已有配置对比，如果配置重名将会提示操作选择。

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-19 15:47:39 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-19 15:47:39 INFO ConflictClass:43 - 类冲突检测
2021-07-19 15:47:45 INFO AasV10Parser:39 - 开始应用迁移
已有name=jdbcds0的jdbc-connection-pool节点
请选择操作? [1:跳过,2:覆盖,-1:退出] █
```

迁移应用到AAS服务器时如果AAS服务器相关目录已有文件存在也会提示相关操作选择。

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-07-19 15:53:28 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-19 15:53:28 INFO ConflictClass:43 - 类冲突检测
2021-07-19 15:53:33 INFO AasV10Parser:39 - 开始应用迁移
2021-07-19 15:53:34 WARN WlsUtil:442 - ejb2-web.ear中的ejb2.jar已包含aas-ejb-jar.xml文件
已有name=jdbcds0的jdbc-connection-pool节点
请选择操作? [1:跳过,2:覆盖,-1:退出] 2
/opt/aas/jenkins-build/ApusicAS/aas/domains/mydomain/applications/_internal/ejb2-web/ejb2-web.ear 文件已存在!
请选择操作? (-1:退出操作,1:覆盖当前文件,2:以后都覆盖,3:跳过该文件,4:重名都跳过) 1
/opt/aas/jenkins-build/ApusicAS/aas/domains/mydomain/applications/ejb2-web/META-INF/MANIFEST.MF 文件已存在!
请选择操作? (-1:退出操作,1:覆盖当前文件,2:以后都覆盖,3:跳过该文件,4:重名都跳过) 2
2021-07-19 15:53:55 INFO AasV10Parser:101 - mysql-connector-java-commercial-5.1.22-bin.jar文件已经存在
@@@应用迁移成功,请重启AAS服务器!@@@
[root@localhost migration]# █
```

## 8 Tomcat迁移

Tomcat是Apache开源的轻量级Java Web应用服务器，支持Servlet和JSP，具有开源免费、轻量灵活、扩展性好等特点，应用广泛。

### 8.1 支持迁移的功能

- 1) Tomcat部署的war包,位于 tomcat安装根目录下的webapps目录下。
- 2) Tomcat JDBC Datasource，配置文件位于
  - `${TOMCAT_HOME}/conf/server.xml`
  - `${TOMCAT_HOME}/conf/context.xml`
  - `${TOMCAT_HOME}/conf/[EngineName]/[HostName]/context.xml.default`
  - `${TOMCAT_HOME}/conf/[EngineName]/[HostName]/[应用名].xml`

注意：tomcat默认引擎EngineName为Catalina，以及默认HostName为localhost。所在目录为 `/conf/Catalina/localhost`。

- 3) Tomcat JavaMail Sessions。配置文件位置同上。

### 8.2 Tomcat 中的示例应用和示例配置

- 1) 示例tomcat服务器webapps目录中包含了 `demo.war` 应用。

```
[root@dns webapps]# ls -lht
总用量 20M
-rw-r--r--.  1 777 root   20M 5月   6 01:50 demo.war
drwxr-x---.  5 777 root   103 4月  12 02:17 manager
drwxr-x---.  5 777 root    68 4月  12 02:17 host-manager
drwxr-x---.  7 777 root   111 4月  12 02:17 examples
drwxr-x---. 14 777 root   4.0K 4月  12 02:17 docs
drwxr-x---.  3 777 root   283 4月  12 02:17 ROOT
```

- 2) 示例tomcat服务器安装目录下 `conf/server.xml` 中配置了如下示例 JDBC Datasource数据源以及 示例 JavaMail Sessions。以下迁移说明均已该示例为演示。

```
<Resource name="jdbc/demo"
  auth="Container"
  type="javax.sql.DataSource"
  username="root"
  password="123456"
  driverClassName="com.mysql.jdbc.Driver"
  url="jdbc:mysql://127.0.0.1:3306/demo?useSSL=false&serverTimezone=UTC"
  maxActive="8"
  maxIdle="4"/>
```

```
<Resource name="mail/Session" auth="Container"
  type="javax.mail.Session"
  mail.smtp.host="smtp.qq.com"
  mail.smtp.user="123456789@qq.com"
  mail.transport.protocol="smtp"
  mail.smtp.auth="true"
  password="123456"/>
```

## 8.3 迁移至AAS10

### 8.3.1 安装迁移工具包

1) 获取安装包 `migration.zip` , 解压至服务器任意目录。假设指定安装解压的目录为 `/usr/local/migration` , 以下操作均以 `/usr/local/migration` 为相对目录操作。

2) 进入“安装路径/conf”目录。编辑目录下的 `conf.properties` , 填写对应的配置。配置文件配置示例如下:

`server_home` 配置tomcat服务器所在目录, 执行配置分析(migrate analyse)时必须填。

`install_home` 配置AASV9或AASV10的安装目录。执行配置迁移安装(migrate install)时必须填。

```
#tomcat服务器的所在目录 ex:/usr/local/tomcat
server_home=/usr/local/apache-tomcat-7.0.108
#Apusic服务器的所在目录 ex:/usr/local/ApusicAS
install_home=/usr/local/ApusicAS
```

### 8.3.2 配置分析

1) 执行 `./migrate.sh analyse` 命令开始分析Tomcat服务器的配置。

如有疑问, 执行 `./migrate -help` 查看命令说明。

```
[WARN] 使用说明:
migrate [options]

选项:
help          打印帮助命令说明
analyse       应用配置解析
install       应用配置迁移至Apusic
rollback      还原迁移
dependent-testing 厂商依赖检测
security-scan jar安全漏洞检测
class-conflict 类冲突检测
```

### 8.3.3 生成分析文件

1) 切换到“安装路径/out”目录下，可以查看生成的配置信息。

注意：生成的配置文件可以迁移至AASV10或者AASV9。如需同时迁移至AASV10和AASV9,只需分析一次即可。

```
./out/
├── domains
│   ├── mydomain
│   │   ├── applications
│   │   │   └── demo.war
│   │   ├── config
│   │   │   ├── domain.json
│   │   │   └── domain.xml
│   │   ├── dependency-check
│   │   │   └── demo
│   │   │       └── dependency-check-report.html
│   │   ├── lib
│   │   │   └── mysql-connector-java-8.0.23.jar
│   │   └── manufacturer-dependent-testing
│   │       └── demo
│   │           └── manufacturer-dependent-testing.txt
```

### 8.3.4 安装解析好的配置至AAS10

1) 切换到“安装路径/bin”目录下执行 `./migrate.sh install` 命令开始安装解析好的配置至AASV10。

如有疑问，执行 `./migrate -help` 查看命令说明。安装执行成功如图所示：

```
2021-07-15 23:06:38 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-15 23:06:38 INFO ConflictClass:43 - 类冲突检测
2021-07-15 23:06:40 INFO AasV10Parser:39 - 开始应用迁移
@@@@@应用迁移成功，请重启AAS 服务器!@@@@@
```

### 8.3.5 重启AAS10 查看应用是否迁移成功

重启AAS10，查看应用是否迁移成功。

### 8.3.6 查看已部署的应用程序

查看部署的应用，可以看到部署成功。

### 8.3.7 查看已迁移的 JDBC Resources

查看迁移的JDBC资源，点击：资源管理->JDBC->JDBC资源。

### 8.3.8 查看已迁移的JavaMail Sessions

查看迁移的JavaMail Sessions，点击：资源管理->JavaMail。

### 8.3.9 回滚迁移至AAS10的功能

1) 切换到“安装路径/bin”目录下执行 `./migrate.sh rollback` 命令开始迁移配置回滚操作。回滚前关闭AASV10。

关闭AAS10执行：切换到“`/AAS10/bin`”目录下，进行AAS10的关闭操作。

Linux 环境：`./asadmin stop-domain`

windows环境：`./asadmin.bat stop-domain`

注意区分系统环境。

## 8.4 迁移至AAS9

### 8.4.1 配置解析以及安装

1).如已经执行过Tomcat服务器的配置分析操作。2) 3) 跳过。

2).切换到“安装路径/bin”目录下执行 `./migrate.sh analyse` 命令开始分析Tomcat服务器的配置。

3).切换到“安装路径/out”目录下，可以查看生成的配置信息。

注意：生成的配置文件可以迁移至AASV10或者AASV9。所以如需迁移至AASV10和AASV9,步骤二只需分析一次即可。

```

./out/
├── domains
│   ├── mydomain
│   │   ├── applications
│   │   │   └── demo.war
│   │   ├── config
│   │   │   ├── domain.json
│   │   │   └── domain.xml
│   │   ├── dependency-check
│   │   │   └── demo
│   │   │       └── dependency-check-report.html
│   │   ├── lib
│   │   │   └── mysql-connector-java-8.0.23.jar
│   │   └── manufacturer-dependent-testing
│   │       └── demo
│   │           └── manufacturer-dependent-testing.txt

```

4. 进入 “`${Migration_home}/conf`” 目录。编辑目录下的 `conf.properties` , 填写AASV9的安装目录。配置文件配置示例如下：执行安装解析好的配置时, `server_home` 可以不填写。

```

#tomcat服务器的所在目录 ex:/usr/local/tomcat
server_home=/usr/local/tomcat7/apache-tomcat-7.0.108
#Apusic服务器的所在目录 ex:/usr/local/ApusicAS
install_home=/usr/local/AAS9

```

5).切换到 “`${Migration_home}/bin`” 目录下执行 `./migrate.sh install` 命令开始安装解析好的配置至AASV9。如有疑问, 执行 `./migrate -help` 查看命令说明。

```

2021-07-16 00:35:55 INFO    AnsApp:94 - 执行AAS本地迁移工具
2021-07-16 00:35:55 INFO    ConflictClass:43 - 类冲突检测
2021-07-16 00:35:57 INFO    AasV9Parser:32 - 开始应用迁移
2021-07-16 00:35:57 INFO    AasV9Parser:102 - 复制依赖jar包到AAS服务器
2021-07-16 00:35:57 INFO    AasV9Parser:108 - @@@@应用迁移成功, 请重启AAS 服务器!@@@@

```

#### 8.4.2 重启AAS9,查看应用是否迁移成功

重启AASV9的启动操作。如执行：`./startapusic`。

### 8.4.3 查看已部署的应用程序

查看部署的应用，点击：应用管理，以下可以看到部署成功。

### 8.4.4 查看已迁移的 JDBC Resources

查看迁移的JDBC资源，点击：数据源。

### 8.4.5 查看已迁移的JavaMail Sessions

查看迁移的JavaMail Sessions，点击：JNDI资源。

### 8.4.6 回滚迁移至AAS9的功能

1) 切换到“`/${Migration_home}/bin`”目录下执行 `./migrate.sh rollback` 命令开始迁移配置回滚操作。回滚前请关闭AASV9服务器。

2) 关闭AAS9执行，如下。

Linux 系统: `./stopapusic`

windows系统: `./stopapusic.cmd`

注意区分系统环境。

## 8.5 类冲突检测

1.进入安装目录执行命令 `./bin/migrate class-conflict`。

输出文件至 `/out/domains/mydomain/class-conflict/[应用名]\[时间戳]/class-conflict.docx`。

```
[root@AAS-168 bin]# ./migrate class-conflict
[INFO] 设置路径/opt/testz/migrate/1105/migration/
2021-11-05 18:40:37 INFO ConflictClass:61 - 类冲突检测
[root@AAS-168 bin]#
```

## 8.6 Jar安全漏洞检测

进入安装目录执行命令 `./bin/migrate security-scan`。

输出检测结果至 `/out/domains/mydomain/dependency-check/[应用名]/dependency-check-report.html`。

```

[root@dns MigrationAAS]# ./bin/migrate security-scan
[INFO] 设置路径./bin/./
[INFO] /usr/local/tomcat7/apache-tomcat-7.0.108/conf/server.xml
[INFO] true
[INFO] The System JDK Version is [ 1.8 ]
[INFO] 开始[ /usr/local/tomcat7/apache-tomcat-7.0.108/webapps/migrate.war ]所依赖jar进行安全漏洞检测...
[INFO] [INFO] Checking for updates
[INFO] [INFO] Skipping NVD check since last check was within 4 hours.
[INFO] [INFO] Check for updates complete (644 ms)
[INFO] [INFO]
[INFO] Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negativ
Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or oth
use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any d
ith the use of this tool, the analysis performed, or the resulting report.
[INFO] About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
[INFO] False Positives: https://jeremylong.github.io/DependencyCheck/general/suppression.html
[INFO] 赞助商 Sponsor: https://github.com/sponsors/jeremylong
[INFO] [INFO] Analysis Started
[INFO] [INFO] Finished Archive Analyzer (0 seconds)
[INFO] [INFO] Finished File Name Analyzer (0 seconds)
[INFO] [INFO] Finished Jar Analyzer (0 seconds)
[INFO] [INFO] Finished Central Analyzer (0 seconds)
[INFO] [INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] [INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] [INFO] Finished Hint Analyzer (0 seconds)
[INFO] [INFO] Created CPE Index (1 seconds)
[INFO] [INFO] Finished CPE Analyzer (2 seconds)
[INFO] [INFO] Finished False Positive Analyzer (0 seconds)
[INFO] [INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] [INFO] Finished Sonatype OSS Index Analyzer (0 seconds)
[INFO] [INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] [INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] [INFO] Analysis Complete (4 seconds)
[INFO] [INFO] Writing report to: /usr/local/MigrationAAS/./bin/./out/domains/mydomain/dependency-check/migrate/dependency-check-report.html
[INFO] 输出jar安全检测报告(dependency-check-report.html)至./bin/./out/domains/mydomain/dependency-check/migrate/成功

```

## 8.7 厂商依赖检测

1) 切换到 “\${Migration\_home}/bin” 目录下执行 `./migrate.sh dependent-testing` 命令开始分析Tomcat 服务器内部应用的厂商依赖。如有疑问，执行 `./migrate -help` 查看命令说明。日志查看

`/out/domains/mydomain/manufacture-dependent-testing/[应用]/manufacture-dependent-testing.txt`。

```

[root@dns MigrationAAS]# ./bin/migrate dependent-testing
[INFO] 设置路径./bin/./
[INFO] /usr/local/tomcat7/apache-tomcat-7.0.108/conf/server.xml
[INFO] true
[INFO] The System JDK Version is [ 1.8 ]
[INFO] 开始/usr/local/tomcat7/apache-tomcat-7.0.108/webapps/migrate.war的厂商依赖检测
[WARN] 检测到类:[com.kingdee.migtate.util.JNDILookUp]可能使用到了JNDI命名服务!
[WARN] 检测到类:[com.kingdee.migtate.controller.JNDILookUp]可能使用到了JNDI命名服务!
[WARN] 检测到类:[com.kingdee.migtate.controller.JDBCDemo]可能使用到了JNDI命名服务!
[ERROR] JNDI命名服务在Tomcat和Apusic上使用的方式区别如下:
Tomcat 中使用JavaMail会话:
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    Session session = (Session) envCtx.lookup("mail/Session")
    ...

Tomcat 中使用DataSource:
    InitialContext initCtx = new InitialContext();
    DataSource dataSource = (DataSource) initCtx.lookup("java:comp/env/jdbc/demo");
    ...

Apusic10 中使用JavaMail会话:
    Context initCtx = new InitialContext();
    Session session = (Session) initCtx.lookup("mail/Session");
    ...

Apusic10 中使用DataSource:
    InitialContext initCtx = new InitialContext();
    DataSource dataSource = (DataSource) initCtx.lookup("jdbc/demo")
    ...

[INFO] 输出厂商依赖结果至./bin/./out/domains/mydomain/manufacture-dependent-testing/migrate/manufacture-dependent-testing.txt

```

### Tomcat 中使用JavaMail会话:

```

Context initCtx = new InitialContext();

Context envCtx = (Context) initCtx.lookup("java:comp/env");

Session session = (Session) envCtx.lookup("mail/Session")

...

```

### Tomcat 中使用DataSource:

```

InitialContext initCtx = new InitialContext();

DataSource dataSource = (DataSource)

```

```
initCtx.lookup("java:comp/env/jdbc/demo");
```

...

Apusic10 中使用JavaMail会话:

```
Context initCtx = new InitialContext();
```

```
Session session = (Session) initCtx.lookup("mail/Session");
```

...

Apusic10 中使用DataSource:

```
InitialContext initCtx = new InitialContext();
```

```
DataSource dataSource = (DataSource) initCtx.lookup("jdbc/demo")
```

AASV9中使用方式和Tomcat保持一致。

注意: 如果Tomcat使用了这种会话方式。在迁移至AAS10时需修改代码中如上列出的使用代码进行更换。

迁移过程中如使用到了Tomcat底层依赖的类, 也需要进行依赖的更换才能在AAS服务器上部署。

```
[WARN] 检测到类[com.kingdee.migrate.mxbean.JMCServer]使用了以下Tomcat服务器内部专属类:
[WARN] org.apache.coyote.ActionCode
[WARN] org.apache.tomcat.SimpleInstanceManager
[WARN] org.apache.catalina.authenticator.DigestAuthenticator
[WARN] org.apache.juli.AsyncFileHandler
[WARN] org.apache.naming.NamingEntry
```

## 8.8 常见问题解答

1) 执行安装过程中提示资源或节点已存在。说明AAS10或者AAS9已经存在同名的JDBC Resources 或 JavaMail Sessions。解决方案:①修改Tomcat 配置文件中的JDBC Resources 或 JavaMail Sessions的配置命名, 即 `<Resource>` 描述符中的name 属性值。更改完毕后, 应用内部对此资源的引用命名也需要更改(具体查看厂商依赖功能)。②修改应用war包名称,默认项目请求前缀为项目war包名。更改完毕后再重新进行迁移配置分析。进行安装。

```
[root@dns bin]# ./migrate install
[INFO] 设置路径 /usr/local/MigrationAAS/
2021-07-16 01:54:10 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-16 01:54:10 INFO ConflictClass:43 - 类冲突检测
2021-07-16 01:54:12 INFO AasV10Parser:39 - 开始应用迁移
已有 name=jdbcdemo的 jdbc-connection-pool节点
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
已有 jndi-name=jdbc/demo的 jdbc-resource节点
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
已有 jndi-name=mail/Session的 mail-resource节点
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
已有 resource-ref对资源 jdbc/demo的引用
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
已有 resource-ref对资源 mail/Session的引用
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
已有 application-ref对应用 demo的引用
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
application name demo 已经被 demo.war使用
请选择操作? [1:跳过,2:覆盖,-1:退出] 1
java.lang.IllegalStateException: Context Path /demo 已被 demo.war模块使用
    at com.apusic.common.install.parser.AasV10Parser.handleApplicationsTag(AasV10Parser.java:329)
    at com.apusic.common.install.parser.AasV10Parser.install(AasV10Parser.java:226)
    at com.apusic.common.install.parser.AasV10Parser.parse(AasV10Parser.java:56)
    at com.apusic.wls.analysis.AnsApp.parse(AnsApp.java:96)
    at com.apusic.Bootstrap.main(Bootstrap.java:32)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.springframework.boot.loader.MainMethodRunner.run(MainMethodRunner.java:49)
    at org.springframework.boot.loader.Launcher.launch(Launcher.java:108)
    at org.springframework.boot.loader.Launcher.launch(Launcher.java:58)
    at org.springframework.boot.loader.JarLauncher.main(JarLauncher.java:88)
```

```
<Resource name="jdbc/demo" auth="Container"
    type="javax.sql.DataSource"
    username="root"
    password="123456"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://127.0.0.1:3306/demo?useSSL=false&serverTimezone=UTC"
    maxActive="8"
    maxIdle="4"/>
tomcat中可能重复的JDBC资源

<Resource name="mail/Session" auth="Container"
    type="javax.mail.Session"
    mail.smtp.host="smtp.qq.com"
    mail.smtp.user="123456789@qq.com"
    mail.transport.protocol="smtp"
    mail.smtp.auth="true"
    password="123456"/>
tomcat中可能重复mail 资源
```

----- 58

2) windows 下迁移提示文件正在被使用。解决方案：关闭AASV9或者AASV10。

```
E:\MigrationAAS\bin>migrate.cmd rollback
JAVA_HOME: D:\jdk
[INFO] 设置路径E:\MigrationAAS\bin\..\
2021-07-16 16:22:01 INFO AasV9Parser:126 - =====从备份文件backup\20210716155814恢复=====
2021-07-16 16:22:01 INFO AasV9Parser:132 - =====备份文件恢复成功=====
E:\MigrationAAS\bin>migrate.cmd install
JAVA_HOME: D:\jdk
[INFO] 设置路径E:\MigrationAAS\bin\..\
2021-07-16 16:27:38 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-07-16 16:27:38 INFO ConflictClass:43 - 类冲突检测
2021-07-16 16:27:40 INFO AasV9Parser:32 - 开始应用迁移
E:\AAS-V9.0\domains\mydomain\upload\demo\demo.war 文件已存在!
请选择操作? (-1:退出操作,1:覆盖当前文件,2:以后都覆盖,3:跳过该文件,4:重名都跳过) 1
2021-07-16 16:28:25 INFO AasV9Parser:102 - 复制依赖jar包到AAS服务器
2021-07-16 16:28:25 INFO AasV9Parser:126 - =====从备份文件backup\20210716162740恢复=====
2021-07-16 16:28:25 INFO AasV9Parser:132 - =====备份文件恢复成功=====
2021-07-16 16:28:25 ERROR AasV9Parser:112 - =====应用迁移失败=====
java.nio.file.FileSystemException: E:\AAS-V9.0\domains\mydomain\lib\mysql-connector-java-8.0.23.jar: 另一个程序正在使用此文件, 进程无法访问。
    at sun.nio.fs.WindowsException.translateToIOException(WindowsException.java:86)
    at sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:97)
    at sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:102)
    at sun.nio.fs.WindowsFileCopy.copy(WindowsFileCopy.java:165)
    at sun.nio.fs.WindowsFileSystemProvider.copy(WindowsFileSystemProvider.java:278)
    at java.nio.file.Files.copy(Files.java:1274)
```

## 9 WebSphere 迁移至AAS

WebSphere 是 IBM 的软件平台。它包含了编写、运行和监视全天候的工业强度的按需应变 Web 应用程序和跨平台、跨产品解决方案所需要的整个中间件基础设施，如服务器、服务和工具。

### 9.1 管控台配置数据源

登录<http://ip:9060/ibm/console> 管控台并配置数据源。如已配置数据源，忽略该操作。

#### 9.1.1 数据源迁移

配置迁移工具WebSphere安装目录配置项 `server_home`：

```
#####注意：windows 系统路径需要把\转换为/  
#=====Web/EJB/DataSource transfer config  
#WebLogic/Tomcat/AASV9/WebSphere 安装目录  
#server_home=/opt/weblogic/wls12130/  
server_home=/opt/websphere/AppServerV9.0  
  
#Apusic服务器安装目录  
#install_home=/opt/aas/ApusicAS  
#install_home=/opt/aas/aasv9  
  
#WebLogic 域路径名,不指定将解析所有域  
#wls.domainPath=D:/develop/wls1036/user_projects/domains/base_domain  
  
#安全检测更新，启用安全检测更新则在安全检测时会联网校验  
security.checker.update.enable=false
```

#### 9.1.2 迁移分析

执行 `sh bin/migrate analyse` 命令生成迁移风险输出:

```
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-08-02 14:28:27 WARN ParserFactory:32 - 未找到install_home服务器配置信息
2021-08-02 14:28:27 INFO AppsParser:29 - 解析applications目录中的独立应用
encoded password == "{xor}lTAWKw==" , decoded password == "root"
2021-08-02 14:28:28 INFO JarSecurity:19 - class文件安全检测
[INFO]
Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.
About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
False Positives: https://jeremylong.github.io/DependencyCheck/general/suppression.html
Sponsor: https://github.com/sponsors/jeremylong
[INFO] Analysis Started
[INFO] Finished Archive Analyzer (0 seconds)
[INFO] Finished File Name Analyzer (0 seconds)
[INFO] Finished Jar Analyzer (0 seconds)
[INFO] Finished Central Analyzer (0 seconds)
[INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
[INFO] Created CPE Index (2 seconds)
[INFO] Finished CPE Analyzer (3 seconds)
[INFO] Finished False Positive Analyzer (0 seconds)
[INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] Finished Sonatype OSS Index Analyzer (2 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Analysis Complete (6 seconds)
[INFO] Writing report to: /opt/migration/migration/out/domains/20210802142827/dependency-check/dependency-check-report.html
[INFO] 输出jar安全检测报告(dependency-check-report.html)至out/domains/20210802142827/dependency-check成功
2021-08-02 14:28:39 INFO VendorclassChecker:213 - 厂商依赖检测
[root@localhost migration]#
```

### 9.1.3 迁移至AASV10

配置迁移安装目录配置项 `install_home` 为AASV10安装目录:

```
#####注意: windows系统路径需要把\转换为/

#=====Web/EJB/DataSource transfer config
#WebLogic/Tomcat/AASV9/WebSphere 安装目录
#server_home=/opt/weblogic/wls12130/
server_home=/opt/websphere/AppServerV9.0

#Apusic服务器安装目录
install_home=/opt/aas/ApusicAS
#install_home=/opt/aas/aasv9

#WebLogic 域路径名,不指定将解析所有域
#wls.domainPath=D:/develop/wls1036/user_projects/domains/base_domain

#安全检测更新,启用安全检测更新则在安全检测时会联网校验
security.checker.update.enable=false
```

执行 `sh bin/migrate install` 安装:

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-08-02 14:38:58 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-08-02 14:38:58 INFO ConflictClass:43 - 类冲突检测
2021-08-02 14:39:00 INFO AasV10Parser:39 - 开始应用迁移
2021-08-02 14:39:01 INFO AasV10Parser:103 - mysql-connector-java-commercial-5.1.17-bin.jar文件已经存在
@@@应用迁移成功,请重启AAS服务器!@@@
[root@localhost migration]#
```

重启AASV10并检测数据源。

## 9.2 EJB迁移

通过WebSphere管控台安装 `ejb-web.ear` 应用。如已部署应用，忽略该操作。

### 9.2.1 迁移分析

配置WebSphere 安装目录配置项 `server_home`：

```
#####注意： windows 系统路径需要把\转换为/

#=====Web/EJB/DataSource transfer config
#WebLogic/Tomcat/AASV9/WebSphere 安装目录
#server_home=/opt/weblogic/wls12130/
server_home=/opt/websphere/AppServerV9.0

#Apusic服务器安装目录
install_home=/opt/aas/ApusicAS
#install_home=/opt/aas/aasv9

#WebLogic 域路径名,不指定将解析所有域
#wls.domainPath=D:/develop/wls1036/user_projects/domains/base_domain

#安全检测更新，启用安全检测更新则在安全检测时会联网校验
security.checker.update.enable=false
```

执行 `sh bin/migrate analyse` 解析WebSphere。

```
[root@localhost migration]# sh bin/migrate analyse
[INFO] 设置路径/opt/migration/migration/
2021-08-02 15:45:05 INFO AppsParser:29 - 解析applications目录中的独立应用
encoded password = "{xor}LTAKw==", decoded password = "root"
2021-08-02 15:45:07 INFO JarSecurity:19 - class文件安全检测
[INFO]
Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.
About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
False Positives: https://jeremylong.github.io/DependencyCheck/general/suppression.html
Sponsor: https://github.com/sponsors/jeremylong
[INFO] Analysis Started
[INFO] Finished Archive Analyzer (0 seconds)
[INFO] Finished File Name Analyzer (0 seconds)
[INFO] Finished Jar Analyzer (0 seconds)
[INFO] Finished Central Analyzer (0 seconds)
[INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
[INFO] Created CPE Index (3 seconds)
[INFO] Finished CPE Analyzer (3 seconds)
[INFO] Finished False Positive Analyzer (0 seconds)
[INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] Finished Sonatype OSS Index Analyzer (0 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Analysis Complete (4 seconds)
[INFO] Writing report to: /opt/migration/migration/out/domains/20210802154506/dependency-check/dependency-check-report.html
[INFO] 输出jar安全检测报告(dependency-check-report.html)至out/domains/20210802154506/dependency-check成功
2021-08-02 15:45:14 INFO VendorClassChecker:213 - 厂商依赖检测
[root@localhost migration]#
```

### 9.2.2 迁移至AASV10

配置 `install_home` 配置项为AASV10安装目录，并执行 `sh bin/migrate install`：

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-08-02 15:48:09 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-08-02 15:48:09 INFO ConflictClass:43 - 类冲突检测
2021-08-02 15:48:11 INFO AasV10Parser:39 - 开始应用迁移
2021-08-02 15:48:12 INFO AasV10Parser:103 - mysql-connector-java-commercial-5.1.17-bin.jar文件已经存在
@@@@应用迁移成功，请重启AAS 服务器!@@@@
[root@localhost migration]#
```

启动AASV10并访问迁移的应用。

### 9.2.3 迁移至AASV9

配置 `install_home` 配置项为AASV9安装目录，并执行 `sh bin/migrate install`：

```
[root@localhost migration]# sh bin/migrate install
[INFO] 设置路径/opt/migration/migration/
2021-08-02 16:18:43 INFO AnsApp:94 - 执行AAS本地迁移工具
2021-08-02 16:18:43 INFO ConflictClass:43 - 类冲突检测
2021-08-02 16:18:44 INFO AasV9Parser:39 - 开始应用迁移
2021-08-02 16:18:44 WARN AasV9Parser:220 - AASV9 数据源配置只支持指定驱动类,数据源配置已被忽略
2021-08-02 16:18:45 INFO AasV9Parser:115 - 复制依赖jar包到AAS服务器
2021-08-02 16:18:45 INFO AasV9Parser:121 - @@@@应用迁移成功，请重启AAS 服务器!@@@@
[root@localhost migration]#
```

启动AASV9并访问迁移的应用。

## 9.3 迁移回退

执行 `sh bin/migrate rollback` 即可回退AAS的配置应用。

## 10 AASV9 迁移至AASV10

由于迁移配置分析输出为AASV10格式数据，如果安装目标机器为AASV9则会对配置解析为AASV9配置。只需要在配置文件中配置 `server_home` 为AASV9安装目录， `install_home` 配置为AASV10的安装目录即可。

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](http://www.apusic.com)

