

解决方案实践

亿信华辰数据中台解决方案实践

文档版本 1.1
发布日期 2024-04-18



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

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。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 方案概述	1
2 资源和成本规划	4
3 实施步骤	7
3.1 EsMDM 主数据管理平台	7
3.1.1 安装环境	7
3.1.2 产品 WAR 包单机部署	9
3.1.2.1 TOMCAT 上部署睿码平台	9
3.1.2.2 Weblogic 上部署睿码平台	12
3.1.2.2.1 在 Weblogic 中创建 Domain	12
3.1.2.2.2 部署睿码平台	15
3.1.2.2.3 Weblogic 集群部署	19
3.1.2.3 Tongweb6.1 上部署睿码平台	31
3.1.2.3.1 部署环境配置	31
3.1.2.3.2 部署睿码	31
3.1.2.4 金蝶 10 上部署睿码平台	34
3.1.2.4.1 部署环境配置	34
3.1.2.4.2 部署睿码	35
3.1.2.5 TAS2.8.5 上部署睿码平台	36
3.1.2.5.1 部署环境配置	36
3.1.2.5.2 部署睿码	37
3.1.2.6 Docker 上部署睿码平台	39
3.1.3 产品 WAR 包集群部署	40
3.1.3.1 Redis 安装	42
3.1.3.2 部署包和产品注册和配置	44
3.1.3.3 集群文件配置	44
3.1.3.4 启动服务器、集群检查与注意事项	49
3.1.3.5 集群扩容说明	50
3.1.4 初始化服务器	52
3.1.5 系统诊断	54
3.1.6 疑难问题处理方案及数据库权限分配	56
3.2 睿治数据治理平台	59
3.2.1 安装环境	59

3.2.2 安装程序及依赖驱动清单.....	61
3.2.3 产品典型部署架构图.....	62
3.2.4 产品 WAR 包单机部署.....	64
3.2.4.1 TOMCAT 上部署数据治理平台.....	64
3.2.4.2 Weblogic 上部署数据治理平台.....	68
3.2.4.2.1 在 Weblogic 中创建 Domain.....	68
3.2.4.2.2 部署数据治理平台.....	71
3.2.4.2.3 Weblogic 集群部署.....	74
3.2.4.3 Tongweb6.1 上部署数据治理平台.....	86
3.2.4.4 金蝶 10 上部署数据治理平台.....	89
3.2.4.5 TAS2.8.5 上部署数据治理平台.....	91
3.2.4.6 Docker 上部署数据治理平台.....	93
3.2.5 产品 WAR 包集群部署.....	95
3.2.5.1 集群场景说明、工作目录指定及 JVM 参数设置.....	95
3.2.5.2 Redis 安装.....	97
3.2.5.3 部署包产品注册与集群文件配置.....	99
3.2.5.4 启动服务器、集群检查及注意事项.....	104
3.2.5.5 集群扩容说明.....	105
3.2.6 实时处理服务部署.....	107
3.2.6.1 Linux 环境单机版部署.....	107
3.2.6.2 Linux 环境集群版部署.....	108
3.2.6.3 “银河麒麟”系统单机版部署.....	109
3.2.6.4 Hdfs、大数据相关驱动安装与 Standalone 模式 local 配置.....	110
3.2.6.5 Standalone 模式 spark 配置.....	111
3.2.7 边缘采集节点部署.....	113
3.2.8 初始化服务器.....	115
3.2.9 系统诊断.....	118
3.2.10 疑难问题处理方案及数据库权限分配.....	119
3.3 ABI 一站式数据分析平台.....	123
3.3.1 安装环境.....	123
3.3.2 导出截图功能要求.....	127
3.3.3 TOMCAT 中部署 ABI.....	127
3.3.3.1 Windows 操作系统.....	127
3.3.3.2 Linux 操作系统.....	129
3.3.3.3 TOMCAT 常用配置.....	130
3.3.4 Weblogic 中部署 ABI.....	132
3.3.4.1 在 Weblogic 中创建 Domain.....	132
3.3.4.2 部署 ABI.....	135
3.3.4.3 Weblogic 单机双节点集群部署.....	138
3.3.4.4 Weblogic 双机集群部署.....	149
3.3.4.5 向计算机分配服务器、检查配置概要.....	157
3.3.4.6 创建 Weblogic 域、集群中辅节点操作.....	158

3.3.4.7 启动集群服务.....	160
3.3.5 Websphere 中部署 ABI.....	170
3.3.5.1 在 Websphere8.5 中部署.....	170
3.3.5.2 Websphere 常用调配.....	174
3.3.5.3 Websphere 集群部署.....	178
3.3.5.3.1 实例说明.....	178
3.3.5.3.2 创建概要文件.....	179
3.3.5.3.3 服务启动.....	180
3.3.5.3.4 创建集群.....	181
3.3.5.3.5 IHS 的部署.....	183
3.3.5.3.6 安装插件.....	185
3.3.5.3.7 应用部署与启动应用程序.....	187
3.3.5.4 启动命令.....	188
3.3.6 金蝶 AAS V9.0 中部署 ABI.....	188
3.3.6.1 部署环境配置.....	189
3.3.6.2 部署 ABI.....	191
3.3.7 中创 InforSuite V10 中部署 ABI.....	192
3.3.8 Tongweb V6.1 中部署 ABI.....	194
3.3.9 华宇 tas2.8.5 中部署 ABI.....	197
3.3.10 Docker 上部署 ABI.....	200
3.3.11 K8S 上部署 ABI.....	201
3.3.12 初始化 ABI.....	206
3.3.13 产品集群部署（单 redis 部署）.....	208
3.3.14 产品集群部署（redis 哨兵部署）.....	220
3.3.15 系统诊断.....	234
3.3.16 疑难问题处理方案.....	235
3.3.16.1 中间件相关.....	235
3.3.16.2 数据库相关.....	235
3.3.16.3 应用相关.....	238
3.3.16.4 Redis 部署相关.....	238
3.3.16.5 Docker 相关、IPv6 部署.....	239
3.3.17 附录.....	240
4 附录.....	250
5 修订记录.....	251

1 方案概述

应用场景

企业在进行数据化转型或数据管理时，存在以下的痛点：

- 数据不准确，存在大量编码不规范、格式不正确、含义不清晰的数据
- 数据难整合，跨业务流、跨架构和跨系统的数据关系混乱，关联分析十分困难
- 数据难溯源，业务数据出现问题时，定位和解读数据要消耗大量的人力和时间
- 数据不安全，数据存储、传递、使用不合理，造成敏感信息泄露

解决方案实践的应用行业推荐：

- 金融行业：金融统一监管报表平台，金融风险全面管控，金融大数据中台
- 租赁行业：租赁行业大数据平台，租赁资产监督管控
- 医疗卫生：省级卫生数据管理与服务平台，医院数据治理与运营分析平台
- 能源：应急与能源行业大数据平台，电网规划大数据平台
- 其他：零售行业数据治理平台，环保行业数据治理平台

本章节介绍如何通过搭建数据中台帮助企业将所有相关数据转化为数据资产，通过算法模型构建企业高度统一、标准化的数据共享服务，提升企业各团队之间的信息共享能力、业务协同能力和经验沉淀能力，从而不断为现有业务和创新业务赋能。

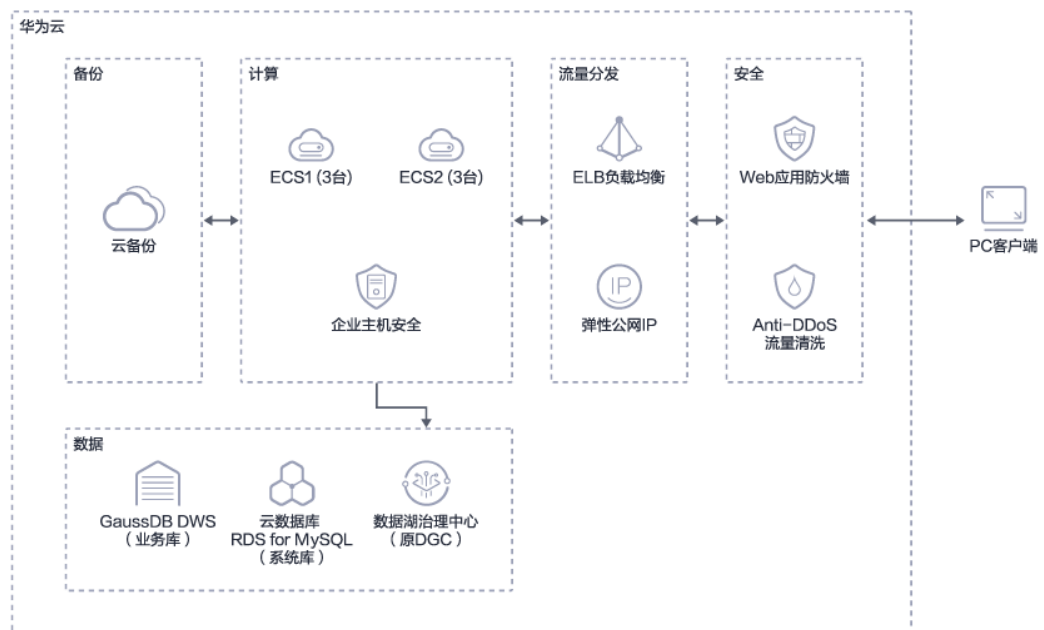
方案架构

图 1-1 制造业数据中台解决方案架构



1. 方案主要由华为云计算底座+DataArts Studio/睿治+数据仓库DWS+ROMA Connect+主数据管理平台+一站式数据分析平台组成的数据中台解决方案；
2. 通过DataArts Studio/睿治开展全链路数据治理管控，构建企业级数据资产，全面梳理企业数据情况；
3. 通过主数据管理平台构建主数据模型，开展主数据集中管控，实现主数据全生命周期管理；
4. 通过一站式数据分析实现数据综合分析，全面提升数据应用价值。

图 1-2 数据中台解决方案部署架构图



架构描述：

1. 方案由企业主机安全等计算，ECS为基础部署应用平台；
2. 方案采用了华为云RDS for mysql作为系统库和GaussDB DWS和DGC两种数据库作为数据分析和存储的数据库；
3. 方案使用了ELB负载均衡和弹性公网IP进行流量分发；
4. 方式使用了云备份作为数据和应用的常规备份以及异地备份；
5. 方式使用了Web应用防火墙和Anti-DDos流量清洗进行安全防控；
6. 方案同时还能够与客户已有ERP，OA等应用系统的集成，打破数据孤岛；
7. 方案针对金融、租赁、能源、制造等行业进行数据中台搭建，覆盖200多个细分行业；

方案优势

- 一站式开发运营平台
全面覆盖数据采集、清洗、融合、存储、分析，贯穿数据全流程一站式开发运营平台，帮助企业构建完整数据中台解决方案。
- 可复用性行业知识库
提供垂直行业可复用的领域知识库，涵盖行业数据标准、行业领域主数据模型、数据分析模块，快速构建端到端的解决方案
- 全链路数据治理管控
数据全生命周期管控，提供全自动元数据采集和关联，智能化映射和落标，数据处理全流程质量监控，异常事件实时通知。
- AI增强分析实现洞察
言出数随，智能问答，以对话方式查询和分析数据。问题一经提出，数据图表即刻呈现，降低AI应用门槛，实现AI+BI智能分析。

2 资源和成本规划

表 2-1 资源和成本规划选购说明

维度	说明
华为云服务	必选 环境搭建需要使用的云服务
应用	必选 第三方软件规划情况，根据需求选择某个规格的EsMDM、睿治数据治理平台及ABI一站式数据分析平台，可单独选购某个产品，也可组合选购

表 2-2 资源和成本规划

华为云服务	规格	数量	计费模式	计费周期	总价
数据仓库服务	DWS-100GB 超高IO磁盘按需套餐包: * 6 (份) 节点规格: dwsx2.2xlarge.m7 云数仓 x86 8 vCPUs 64 GB * 3 节点	1	包周期	1月	¥6434.94
弹性负载均衡	实例规格类型: 共享型负载均衡 IP费用: 1LCU 全动态BGP 流量: 全动态BGP 流量 1GB	1	按需计费	1448小时	¥29.76
企业主机安全	规格: 企业版	6	包周期	1月	¥540.00
Anti-DDoS流量清洗	免费	1	免费	免费	免费
云数据库RDS(for MySQL)	规格: MySQL 8.0 主备 通用型 8核32GB 存储空间: SSD云盘 400GB	1	包周期	1月	¥2640.00

华为云服务	规格	数量	计费模式	计费周期	总价
弹性云服务器	规格: X86计算 通用计算增强型 c6s.xlarge.2 4核 8GB 镜像: CentOS CentOS 7.9 64bit 系统盘: 通用型SSD 40GB 数据盘: 通用型SSD 100GB 弹性公网IP: 全动态BGP 独享 按带宽计费 10Mbit/s	6	包周期	1月	¥5802.00
弹性公网IP	带宽费用: 独享 全动态BGP 按带宽计费 10Mbit/s 弹性公网IP费用: 1个	1	包周期	1月	¥515.00
云备份	存储库类型: 云服务器备份存储库 1000GB	1	包周期	1月	¥200.00
Web应用防火墙	规格选择: 入门版	1	包周期	1月	¥99.00
数据湖治理中心 (原DGC)	版本: 专家版	1	包周期	1月	¥5000.00

表 2-3 a

应用	规格	数量	计费模式	计费周期	总价
EsMDM主数据管理平台	标准版	1	包周期	1年	¥256000.00
EsMDM主数据管理平台	企业版	1	包周期	1年	¥373000.00
EsMDM主数据管理平台	标准版	1	按次买断	-	¥550000.00
EsMDM主数据管理平台	企业版	1	按次买断	-	¥800000.00
睿治数据治理平台	标准版	1	按次买断	-	¥2400000.00
ABI一站式数据分析平台	标准版、50用户	1	包周期	1年	¥50000.00
ABI一站式数据分析平台	标准版、100用户	1	包周期	1年	¥100000.00
ABI一站式数据分析平台	标准版、不限用户	1	包周期	1年	¥150000.00

应用	规格	数量	计费模式	计费周期	总价
ABI一站式数据分析平台	企业版、50用户	1	包周期	1年	¥75000.00
ABI一站式数据分析平台	企业版、100用户	1	包周期	1年	¥150000.00
ABI一站式数据分析平台	企业版、不限用户	1	包周期	1年	¥200000.00
ABI一站式数据分析平台	高可用版、不限用户数	1	包周期	1年	¥300000.00

3 实施步骤

3.1 EsMDM主数据管理平台

3.2 睿治数据治理平台

3.3 ABI一站式数据分析平台

3.1 EsMDM 主数据管理平台

3.1.1 安装环境

服务器推荐配置

- 小范围试用和用户量较小的：用户数小于100，并发用户数小于10，主数据分发总数据量达百万级

表 3-1 用户量小的配置

配置项目	单项配置说明
应用服务器	DELL PowerEdge R340或其他同类产品 CPU：英特尔® 至强® E-2234处理器（4C/8T） 内存：16G 硬盘：RAID5 1T 网络：1000Mbps
数据库服务器	DELL PowerEdge R440或其他同类产品 CPU：英特尔® 至强® E-2274G处理器（4C/8T） 内存：32G 硬盘：RAID5 按需 网络：1000Mbps

配置项目	单项配置说明
操作系统及软件	操作系统: Windows、CentOS、Red Hat 中间件: Tomcat8 数据库: Oracle11g

- **大型企业、省级机构:** 用户数小于300, 并发用户数小于30, 主数据分发总数据量达千万级

表 3-2 用户量大的配置

配置项目	说明
应用服务器	IBM X3650或HP PowerEdge R720 CPU: 英特尔® 至强®银牌 4208处理器 (8C/16T) 内存: 32G 硬盘: RAID1 1T 网络: 1000Mbps
数据库配置	DELL PowerEdge R540或其他同类产品 CPU: 英特尔® 至强®金牌 6226处理器 (12C/24T) 内存: 64G 硬盘: 按需 网络: 1000Mbps
推荐操作系统及软件	操作系统: CentOS6 中间件: tomcat-8.5.12 数据库: Oracle11g

客户端配置

表 3-3 客户端

所属项	说明
CPU	不低于酷睿i3
内存	不小于4G
网络	不低于100KB/s传输速度

表 3-4 客户端浏览器

操作系统	浏览器
Windows	IE11,Chrome浏览器, 360安全浏览器

程序安装概况

常规场景部署：为传统部署，仅需通过中间件部署程序war包。

部署流程：

1. 在服务器中安装中间件后，在中间件上放入应用war包完成单机环境部署，参考第二章内容；
2. 如果需要搭建集群应用环境，则需要通过redis完成集群环境部署，参考第三章内容。

3.1.2 产品 WAR 包单机部署

3.1.2.1 TOMCAT 上部署睿码平台

TOMCAT 及 JDK 版本推荐

因jar包中存在module-info.class需升级tomcat版本，tomcat版本需要用9.0.0.M18以上、8.5.12以上的版本，JDK版本必须1.8及以上。

Windows 操作系统

下面讲述的是在Windows环境下安装部署睿码的整个过程。

安装JDK

安装JAVA运行环境（必须jdk1.8或以上版本），如下图所示：

图 3-1 安装 JDK1



图 3-2 安装 JDK2



安装过程中，可以默认设置，直接NEXT完成安装，具体过程略。

安装TOMCAT

安装TOMCAT有两种方法。一种是使用TOMCAT的安装版程序，一种是使用TOMCAT的解压版文件。这两种方法的区别在于，通过安装版程序安装TOMCAT，不需要设置JAVA环境变量，使用解压版文件需要设置环境变量。

现在基本都使用TOMCAT的解压版程序，只需将TOMCAT压缩文件解压到某一目录即可。

- 配置环境变量

如果是解压版的TOMCAT，需要在startup.bat中设置JAVA环境变量。

```
Set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_221;红色路径改为实际JDK安装路径
```

- 参数优化

修改内存大小一般在启动文件startup.bat中设置，设置内存不能低于如下数值：

```
Set JAVA_OPTS=-Xmx2048m -Xms256m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m
```

- 部署睿码

找到TOMCAT安装目录中的webapps文件夹，删除除ROOT文件夹外的其他文件和文件夹。

将睿码服务器的WAR包文件复制到上述的webapps文件夹中。

- 启动服务器

运行TOMCAT安装目录中的bin\startup.bat，启动TOMCAT服务。

在IE地址栏中输入报表服务器地址，输入地址时，应带上TOMCAT的端口和应用的目录，即WAR包的目录名，完成服务器的初始化工作。

Linux 操作系统

在Linux中部署与windows中基本相同，下面以TOMCAT为例讲解在linux下部署的步骤。

Linux 下使用“变量名=变量值”设置变量，并使用 export 命令将其导出为环境变量。下面是安装并配置jdk、tomcat的环境变量。

安装JDK

```
#将jdk-8u131-linux-x64.tar.gz上传到/usr/local目录下  
#cd /usr/local (到usr/local目录下)  
#chmod a+x jdk-8u131-linux-x64.tar.gz (执行命令)  
#tar -xzf jdk-8u131-linux-x64.tar.gz (执行命令)
```

安装TOMCAT

同上，把tomcat (apache-tomcat-8.5.12.zip) 上传到你想要安装的位置，如： /usr/local下

```
# unzip apache-tomcat-8.5.12.zip (解包) 到此就完成安装了。
```

配置环境变量

在startup.sh或catalina.sh的开头位置设置JAVA_HOME

```
JAVA_HOME=/usr/local/jdk1.8.0_131
```

参数优化

修改内存大小一般在启动文件startup.sh中设置，设置内存不能低于如下数值：

```
export JAVA_OPTS=" -Xmx2048m -Xms256m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -  
Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK"
```

部署睿码

将mdm (此名可以更改).war包复制到\Tomcat \webapps文件夹下

启动服务器；启动tomcat

```
#!/startup.sh
```

访问服务器，形如： http://127.0.0.1:8080/mdm，完成服务器的初始化工作。后缀mdm就是部署时war包的名称，如有更改，请对应更改。

TOMCAT 常用配置

修改端口

TOMCAT安装时默认的端口8080，TOMCAT安装版可以在安装中修改。对于安装完成后和解压版TOMCAT如何修改呢？

找到TOMCAT安装目录下的\conf\server.xml文件，用记事本打开，搜索安装时设置的端口值或默认的值8080，将其替换为要设置的端口值。

在修改端口时要注意，先要停止TOMCAT服务，再来修改。修改完成后，重启TOMCAT服务，即可通过修改后的端口访问WEB应用。

JVM设置

TOMCAT默认内存最大内存为128M，在大用户量情况下容易出现内存不够的情况，所以需要调整。

修改内存大小一般在启动文件startup.bat或startup.sh中设置，设置内存不能低于如下数值：

```
Set JAVA_OPTS=-Xmx2048m -Xms256m -XX:MetaspaceSize=128m  
-XX:MaxMetaspaceSize=256m(windows)
```

```
JAVA_OPTS=" -Xmx2048m -Xms256m -XX:MetaspaceSize=128m  
-XX:MaxMetaspaceSize=256m" (linux)
```

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，在64bit环境下分配8G内存。

⚠ 注意

某些JDK版本或操作系统限制最大堆内存，不支持分配超过1.5G的内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

线程调整

表 3-5 属性

属性名	描述
maxThreads	Tomcat使用线程来处理接收的每个请求。这个值表示Tomcat可创建的最大的线程数。
acceptCount	指定当所有可以使用的处理请求的线程数都被使用时，可以放到处理队列中的请求数，超过这个数的请求将不予处理。
minSpareThreads	Tomcat初始化时创建的线程数。
maxSpareThreads	一旦创建的线程超过这个值，Tomcat就会关闭不再需要的socket线程。

上述参数在conf/ server.xml中进行设置，TOMCAT上述默认值如下：

```
maxThreads: 150  
acceptCount: 100  
minSpareThreads: 25  
maxSpareThreads: 75
```

在一般情况下够用，如果同时并发用户数量很大，需要调整maxThreads参数，调整原则为每秒用户请求数*请求操作执行时间，比如每秒有100个请求，每个请求执行时间为1秒，maxThreads设置为100即可。同时要适当调整maxSpareThreads参数，不要关闭socket线程后又马上生成的操作。

3.1.2.2 Weblogic 上部署睿码平台

3.1.2.2.1 在 Weblogic 中创建 Domain

执行安装目录下/Oracle/Middleware/Oracle_Home/wlserver/common/bin/config.sh

图 3-3 配置向导 1



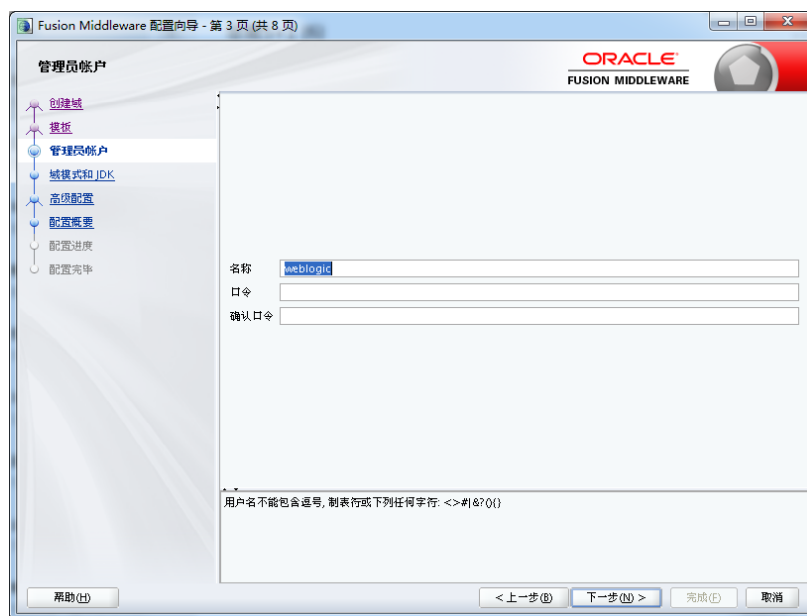
按照实际情况修改路径

图 3-4 配置向导 2



设置账户密码，下一步

图 3-5 配置向导 3



选择开发或生产模式，下一步

图 3-6 配置向导 4



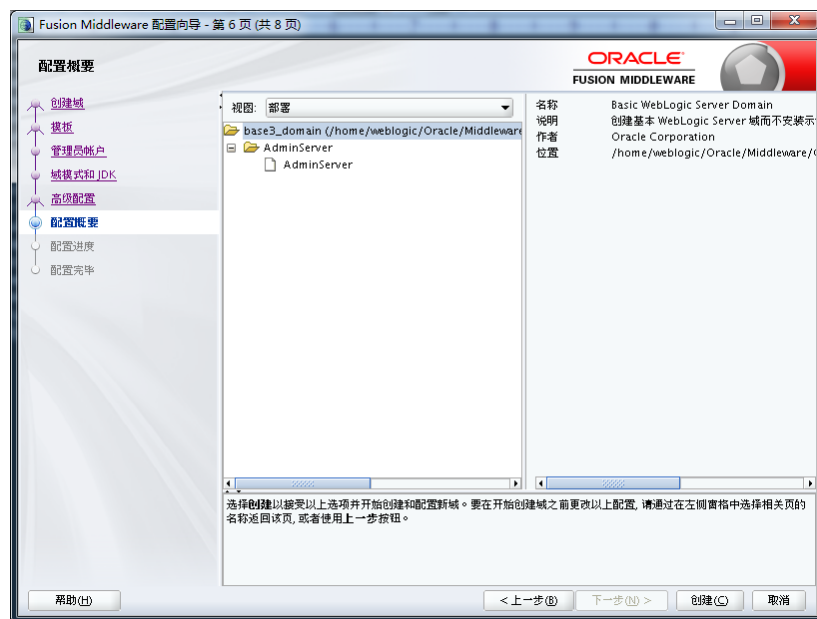
此处勾选可以设置端口和集群配置，默认下一步，端口使用7001

图 3-7 配置向导 5



单击创建开始创建新城

图 3-8 配置向导 6



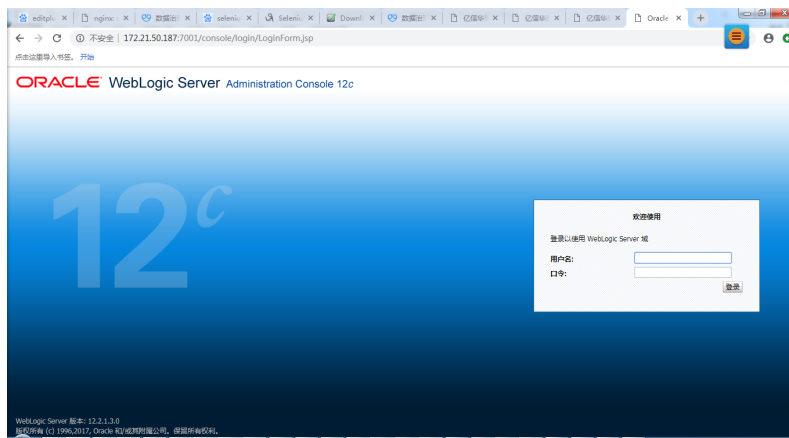
通过startWebLogic.sh(cmd)启动weblogic, 直到窗口中出现listening on port 7001 (端口号根据实际配置会不一样), 表明服务已经成功启动。这时就可以在IE浏览器里键入http://localhost:7001/console, 进入Weblogic控制台, 输入您的管理员名和密码, 即可进行控制台管理。

3.1.2.2.2 部署睿码平台

Weblogic 12c上部署

Weblogic管理控制台访问地址：<http://服务器IP:7001/console>。在IE浏览器里键入该地址，进入Weblogic控制台登录页面，输入用户名和密码进行登录。

图 3-9 登录



登录成功后，就出现Weblogic的控制台管理界面，如下：

图 3-10 管理界面



单击“部署”节点，

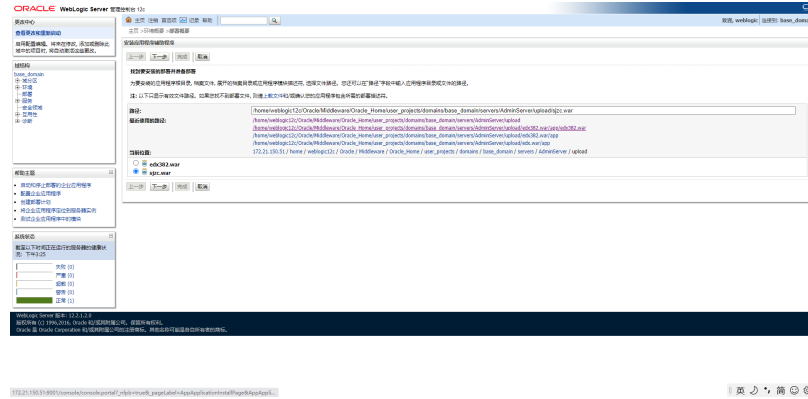
图 3-11 部署



单击安装按钮

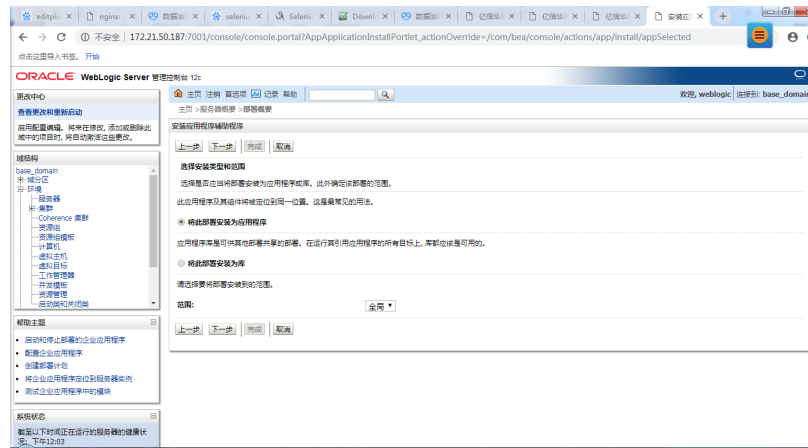
选择服务器的war包，选择下一步

图 3-12 下一步 1



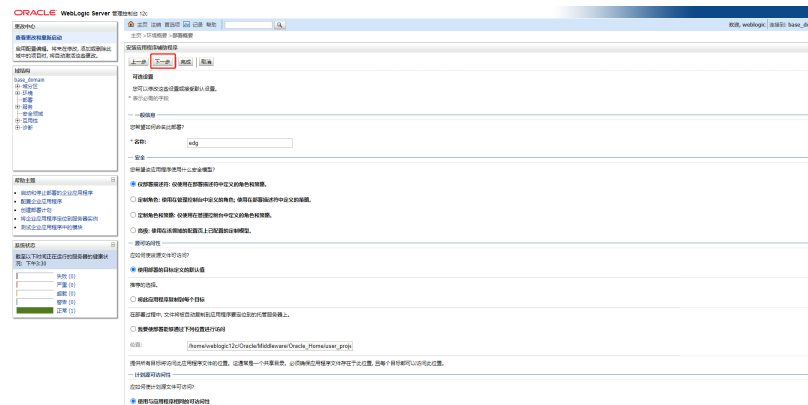
选择下一步

图 3-13 下一步 2



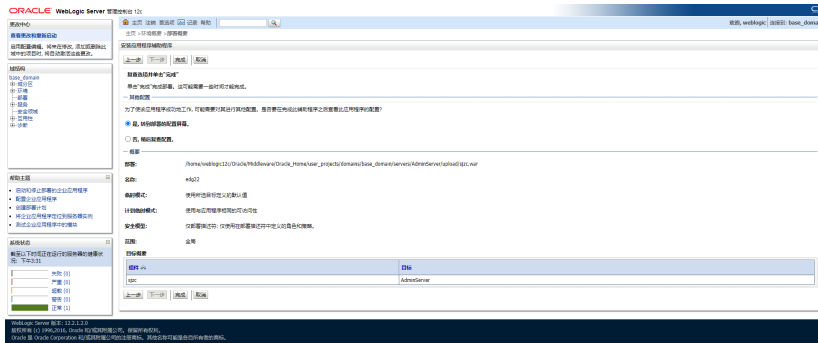
单击“下一步”按钮。

图 3-14 下一步 3



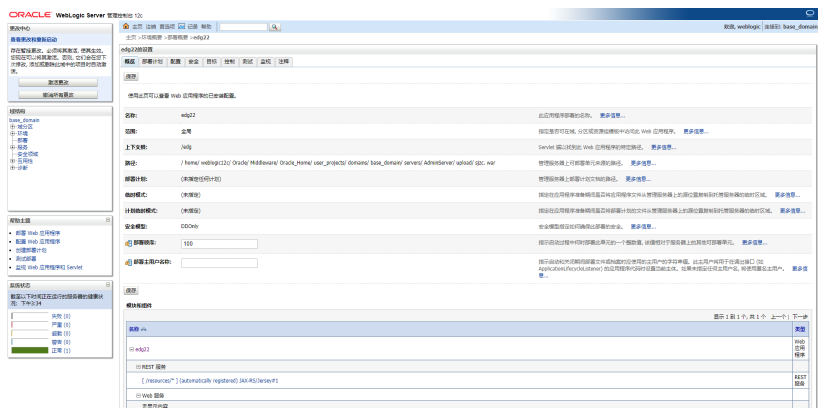
单击“完成”按钮。

图 3-15 下一步 4



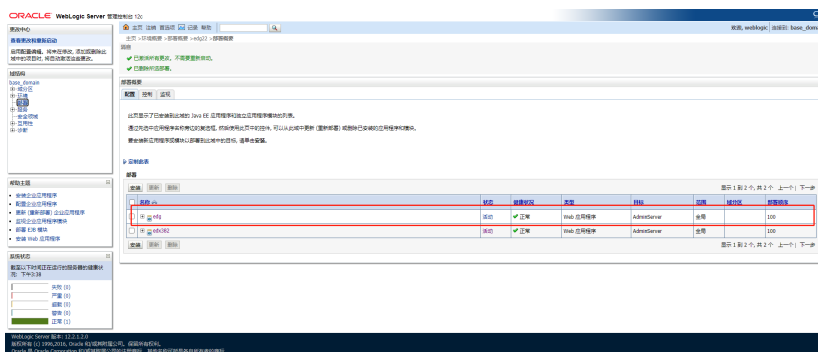
单击“保存”按钮

图 3-16 保存



部署完成

图 3-17 完成



参数设置

在startWebLogic.sh (cmd) 中，增加


```
Set JAVA_OPTIONS=-Xmx2048m -Xms256m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -  
Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK (Windows环境)  
JAVA_OPTIONS=" -Xmx2048m -Xms256m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -  
Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK (Linux环境)"
```

其中，设置内存不能低于上述值。

3.1.2.2.3 Weblogic 集群部署

📖 说明

本章节介绍了Weblogic如何配置集群，也介绍了在集群部署方式下，睿码平台应用配置集群。

集群（Cluster）是由两台或多台节点机（服务器）构成的一种松散耦合的计算节点集合，为用户提供网络服务或应用程序(包括数据库、Web服务和文件服务等)的单一客户视图，同时提供接近容错机的故障恢复能力。集群系统一般通过两台或多台节点服务器系统通过相应的硬件及软件互连，每个群集节点都是运行其自己进程的独立服务器。这些进程可以彼此通信，对网络客户机来说就像是形成了一个单一系统，协同起来向用户提供应用程序、系统资源和数据。除了作为单一系统提供服务，集群系统还具有恢复服务器级故障的能力。集群系统还可通过在集群中继续增加服务器的方式，从内部增加服务器的处理能力，并通过系统级的冗余提供固有的可靠性和可用性。

睿码平台支持集群部署，下面以Weblogic11为例讲解如何部署集群，并在集群中部署服务。

创建新的Domain

选择“Toolà Configuration Wizard”，单击“下一步”按钮

图 3-18 配置向导 1



选择安装域源

选择安装“Weblogic Server”，单击“下一步”按钮

图 3-19 下一步



指定域名和位置；输入域的名称和位置

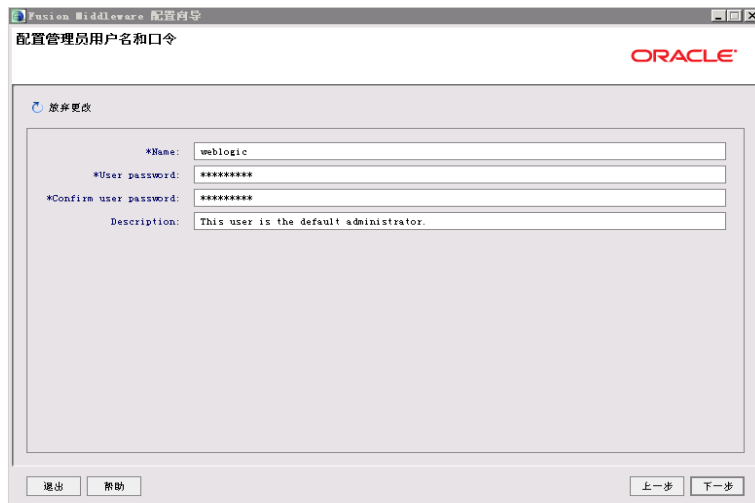
图 3-20 指定域名和位置



配置管理员用户名和密码

输入Weblogic管理员管理服务器 (AdminServer) 的用户名和密码，在此以“Weblogic”作用登录管理服务器的用户名和密码，单击“下一步”按钮

图 3-21 配置管理员用户名和密码



配置域启动模式和JDK

选择“生产模式”和“Sun SDK”后，单击“下一步”按钮

图 3-22 配置服务器启动模式和 JDK



选择管理服务器，受管服务器集群和计算机

选择“是”后，单击“下一步”按钮

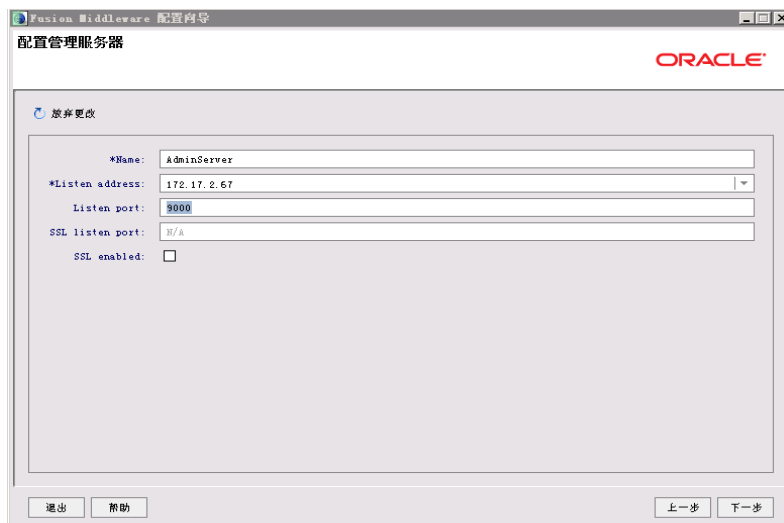
图 3-23 选择可选配置



配置管理服务器

输入Administrator Server的名称，监听地址，监听端口，如果需要SSL支持的话可以在“SSL enabled”后面的复选框上打勾，配置SSL监听端口，单击“下一步”按钮

图 3-24 配置管理服务器

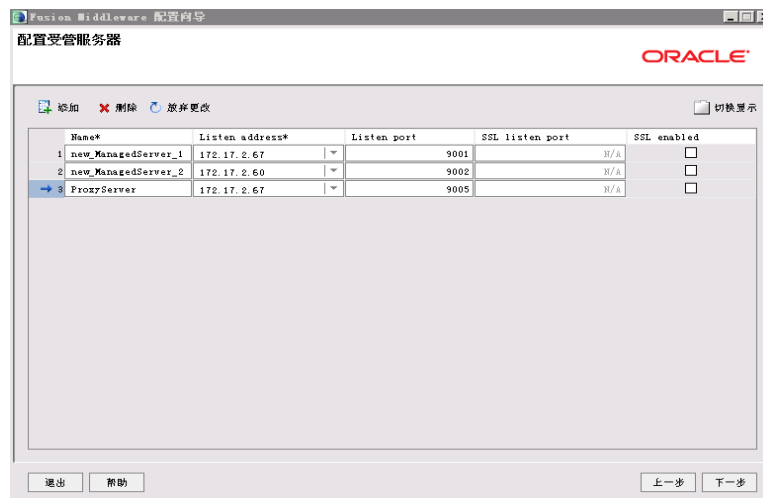


根据需要修改端口（默认为7001）

配置受管理服务器（代理服务器实际上也是一个受管理服务器）

输入Managed Server的名称，监听地址，监听端口，如果需要SSL支持的话，可以在“SSL enabled”复选框上打勾，配置SSL监听端口。可以配置多个Managed Server。单击“下一步”按钮

图 3-25 配置受管服务器



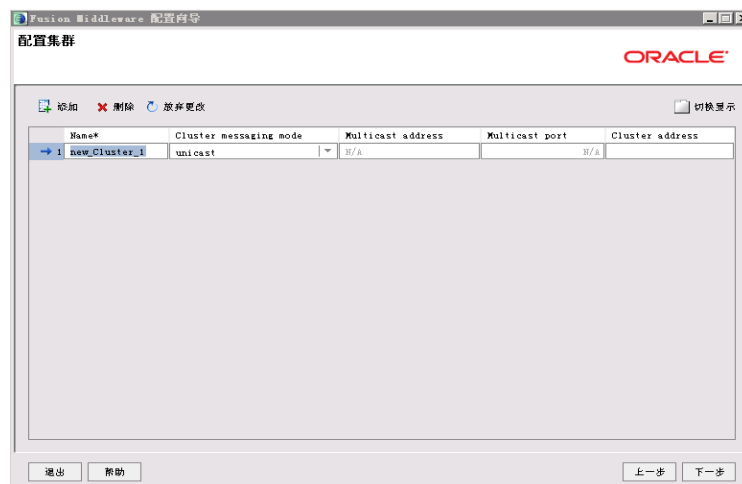
说明

我这台机器的IP是：172.17.2.67，另一台机上要安装受管理服务器的IP地址是：172.17.2.60

配置群集信息

单击“切换显示”，再单击“添加”，采用默认。单击“下一步”按钮。

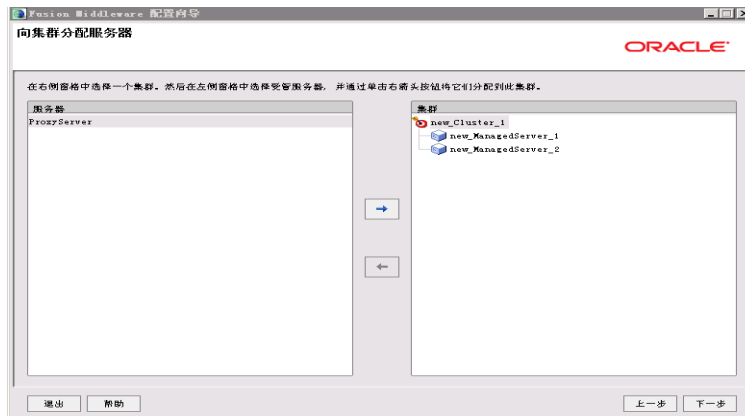
图 3-26 配置集群



将受管理服务器添加到集群中

选择左面列表中的Managed Server，将其添加到右面的列表中，这里代理服务器不要添加到右边，单击“下一步”按钮

图 3-27 向集群分配服务器



创建HTTP代理应用程序

选中“为群集创建Http代理”，myCluster代理服务器选择前面没有加入myCluster的ProxyServer，单击“下一步”按钮

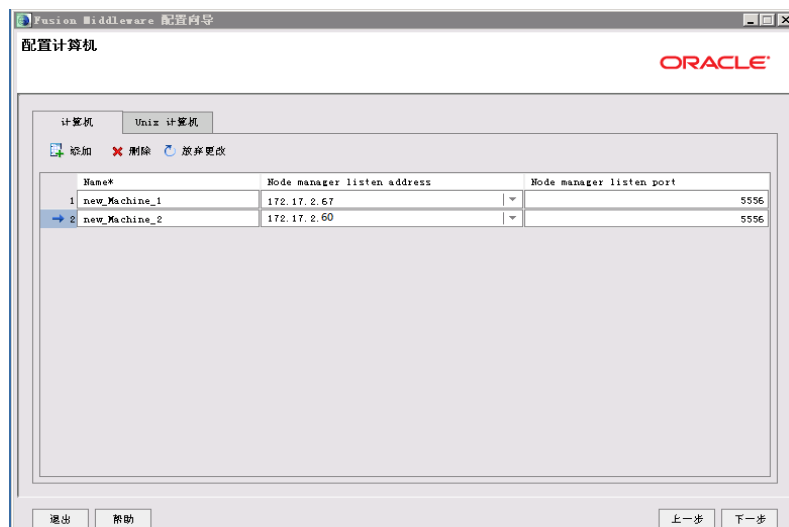
图 3-28 创建 HTTP 代理应用程序



配置计算机

添加两台计算机，按之前地址配置，端口采用默认，单击“下一步”按钮

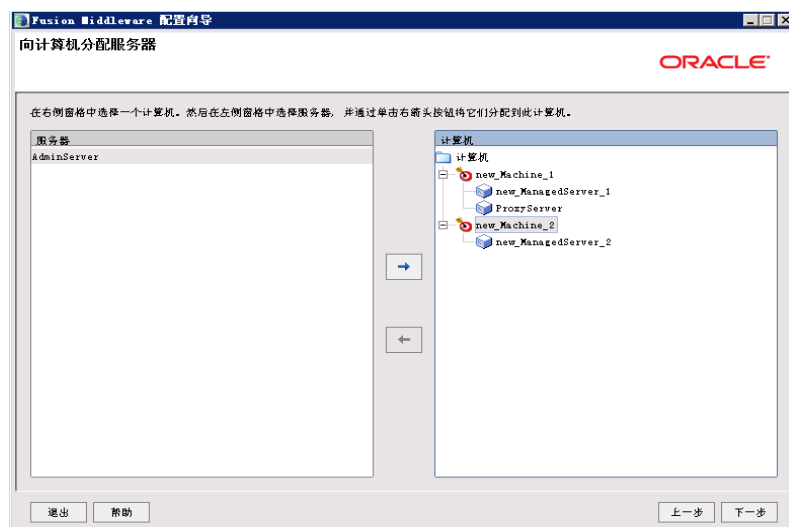
图 3-29 配置计算机



向计算机分配服务器

按照之前规划向计算机分配服务器，如下图示

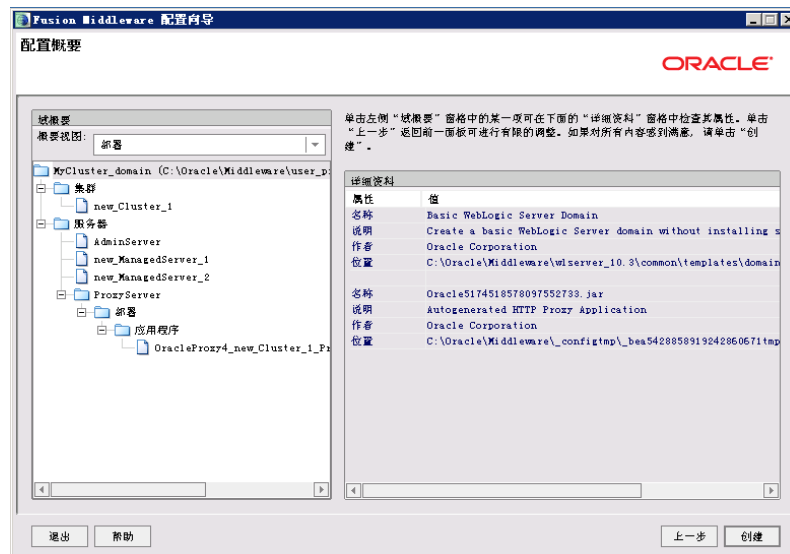
图 3-30 向计算机分配服务器



检查Weblogic域

在此检查前面所做的配置，如果没有问题单击“下一步”按钮

图 3-31 配置概要



创建Weblogic域

单击“创建”按钮，创建无误后，单击“完成”按钮

图 3-32 创建域



集群中辅节点操作

本例是weblogic双机集群，以上操作均在主节点中完成。

集群中的所有 Server 必须位于同一网段，并且必须是 IP 广播(UDP)可到达的；

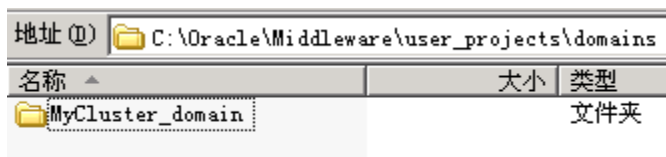
集群中的所有 Server 必须使用相同的版本，包括 Service Pack；

集群中的 Server 必须使用永久的静态 IP 地址。动态 IP 地址分配不能用于集群环境。如果服务器位于防火墙后面，而客户机位于防火墙外面，那么服务器必须有公共的静态 IP 地址，只有这样，客户端才能访问服务器；

辅节点weblogic安装和主节点相同版本，安装路径相同，过程略。辅节点具体操作如下：

1. 将主节点创建好的domain文件夹复制到辅节点机器，如下图

图 3-33 地址



2. 修改辅节点文件C:\Oracle\Middleware\wlserver_10.3\common\nodemanager\nodemanager.domains

添加如下内容并保存（实际环境中主要domain名称和路径）：

```
MyCluster_domain=C:\\Oracle\\Middleware\\user_projects\\domains\\MyCluster_domain
```

启动集群服务

一般常用两种启动方式，一种是通过编辑脚本来启动，一种是启动nodemanager和主节点控制台后通过界面来启动，推荐使用后者。

1. 编辑脚本来启动

编辑加密文件

在域目录servers/AdminServer/security、servers/ManagedServer_1/security、servers/ManagedServer_2/security、servers/proxyServer/security文件夹下（如果没有需新建）建立boot.properties文件，内容为：

```
username=Weblogic  
password=Weblogic1
```

编辑启动管理服务器的文件

在刚才创建的域目录下的bin目录中创建startAdminServer.cmd，内容为：

```
@ECHO OFF  
@REM 启动管理服务器.  
start .\startWeblogic
```

编辑启动代理服务器的文件

在创建的域目录下的bin目录中创建startProxyServer.cmd，内容为：

```
@ECHO OFF  
@REM 启动代理服务器.  
@REM 使用startManagedWeblogic命令启动Managed Server，它的命令格式是：  
@REM startManagedWeblogic.cmd Managed Server的名称 Administrator Server地址  
@REM startManagedWeblogic 受管理服务器名称 管理服务器URL  
start .\startManagedWeblogic ProxyServer http://90.0.13.10:7001
```

编辑启动受管理服务器(ManagedServer_1)的文件

在刚才创建的域目录下的bin目录中创建startManagedServer_1.cmd，内容为：

```
@ECHO OFF  
@REM 启动受管理服务器ManagedServer_1.  
@REM 启动Managed Server  
@REM 使用startManagedWeblogic命令启动Managed Server，它的命令格式是：  
@REM startManagedWeblogic.cmd Managed Server的名称 Administrator Server地址  
@REM startManagedWeblogic 受管理服务器名称 管理服务器URL  
start .\startManagedWeblogic ManagedServer_1 http://90.0.13.10:7001
```

说明

使用startManagedWeblogic命令启动Managed Server，它的命令格式是：
startManagedWeblogic.cmd Managed Server的名称 Administrator Server地址
如果有多个受管理节点，可以做多个受管理服务器。

2. 启动nodemanager和主节点控制台通过weblogic控制台来启动

主节点操作如下：

启动weblogic控制台，和前面介绍启动方式一样，此处略

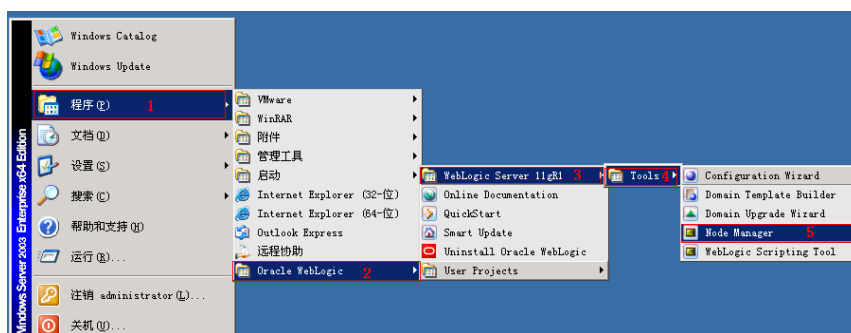
启动Node Manager

辅节点操作如下：（辅节点只需要启动Node Manager）

启动Node Manager

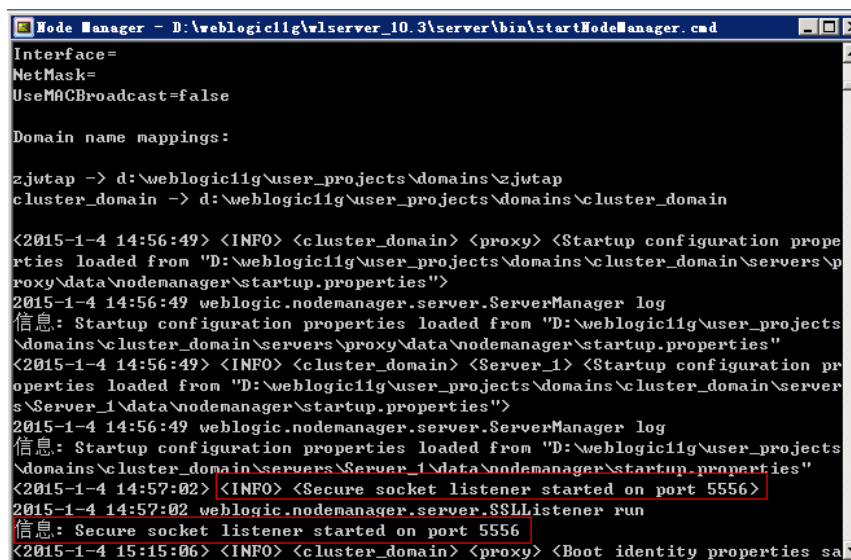
启动NodeManager方法如下：按如下图1-5顺序启动

图 3-34 启动



出现下图，5556端口处于监听表示启动成功。

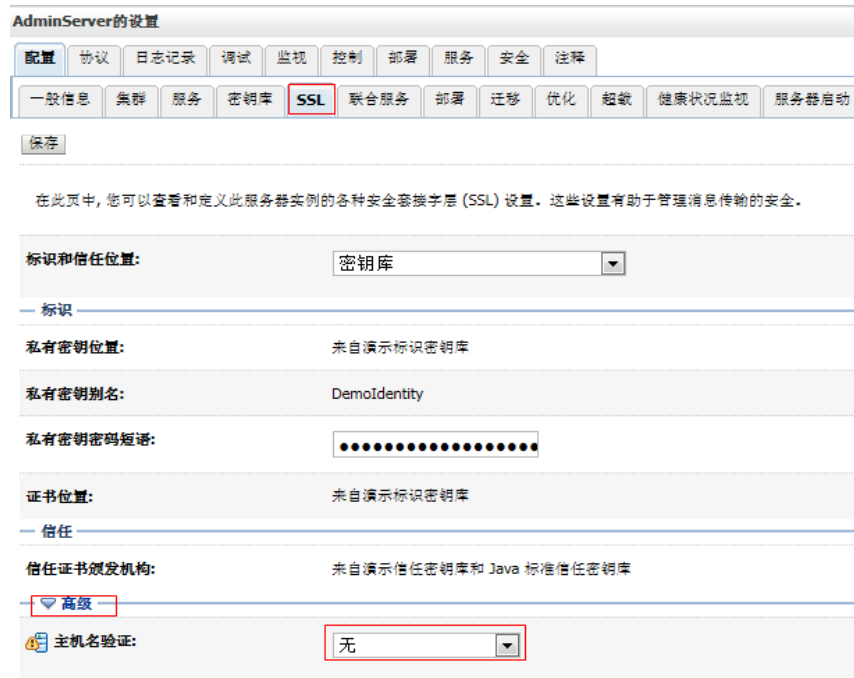
图 3-35 图示



此时，登录到weblogic控制台（<http://172.17.2.67.9000/console>）

修改AdminServer的设置SSL 主机验证为无

图 3-36 AdminServer 的设置



修改完毕后重新启动weblogic控制台，查看计算机状态，确保为可访问

图 3-37 Plachine_1 的设置



然后在服务器界面，依次启动代理和节点（即主控和受控节点）

切换到控制页面，一次勾选节点服务器和代理，单击“启动”，启动成功后如下图 RUNNING 状态，启动时间视机器配置而定，此时整个集群服务启动完毕。

图 3-38 启动完毕



内存调整

修改域目录下\bin下的setDomainEnv.cmd（linux下commEnv.sh）对应JDK的内存。

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，一般建议在32bit环境下最大内存不要超过2G，在64bit环境下不要超过系统可用内存的70%。

注意

某些JDK版本限制最大堆内存，不支持分配超过1.5G的内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

Weblogic建议最大最小内存分配一样大小。

Session设置

Application.properties

在war包中配置文件web-inf\classes\application.properties，添加如下内容：

```
server.session.timeout=1800
```

此例表示Session将在30分钟后过期

如果不配置，则默认30分钟

Weblogic.xml

在war包中找到web-inf\weblogic.xml，设置Weblogic特有部署描述符Weblogic.xml的<session-descriptor>元素的TimeoutSecs属性。这个值以秒为单位：

```
<session-descriptor>  
  <session-param>  
    <param-name>TimeoutSecs</param-name>  
    <param-value>3600</param-value>  
  </session-param>  
</session-descriptor>
```

默认值是3600秒

服务器端压力较大时建议减小此时间。

JNDI配置

登录weblogic控制台，进入服务->JDBC->数据源

单击【新建】按钮

填写JNDI名称，单击【下一步】

默认下一步

默认下一步

填写数据库“实例名”、“数据库服务器ip”、“数据库端口”、“连接的数据库用户名”、“密码”，单击【下一步】按钮

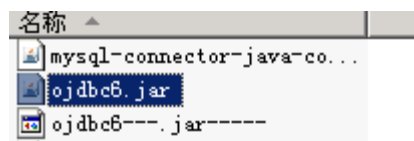
单击【测试配置】，确认连接成功，单击【下一步】按钮

选择需要使用该数据源的server节点，单击【下一步】按钮

配置好JNDI后需要添加jar包到weblogic的oracle\Middleware\wlserver_10.3\server\lib的目录下，需添加：oracle_ojdbc14-1.0.0.jar，它是数据库驱动包，该包在睿码平台

war包的\WEB-INF\lib路径下（jar包名视睿码平台版本而定），添加后需要改名为ojdbc6.jar，并且注销weblogic自带的ojdbc6.jar文件。

图 3-39 名称



最后还需要到睿码平台工作目录conf下修改jdbc.conf文件，在文件中增加第三方连接池参数datasource3=JDBC_QBI（其中JDBC_QBI为JNDI名称）。

配置完成启动weblogic，管理员admin登录查看数据库连接池基本信息。

JNDI数据库配置参数调优

登录weblogic控制台，单击进入“服务 > JDBC > 数据源”。单击需要调优的jdbc数据源名称。

进入数据源配置页面，选择“配置 > 连接池”选项卡。

在该页面可以修改“初始容量”（数据库空闲连接池）、“最大容量”（数据库最大连接数），单击下面的“高级”设置里面还有“最大等待连接数”。修改前需要单击“锁定并编辑”，本节不做详细讲解。

3.1.2.3 Tongweb6.1 上部署睿码平台

3.1.2.3.1 部署环境配置

安装前准备

安装JDK1.8或以上版本，具体参照第三章中1.1和2.1步骤

开始安装

使用root用户登录，进入安装包文件所在的路径，先使用命令：`chmod +x Install_TW6.1.5.17_Enterprise_Linux.bin`，为安装包文件增加可执行权限。

使用命令执行文件：`./Install_TW6.1.5.17_Enterprise_Linux.bin`，安装过程简单，此处省略。

启动和停止

安装成功后进入安装目录，如/data/TongWeb6.1/bin/，通过`nohup ./startserver.sh &`来启动服务；服务启动后，通过命令`./stopserver.sh`来停止服务

管理控制台

已经正确启动了Tongweb应用服务器后，在浏览器中输入URL：`http://ip:9060/console`；

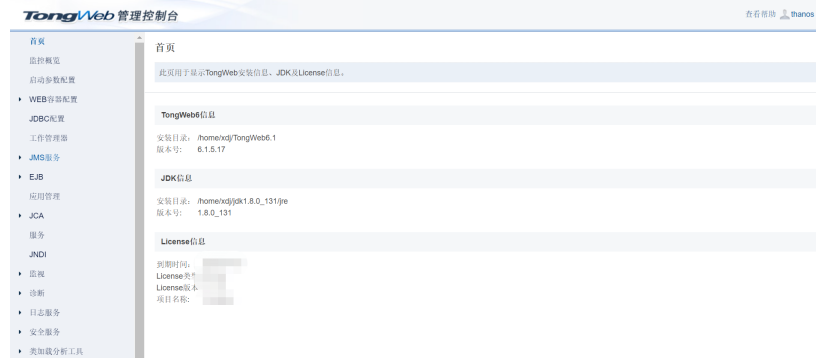
ip：指安装Tongweb应用服务器软件的计算机的IP地址

9060：是Tongweb应用服务器默认的http端口

3.1.2.3.2 部署睿码

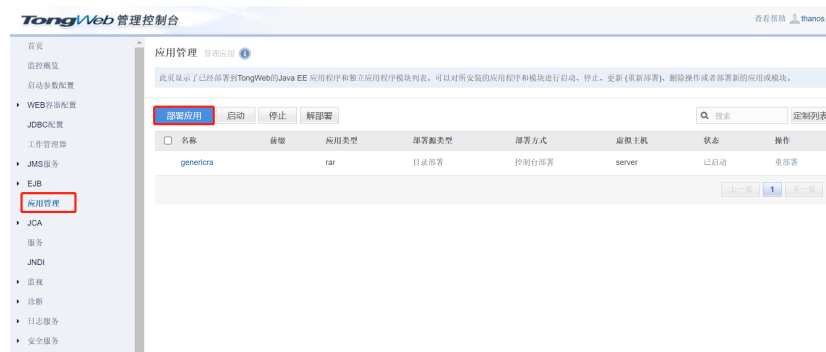
进入管理控制台，界面如下图：

图 3-40 管理控制台



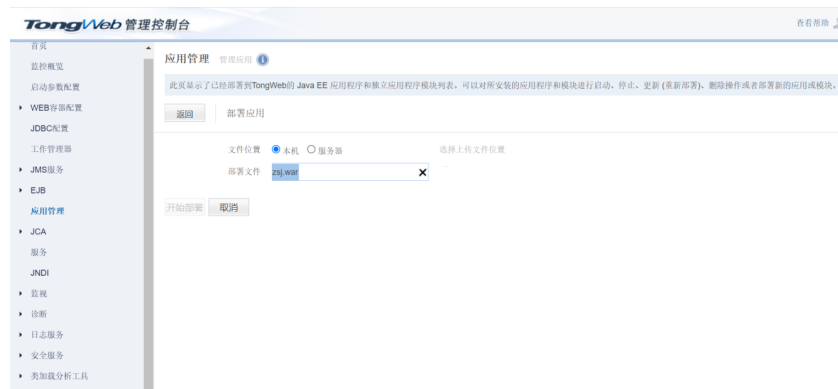
单击“应用管理”，进入应用部署界面，单击“部署应用”，进入下一界面

图 3-41 下一界面



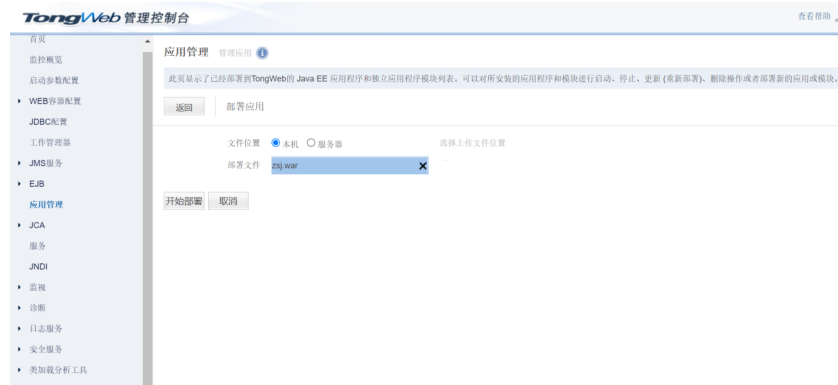
单击“选择文件”，选择所要部署的产品

图 3-42 选择



上传完毕后，单击“开始部署”

图 3-43 开始部署



单击下一步

图 3-44 下一步



图 3-45 下一步 2



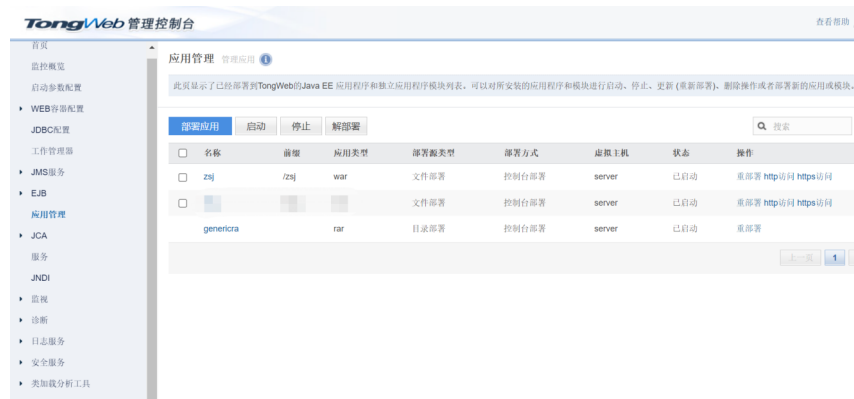
单击“完成”等待部署完毕

图 3-46 部署完毕



成功部署，状态为已启动，单击“http访问”即可进入产品访问地址

图 3-47 访问



3.1.2.4 金蝶 10 上部署睿码平台

3.1.2.4.1 部署环境配置

本章节主要讲述在linux环境下如何安装及配置金蝶中间件。

安装应用服务器

确保所登录用户对AAS-V10.0.zip有可读写权限(在终端中用命令 `ls -l AAS-V10.0.zip`查看)，可通过以下命令将AAS-V10.0.zip解压缩到安装目录：

可参考以下命令(具体以实际为准)：

`unzip /data/AAS-V10.0.zip`其中：

`/data/AAS-V10.0.zip`：表示金蝶10.0上传到服务器上的安装包。

启动服务器

进入/data/AAS-V10/ApusicAS/aas/bin目录，输入startserv命令并执行即可。

可参考以下命令(具体以实际为准)：

```
cd /data/AAS-V10/ApusicAS/aas/bin
```

前台启动


```
sh ./ startserv
```

后台启动

```
nohup ./startserv &
```

后台启动可以tail -f nohup.out 查看输出的日志

管理控制台

已经正确的启动了Apusic应用服务器后，在浏览器中输入以下URL：https://ip:6848/

ip：指安装Apusic应用服务器软件的计算机的IP地址

6848：是Apusic应用服务器默认的http端口

访问成功后出现如下图所示的页面：

图 3-48 访问成功界面



Apusic 管理控制台的初始用户名为 admin，密码为上面配置管理端密码。

停止服务器

```
cd /data/AAS-V10/ApusicAS/aas/bin，进入安装目录下
```

```
执行./stopserv关闭Apusic，如果无法关闭，直接kill -9 PID
```

3.1.2.4.2 部署睿码

📖 说明

war包需要删除WEB-INF\lib\eweb-1.3.1-SNAPSHOT.jar\META-INF\services
\javax.servlet.ServletContainerInitializer文件

进入控制台管理界面，单击右边“应用程序”，单击“部署”

图 3-49 部署 1

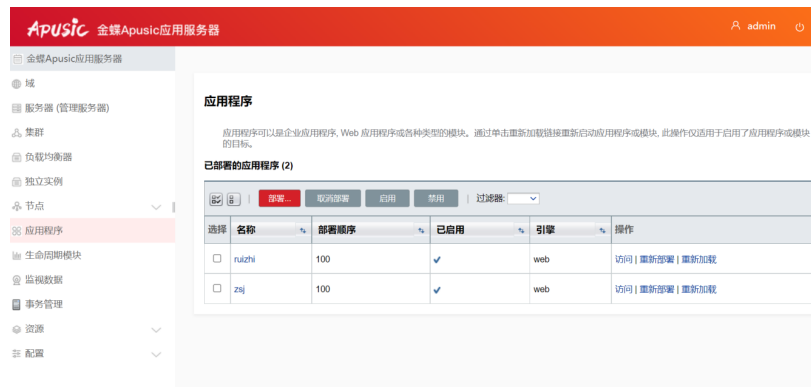
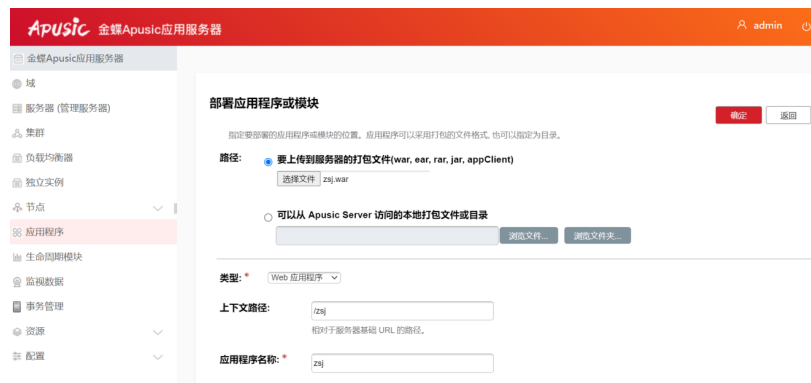


图 3-50 部署 2



图 3-51 部署 3



3.1.2.5 TAS2.8.5 上部署睿码平台

3.1.2.5.1 部署环境配置

安装前准备

安装JDK1.8或以上版本，具体参照前面章节步骤

开始安装

使用root用户登录，进入安装包文件所在的路径，

执行命令：./java -jar tas-installer-2.8.5.jar -console，安装过程简单，此处省略。

启动和停止

安装成功后进入安装目录，如/data/tas2.8.5/bin，通过./ StartTAS.sh来启动服务
服务启动后,通过命令./ StopTAS.sh来停止服务。

管理控制台

已经正确启动了Tas应用服务器后，在浏览器中输入以下URL：http://ip:8081/tas-console

ip：指安装Tas应用服务器软件的计算机的IP地址

8081：是Tas应用服务器默认的http端口

3.1.2.5.2 部署睿码

进入管理控制台，单击左侧的“发布应用程序”

图 3-52 管理控制台 1



第一种方式：单击“上传文件”，选择需要上传的文件，说明：文件小于200M时，选择上传war文件

图 3-53 管理控制台 2



第二种方式：文件大于200M时，选择服务器上的war文件

图 3-54 管理控制台 3



这里使用第二种方式

图 3-55 管理控制台 4



单击确定，开始部署

图 3-56 管理控制台 5



成功部署之后，应用程序列表中会出现刚刚部署的应用程序，单击访问即可进入应用访问地址

图 3-57 管理控制台 6



3.1.2.6 Docker 上部署睿码平台

制作 Dockerfile

Dockerfile是一个包含用于组合映像的命令的文本文件。可以使用在命令行中调用任何命令。Docker通过读取Dockerfile中的指令自动生成映像。可以参照以下方式制作Dockerfile。

```
FROM centos:7
ENV JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF8 -Xmx2048m -Xms256m -XX:MaxMetaspaceSize=512m -XX:MetaspaceSize=256m -Duser.timezone=Asia/Shanghai -Desen.mdm.workdir=/usr/local/workdir -Djava.awt.headless=true"
ENV JAVA_HOME /usr/local/java
ENV JRE_HOME /usr/local/java/jre
ENV PATH /usr/local/java/bin:$PATH
WORKDIR /usr/local/
ADD jdk-8u131-linux-x64.tar.gz /usr/local
ADD apache-tomcat-8.5.12.tar.gz /usr/local
RUN mv jdk1.8.0_131 java && mv apache-tomcat-8.5.12 tomcat
COPY mdm.war /usr/local/tomcat/webapps
EXPOSE 6060
ENTRYPOINT ["/usr/local/tomcat/bin/catalina.sh","run"]
```

说明

jdk-8u131-linux-x64.tar.gz、apache-tomcat-8.5.12.tar.gz、mdm.war、Dockerfile这几个均放在同一路径下。

构建镜像

1. 使用当前目录的 Dockerfile 创建镜像

```
docker build -t mdm:v1 .
```

-t: 镜像的名字及标签

2. 通过 -f Dockerfile 文件的位置

```
docker build -f /opt/Dockerfile -t mdm:v1 .
```

上下文路径，是指 docker 在构建镜像，有时候想要使用到本机的文件（比如复制），docker build 命令得知这个路径后，会将路径下的所有内容打包。

```
Successfully built 5d113977d755
Successfully tagged mdm:v1
```

构建过程最后出现上面显示，即代表镜像构建成功。

启动容器

```
docker run -d -p 6060:8080 -v /opt/mdmworkdir:/usr/local/workdir mdm:v1
```

- d 服务形式启动
- p 端口映射, 宿主机6060端口: docker容器8080端口, 6060可随意修改成未被占用的端口
- v 工作目录路径,宿主机/opt/mdmworkdir: 容器/usr/local/workdir

访问服务器

访问地址: `http://ip:6060/mdm`, ip为宿主机ip地址, 6060为宿主机映射的端口, mdm为war包名称, 根据实际修改。

工作目录及注册说明

制作Dockerfile时, 环境变量中的-Desen.mdm.workdir=/usr/local/workdir是用来设置容器内部的产品工作目录, 启动时再把工作目录映射到宿主机上。

注册时通过注册界面导入注册申请文件, 不要直接把注册申请文件放在宿主机工作目录下。

相关问题

报错无法获取服务器硬件信息, 此问题是由于产品在注册时需要获取硬件信息, 但是容器内部没有相关命令, 须在构建镜像时加上一行RUN yum install -y net-tools dmidecode

3.1.3 产品 WAR 包集群部署

集群场景说明

集群4个节点部署在两台服务器上, 对应的服务器IP、端口、工作目录如下。Redis单独一台服务器部署。以双机双节点的配置睿码平台集群为例, 说明在睿码平台中如何配置集群。

表 3-6 服务器 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine1	Server1 (主控)	192.168.1.101	9001	/work_dir/rootdir1/
Machine1	Server2	192.168.1.101	9002	/work_dir/rootdir2/
Machine2	Server3	192.168.1.102	9003	/work_dir/rootdir3/
Machine2	Server4	192.168.1.102	9004	/work_dir/rootdir4/

表 3-7 服务器 2

物理机器	安装软件	IP地址	端口号
Machine3	Redis	192.168.1.103	6379

工作目录指定

涉及到集群的工作目录指定，每个节点必须指定一个单独的工作目录，而且多节点不能同时使用同一个工作目录。

指定工作目录需要在中间件启动文件设置，在Server1的启动文件中设置：

-Desen.mdm.workdir =/work_dir/rootdir1

在Server2的启动文件中设置：

-Desen.mdm.workdir =/work_dir/rootdir2

在Server3的启动文件中设置：

-Desen.mdm.workdir =/work_dir/rootdir3

在Server4的启动文件中设置：

-Desen.mdm.workdir =/work_dir/rootdir4

JVM 参数设置

表 3-8 JVM 参数设置

节点名称	参数设置	备注
Server1	-Desen.mdm.workdir=/work_dir/rootdir1 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	-Xmx和-Xms根据服务器内存大小设置，64位OS建议分配8G-16G
Server2	-Desen.mdm.workdir =/work_dir/rootdir2 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

Server3	-Desen.mdm.workdir =/work_dir/ rootdir3 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上
Server4	-Desen.mdm.workdir =/work_dir/ rootdir4 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

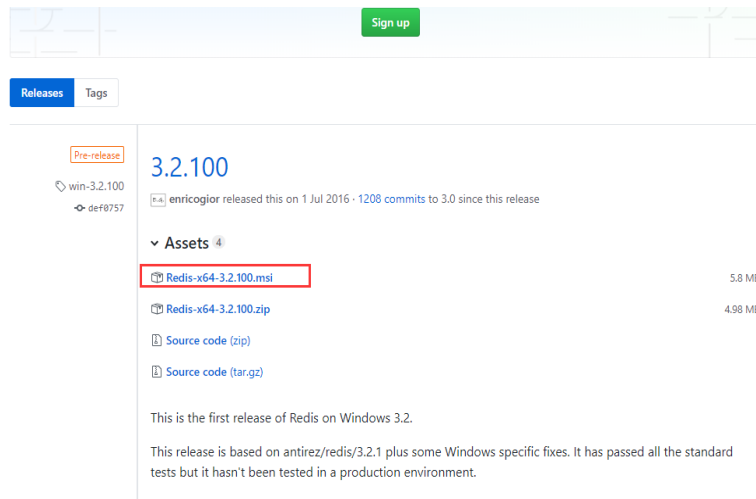
3.1.3.1 Redis 安装

Redis支持32位和64位的Windows和Linux操作系统。

Windows 下安装 redis

下载地址：<https://github.com/MSOpenTech/redis/releases>

图 3-58 下载



根据需要使用合适版本的Redis，下载zip包，下载完成后解压缩，会得到一个redis-server.exe。下载完成后，双击安装，安装完成后，启动服务即可。

使用客户端测试连接：redis-desktop-manager-0.9.3.817.exe

其实Redis在Windows上不需要安装，在命令行下执行这个redis-server.exe，就会启动redis服务，启动后不要关闭这个命令行窗口，关闭它服务就停止了。

Linux 下安装 redis

redis在linux下的安装使用包管理器来安装即可，对于没有外网的环境也可以复制压缩包，解压后手动安装。

安装

```
# wget http://download.redis.io/releases/redis-5.0.5.tar.gz
# tar -zxf redis-5.0.5.tar.gz
# cd redis-5.0.5
# make
# make install
```

启动

编译完成以后，启动 redis-server

```
# cd src
# ./redis-server ../redis.conf &
```

访问redis

用自带的redis-cli客户端工具就可以访问Redis服务器了。

Redis的默认服务端口是6379

```
# redis-cli
127.0.0.1:6379>
```

Redis配置文件说明

Redis的配置文件Redis.conf在Redis的安装目录下，里面有很多关于Redis的设置。用户可以通过直接修改文件更改配置，也可以通过命令像设置key-value数据库一样修改配置

```
config get xxx //获取某一个配置的值
config set xxx xxx //设置某一项配置
config get * 获取所有配置
```

建议直接修改配置文件，因为通过命令修改，重启Redis后就会失效。

表 3-9 主要配置参数列表

参数	值	描述
port	6379	redis的端口
bind	127.0.0.1	绑定的ip地址，默认是127.0.0.1。 如果本机有多个网卡，且限制只能通过其中一个网卡对应的IP才能访问redis时，可绑定该IP，例：bind IP
protected-mode	yes	保护模式，默认为yes，如果要其他网络能访问，改为no即可

参数	值	描述
requirepass	-	登录密码，默认是空
maxclients	0	同一时间能够连接到redis的客户端数量，默认是无限制，0表示不限制
maxmemory	-	redis能够使用的最大内存 单位byte, 默认无限制

3.1.3.2 部署包和产品注册和配置

当在所有节点部署mdm包后，需要分别访问下列地址完成产品的注册。

<http://192.168.1.101:9001/mdm>

<http://192.168.1.101:9002/mdm>

<http://192.168.1.102:9003/mdm>

<http://192.168.1.102:9004/mdm>

在所有4个节点均完成注册后，分别登录所有节点，确认所有节点可以单独访问。

3.1.3.3 集群文件配置

停掉所有服务器，分别在各个工作目录的conf文件夹中放置cluster.properties，其中Server1为主控节点

表 3-10 集群文件配置

节点名称	配置内容
Server 1	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=true cluster.type=redis cluster.group=group1 cluster.nodeid=node1 cluster.name=测试集群节点1 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server 2	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node2 cluster.name=测试集群节点2 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server 3	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node3 cluster.name=测试集群节点3 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server 4	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node4 cluster.name=测试集群节点4 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码,则配置 cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

配置文件中参数说明:

表 3-11 参数说明

参数名	描述	例子
cluster.enabled	是否启用集群	默认不启用
cluster.nodeid	集群节点id	node1

参数名	描述	例子
cluster.name	集群节点名称	测试集群节点1
cluster.type	集群类型（redis： redis集群，netty： netty集群）	redis，默认使用redis
cluster.message.retry	消息发送失败后重发 次数	3
cluster.health.interval	心跳时间间隔（单 位：毫秒）	1500，默认1500
cluster.health.retry	集群节点离线时重试 次数，即连续几次没 有收到心跳则判定节 点离线	3
cluster.heath.rejoin	集群节点离线后重新 加入重试次数，即集 群节点连续几次都受 到心跳消息则判定节 点加入	3
cluster.redis.ip	redis ip	172.21.1.65
cluster.redis.port	redis端口	6379
cluster.redis.password	redis密码，没有密 码可不传	123
cluster.redis.nodes	redis本身的集群内 的其他节点（除上述 配置的其他redis集 群节点），没有可不 配置	172.21.1.66:6379;172.21.1.67:637 9

3.1.3.4 启动服务器、集群检查与注意事项

启动服务器

集群配置文件完成后，先启动server1，待server1启动完成后，可同时启动其它三个服务器。

集群检查

以admin身份登录睿码平台，依次进入“系统设置 > 性能与维护 > 集群信息”，检查如下信息：

- 检查所有配置节点是否存在
- 检查所有节点在线状态

- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

注意事项

- **集群内各服务器版本需保持一致：**只能同一版本的服务器做集群
- **各服务器的数据库配置需一致：**各服务器数据库配置需要保持一致，其中url中ip地址也要一致
- **各服务器redis配置需一致：**各服务器配置的redis相关参数需要是一致的，如redis ip、端口、密码

3.1.3.5 集群扩容说明

新增N个节点，与原集群各节点配置保持一致，连接同一个Redis，即可达到集群扩容目的。

例如新增物理机器Machine4，节点名称Server5,IP地址192.168.1.103，端口号9005

表 3-12 物理机器

物理机器	节点名称	IP地址	端口号	工作目录
Machine4	Server5	192.168.1.103	9005	/work_dir/rootdir5/

指定工作目录、设置JVM参数后，正常注册睿码平台，然后停掉新增节点服务器，在工作目录的conf文件夹中放置cluster.properties，配置内容为：

表 3-13 配置内容

节点名称	配置内容
Server 5	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node5 cluster.name=测试集群节点5 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

📖 说明

该配置文件中cluster.redis.ip与原集群各节点保持一致。

然后启动新增节点服务器，启动成功后以admin身份登录睿码平台，依次进入“系统设置 > 性能与维护 > 集群信息”，检查如下信息：

- 检查所有配置节点是否存在

- 检查所有节点在线状态
- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

3.1.4 初始化服务器

通过以上章节了解服务器如何部署在各web应用服务器上后，您可以通过如下步骤完成睿码初始化工作。

1. 输入登录网址，如：<http://127.0.0.1:8080/mdm>，进入睿码服务器配置界面

图 3-59 配置界面



2. 勾选同意以上协议，单击“下一步”

图 3-60 下一步



设置工作目录，填写路径后单击检测按钮，保证工作目录可用

3. 单击“下一步”，设置数据库

图 3-61 设置数据库

此处可以单击测试，测试数据库链接。

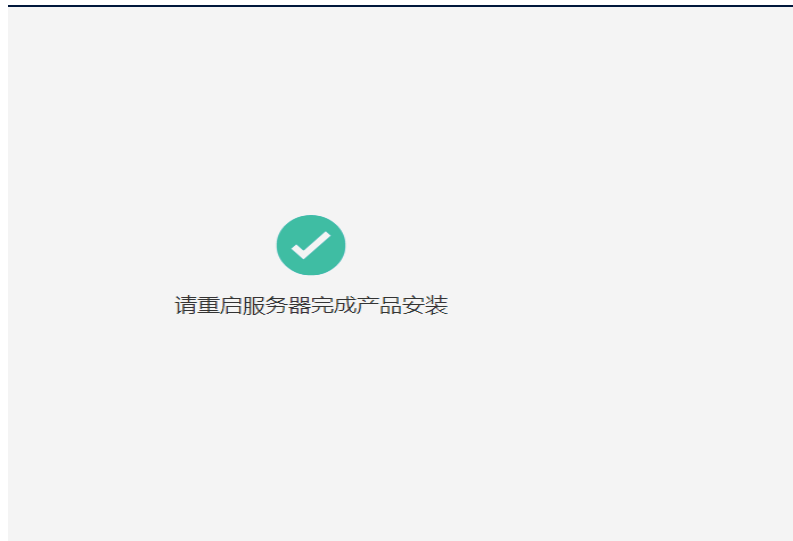
4. 单击“下一步”，注册服务器

图 3-62 注册服务器

此处，需要填写用户名称，项目名称，联系电话和联系邮箱等，然后生成注册申请，将生成的注册申请文件导出，进行注册码申请，由具有审批权限的人员将生成的许可文件发给部署人员，然后导入许可文件。

5. 单击“下一步”，开始安装

图 3-63 开始安装



安装完成后请重启中间件

6. 进入服务器的登录界面

图 3-64 登录界面



管理员admin，初始密码为admin。

3.1.5 系统诊断

很多问题其实都是部署的时候没有正确设置某些参数，或者是没有按要求修改配置文件，往往在遇到异常时很难定位到这些错误，睿码系统能自动诊断这些文件，并加以提示。

在管理员登录界面后单击系统诊断提示的警告会出现如下诊断界面

图 3-65 诊断界面



系统进行了下列检测

表 3-14 检测

诊断项	检测内容 (中文)	The detection content (English)
时区设置	必须是Asia/Shanghai, 否则提示错误	必须是Asia/Shanghai, 否则提示错误
file.encoding设置	必须是"UTF-8","UTF8","GB18030","GBK",否则提示错误	必须是"UTF-8",否则提示错误
sun.jnu.encoding设置	同上	同上
操作系统字符集	检查文件名或文件内容是否支持GBK编码, 不支持提示错误	检查文件名或文件内容是否支持"UTF-8"编码, 不支持提示错误
java.awt.headless设置	windows不检查, linux检查或未设置提示错误	windows不检查, linux检查或未设置提示错误
最大内存设置	32位要求大于1024M, 64位要求大于2048M	32位要求大于1024M, 64位要求大于2048M
可用内存	<%5 系统已几乎无可用内存 <30% 系统可用内存偏低 >=30% 通过	<%5 系统已几乎无可用内存 <30% 系统可用内存偏低 >=30% 通过
可用字体	必须有宋体字体, 否则不能通过	-
注册信息	系统剩余使用天数	-

3.1.6 疑难问题处理方案及数据库权限分配

如何指定工作目录

按照优先级由低到高排列

1. 通过界面初始化
windows环境在注册表下面，Linux环境在当前执行用户home目录下的.java下
2. 通过java环境变量指定
在启动参数中加上-Desen.mdm.workdir=/opt/workdir/mdm
3. 在war包中指定
在war包中的WEB-INF\classes\ application.properties，
指定esen.mdm.workdir=/opt/workdir/mdm



注意

最前方的#代表注释

环境乱码如何配置

环境部署好后，如果面包屑出现乱码现象，将tomcat里面的参数修改即可。

路径为apache-tomcat-8.5.12/conf/server.xml，在端口号后面加上URIEncoding="UTF-8"，改变tomcat的字符集，如图所示；

图 3-66 环境乱码如何配置

```
<!-- Editable user database that can also be used by
UserDatabaseRealm to authenticate users
-->
<Resource name="UserDatabase" auth="Container"
type="org.apache.catalina.UserDatabase"
description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
<!-- A "Service" is a collection of one or more "Connectors" that share
a single "Container" Note: A "Service" is not itself a "Container",
so you may not define subcomponents such as "Valves" at this level.
Documentation at /docs/config/service.html
-->
<Service name="Catalina">
<!--The connectors can use a shared executor, you can define one or more named thread pools-->
<!--
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
maxThreads="150" minSpareThreads="4"/>
-->
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" URIEncoding="UTF-8"/>
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
-->
<!-- Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the BIO implementation that requires the JSSE
style configuration. When using the APR/native implementation, the
OpenSSL style configuration is required as described in the APR/native
documentation -->
<!--
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
-->
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

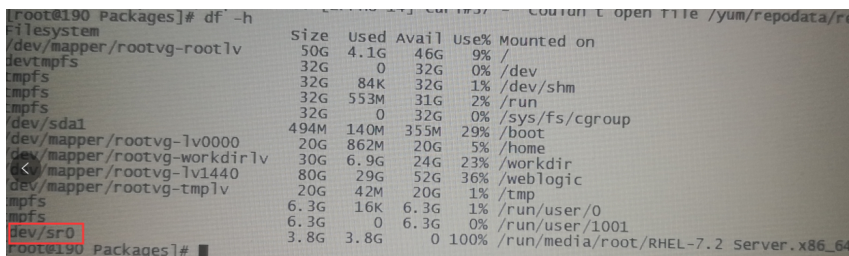
Redis 部署相关

问题现象：由于某些项目服务器的操作系统是最小安装的，安装redis时会提示缺少gcc等rpm包

解决办法：

1. 如果处于外网环境（能ping通baidu.com），可直接执行命令进行安装
yum install -y gcc-c++
如果缺少其他依赖包，同样可以用yum进行安装
Make报错问题解决后，先清理上一次make产生的文件再次make
make distclean
make
2. 如果处于内网环境，则需要如下几步配置yum源实现yum安装：
 - a. 给服务器所在虚拟机或物理机插入相同版本的操作系统光盘
 - b. 在操作系统层面将光盘挂载在某一个目录下，如下图

图 3-67 挂载



```
# mount /dev/sr0 /mnt/cdrom
```

#/dev/sr0为光盘路径，/mnt/cdrom为挂载路径

- c. 在/etc/yum.repos.d目录下备份原有文件，新建local.repo文件，后缀必须为.repo，增加如下内容：

```
[local] #必须，唯一的repo名称  
name=local #自定义名称  
baseurl=file:///mnt/cdrom #光盘mount后的路径  
enable=1 #启用该repo，0为禁用  
gpgcheck=0 #不检查rpm包来源
```

睿码目前支持的系统库

Oracle库：Oracle11g、Oracle12c

PG库：PostgreSQL 10.6

达梦数据库：达梦7、达梦8

Mysql数据库：mysql8

📖 说明

mysql8作为系统库时，需要更换mysql8驱动，把自带的mysql5驱动删除掉，url地址加一个固定参数&allowPublicKeyRetrieval=true

睿码目前支持的中间件

Tomcat: tomcat版本需要用9.0.0.M18以上、8.5.12以上的版本

Weblogic: weblogic12c

Tas: tas2.8.5及以上

TongWeb: TongWeb6.1

睿码目前支持的浏览器

Chrome (v77以上)、ie11、360浏览器 (v10以上)

测试通过的软硬件环境

表 3-15 测试通过的软硬件环境

总体测试环境 http://172.21.50.101:8081/ mdm25linux	【数据库】Oracle19c
	【客户端】win7,win10 ie11, chrome
	【中间件】tomcat9, jdk1.8
	【中间件所在的操作系统】CentOS release 7

IPv6 部署

睿码251支持ip v6部署，此处以tomcat为例介绍如何配置

1. 修改tomcat/conf/server.xml文件，找到下图位置，添加address=" ip v6地址”，

图 3-68 位置

```
<Connector port="8080" protocol="HTTP/1.1" address="fd15:4ba5:5a2b:1008:d8b3:df29:dfd2:7776" connectionTimeout="20000" redirectPort="8443" />
```

2. 其余配置和ip v4一样，可参考前面章节
3. 初始化睿码，在数据库设置这里，填写链接地址时，ipv6地址要用[]括起来，即标红部分。如使用oracle11g，则参照下面写法
jdbc:oracle:thin:@//[fd15:4ba5:5a2b:1008:d8b3:df29:dfd2:7776]:1523/orcl
4. 注册成功后，ipv6访问地址也要用[]括起来，如：
http://[fd15:4ba5:5a2b:1008:d8b3:df29:dfd2:7776]:8080/mdm

数据库权限分配

表 3-16 数据库权限分配

数据库	分配权限
Oracle	connect、resource、create any view 说明 oracle12c及以上需要alter user xxx quota unlimited on users

3.2 睿治数据治理平台

3.2.1 安装环境

服务器配置推荐

- **推荐客户：小范围试用和用户量较小的**
用户数小于100，并发用户数小于10，后端待分析数据源单表数据量百万级

表 3-17 配置项目 1

配置项目	单项配置说明
应用服务器	DELL PowerEdge R340或其他同类产品 CPU：英特尔® 至强® E-2254处理器（4C/8T） 内存：16G 硬盘：RAID5 1T 网络：1000Mb/s
数据库服务器	DELL PowerEdge R440或其他同类产品 CPU：英特尔® 至强® E-2274G处理器（4C/8T） 内存：32G 硬盘：RAID5 按需 网络：1000Mb/s
操作系统及软件	操作系统：Windows、CentOS、Red Hat 中间件：tomcat-8.5.12、tomact-9.0.54 数据库：Oracle11g

- **推荐客户：大型企业、省级机构**
用户数小于300，并发用户数小于30，后端待分析数据源单表数据量千万级

表 3-18 配置项目 2

配置项目	说明
应用服务器	IBM X3650或HP PowerEdge R720 CPU：英特尔® 至强®银牌 4208处理器（8C/16T） 内存：32G 硬盘：RAID1 1T 网络：1000M

配置项目	说明
数据库配置	DELL PowerEdge R540或其他同类产品 CPU: 英特尔® 至强®金牌 6226处理器 (12C/25T) 内存: 64G 硬盘: 按需 网络: 1000M
推荐操作系统及软件	操作系统: CentOS6 中间件: tomcat-8.5.12 数据库: Oracle11g

客户端配置

- 客户端(建议配置)

表 3-19 客户端

所属项	说明
CPU	不低于酷睿i3
内存	不小于4G
网络	不低于100KB/s传输速度

- 客户端浏览器

表 3-20 客户端浏览器

操作系统	浏览器
Windows	IE11,Chrome浏览器, 360安全浏览器

程序安装概况

- 常规场景部署

该部署流程为传统部署, 仅需通过中间件部署程序war包。

部署流程:

- 在服务器中安装中间件后, 在中间件上放入应用war包完成单机环境部署, 参考第三章内容;
- 如果需要搭建集群应用环境, 则需要通过redis完成集群环境部署, 参考第四章内容。

- 实时处理场景部署

当需要使用到数据集成模块下实时数据处理功能时, 采用该套部署流程。通过中间件部署程序war包的同时, 还需要部署实时处理服务, 然后根据实际应用情况选择性部署边缘采集节点, 当应用场景中不包含边缘采集服务时, 可不用部署边缘

采集节点，整体部署环境适用于通过批处理组件、交换组件及实时组件创建ETL任务的业务场景。

部署流程：

- a. 在服务器中安装中间件后，在中间件上放入应用war包完成单机环境部署，参考第三章内容；
- b. 如果需要搭建集群应用环境，则需要通过redis完成集群环境部署，参考第四章内容；
- c. 完成实时处理服务的安装部署，参考第五章内容；

3.2.2 安装程序及依赖驱动清单

以下内容为本次安装部署中安装程序及依赖驱动的清单列表，当需要使用到主数据数据集成中的实时数据处理功能时，需根据实际需求安装对应服务器及驱动。

- **安装程序清单**

表 3-21 安装程序清单

安装程序名称	支持的操作系统	说明
esendatagovernanceV2.5.1.war	Linux/windows	程序war包
edi-RTPS-el7.zip	Linux/windows	实时处理服务程序
edi-RTPS-kylin.zip	银河麒麟4.0	实时处理服务程序
edi-edgenodeagent-linux.zip	Linux	边缘节点程序
edi-edgenodeagent-win.zip	Windows	边缘节点程序
edataexchange-mq-linux.zip	Linux	文件传输代理数据源依赖的mq服务
edataexchange-mq-windows.zip	Windows	文件传输代理数据源依赖的mq服务

- **依赖驱动清单**

表 3-22 依赖驱动清单

安装程序名称	放置路径	说明
edataexchange-hadoop依赖的jar包.zip	工作目录lib\edi\cdh_hadoop	大数据组件依赖驱动
hadoop依赖.zip	工作目录lib\edi\hadoop	HDFS数据源依赖jar包
jdbc-petabase6.0.zip	外挂驱动：服务器任意目录 内置驱动：war包\lib	Petabase数据库依赖的驱动，未防止jar包冲突建议采用外挂驱动的方式

安装程序名称	放置路径	说明
hbase所需依赖.rar	缘采集节点flume\lib	边缘采集需要的hbase依赖
hdfs所需依赖.rar	缘采集节点flume\lib	边缘采集需要的hdfs依赖
hive&impala.zip	外挂驱动：服务器任意目录 内置驱动：war包\lib	数据质量使用hive、impala分析库，未防止jar包冲突建议采用外挂驱动的方式
星环驱动.zip	外挂驱动：服务器任意目录 内置驱动：war包\lib	数据质量使用星环分析库，未防止jar包冲突建议采用外挂驱动的方式

*外挂驱动方式：在数据源连接池的扩展属性中添加参数driverFiles，如driverFiles=/opt/hive_jdbc

3.2.3 产品典型部署架构图

单机部署

传统单机部署由数据库服务器、应用服务器以及客户端组成，其中应用服务器需要安装对应中间件，并上传程序war包，并创建对应工作目录。数据库服务器和应用服务器通过JDBC传输数据，应用服务器和客户端之间通过HTTP协议访问。

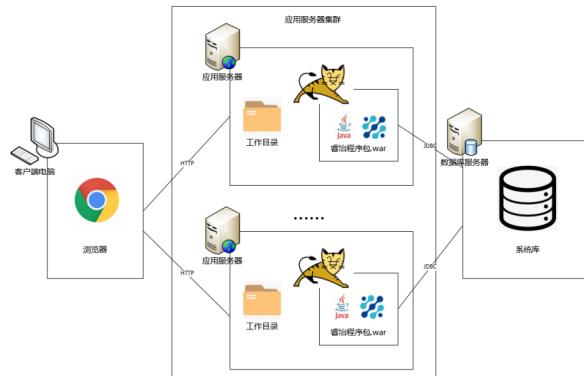
图 3-69 单机部署



集群部署

集群部署由数据库服务器、应用服务器以及客户端组成，其中多个应用服务器设置集群，并可单独设置是否负载均衡。

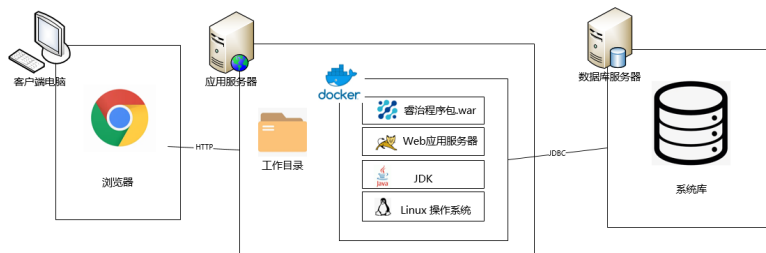
图 3-70 集群部署



Docker 部署

Docker部署由数据库服务器、应用服务器以及客户端组成，其中应用服务器需要部署docker容器，docker容器中包含程序war包镜像、操作系统、jdk等内容，并在应用服务器创建对应工作目录。

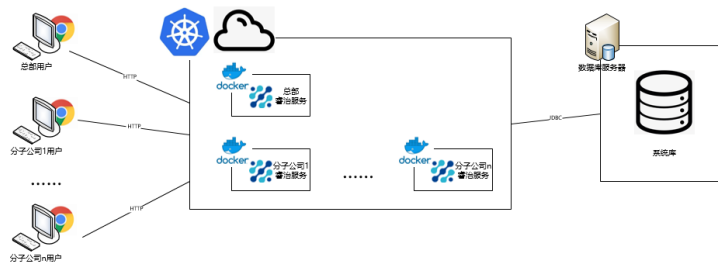
图 3-71 Docker 部署



集团应用部署_多节点统一控制

集团应用部署_多节点统一控制部署由数据库服务器、多个docker部署的服务及不同节点的客户端组成。该部署主要应用的业务场景为集团公司根据业务需要搭建不同服务，但需要统一在一个docker容器进行维护。

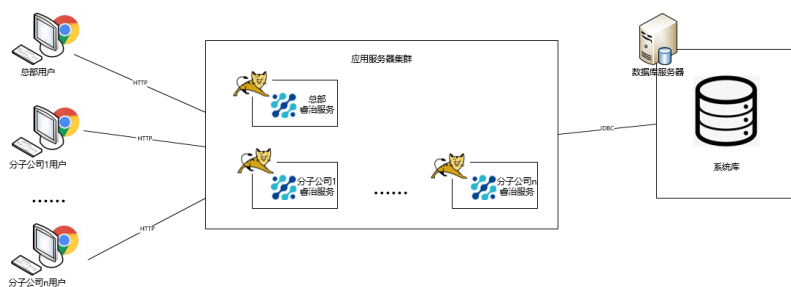
图 3-72 多节点统一控制



集团应用部署_多应用集群

集团应用部署_多应用分别部署由数据库服务器、多个应用服务器以及不同节点的客户端组成。该部署主要应用的业务场景为集团公司根据业务需要搭建不同应用服务器，各应用服务器组成集群，各应用服务器各自维护。

图 3-73 多应用集群



3.2.4 产品 WAR 包单机部署

3.2.4.1 TOMCAT 上部署数据治理平台

TOMCAT 及 JDK 版本要求

因jar包中存在module-info.class需升级tomcat版本，tomcat版本需要用9.0.54、8.5.12以上的版本，JDK版本必须1.8。

Windows 操作系统

下面讲述的是在Windows环境下安装部署数据治理的整个过程。

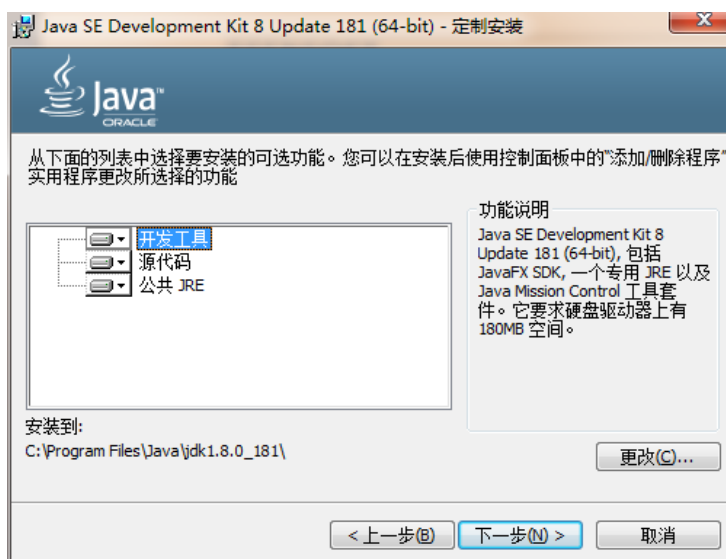
安装JDK

安装JAVA运行环境（必须jdk1.8或以上版本），如下图所示：

图 3-74 安装 JAVA 运行环境 1



图 3-75 安装 JAVA 运行环境 2



安装过程中，可以默认设置，直接NEXT完成安装，具体过程略。

- 安装TOMCAT

安装TOMCAT有两种方法。一种是使用TOMCAT的安装版程序，一种是使用TOMCAT的解压版文件。这两种方法的区别在于，通过安装版程序安装TOMCAT，不需要设置JAVA环境变量，使用解压版文件需要设置环境变量。

现在基本都使用TOMCAT的解压版程序，只需将TOMCAT压缩文件解压到某一目录即可。

- 配置环境变量

如果是解压版的TOMCAT，需要在startup.bat中设置JAVA环境变量。

Set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_221,路径改为实际JDK安装路径

- 参数优化

修改内存大小一般在启动文件startup.bat中设置，设置内存不能低于如下数值：

```
Set JAVA_OPTS=-Xmx2048m -Xms1025m -XX:MetaspaceSize=128m -  
XX:MaxMetaspaceSize=256m
```

- 部署数据治理
找到TOMCAT安装目录中的webapps文件夹，删除除ROOT文件夹外的其他文件和文件夹。
将数据治理服务器的WAR包文件复制到上述的webapps文件夹中。
- 启动服务器
运行TOMCAT安装目录中的bin\startup.bat，启动TOMCAT服务。
在IE地址栏中输入报表服务器地址，输入地址时，应带上TOMCAT的端口和应用的目录，即WAR包的目录名，完成服务器的初始化工作。

Linux 操作系统

在Linux中部署与windows中基本相同，下面以TOMCAT为例讲解在linux下部署的步骤。

Linux 下使用“变量名=变量值”设置变量，并使用 export 命令将其导出为环境变量。下面是安装并配置jdk、tomcat的环境变量。

安装JDK

```
#将jdk-8u131-linux-x64.tar.gz上传到/usr/local目录下  
#cd /usr/local (到usr/local目录下)  
#chmod a+x jdk-8u131-linux-x64.tar.gz (执行命令)  
#tar -xzf jdk-8u131-linux-x64.tar.gz (执行命令)
```

安装TOMCAT

同上，把tomcat (apache-tomcat-8.5.12.zip) 上传到你想要安装的位置，如： /usr/local下

```
# unzip apache-tomcat-8.5.12.zip (解包) 到此就完成安装了。
```

配置环境变量

在startup.sh或catalina.sh的开头位置设置JAVA_HOME

```
JAVA_HOME=/usr/local/jdk1.8.0_131
```

参数优化

修改内存大小一般在启动文件startup.sh中设置，设置内存不能低于如下数值：

```
export JAVA_OPTS=" -Xmx2048m -Xms1025m -XX:MetaspaceSize=128m  
-XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai  
-Dfile.encoding=GBK -Dsun.jnu.encoding=GBK"
```

部署数据治理

将edg(此名可以更改).war包复制到\Tomcat \webapps文件夹下

启动服务器

启动tomcat

```
#!/startup.sh
```

访问服务器，形如： http://127.0.0.1:8080/edg，完成服务器的初始化工作。后缀edg就是部署时war包的名称，如有更改，请对应更改。

TOMCAT 常用配置

修改端口

TOMCAT安装时默认的端口8080，TOMCAT安装版可以在安装中修改。对于安装完成后和解压版TOMCAT如何修改呢？

找到TOMCAT安装目录下的\conf\server.xml文件，用记事本打开，搜索安装时设置的端口值或默认的值8080，将其替换为要设置的端口值。

在修改端口时要注意，先要停止TOMCAT服务，再来修改。修改完成后，重启TOMCAT服务，即可通过修改后的端口访问WEB应用。

JVM设置

TOMCAT默认内存最大内存为128M，在大用户量情况下容易出现内存不够的情况，所以需要调整。

修改内存大小一般在启动文件startup.bat或startup.sh中设置，设置内存不能低于如下数值：

```
Set JAVA_OPTS=-Xmx2048m -Xms1025m -XX:MetaspaceSize=128m  
-XX:MaxMetaspaceSize=256m (windows)  
JAVA_OPTS=" -Xmx2048m -Xms1025m -XX:MetaspaceSize=128m  
-XX:MaxMetaspaceSize=256m" (linux)
```

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，在64bit环境下分配8G内存。

注意

某些JDK版本或操作系统限制最大堆内存，不支持分配超过1.5G的内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

线程调整

表 3-23 TOMCAT 常用配置

属性名	描述
maxThreads	Tomcat使用线程来处理接收的每个请求。这个值表示Tomcat可创建的最大的线程数。
acceptCount	指定当所有可以使用的处理请求的线程数都被使用时，可以放到处理队列中的请求数，超过这个数的请求将不予处理。
minSpareThreads	Tomcat初始化时创建的线程数。
maxSpareThreads	一旦创建的线程超过这个值，Tomcat就会关闭不再需要的socket线程。

上述参数在conf/ server.xml中进行设置，TOMCAT上述默认值如下：

maxThreads: 150

acceptCount: 100
minSpareThreads: 25
maxSpareThreads: 75

在一般情况下够用，如果同时并发用户数量很大，需要调整maxThreads参数，调整原则为每秒用户请求数*请求操作执行时间，比如每秒有100个请求，每个请求执行时间为1秒，maxThreads设置为100即可。同时要适当调整maxSpareThreads参数，不要关闭socket线程后又马上生成的操作。

3.2.4.2 Weblogic 上部署数据治理平台

3.2.4.2.1 在 Weblogic 中创建 Domain

执行安装目录下/Oracle/Middleware/Oracle_Home/wlserver/common/bin/config.sh

图 3-76 配置向导 1



按照实际情况修改路径

图 3-77 配置向导 2



设置账户密码，下一步

图 3-78 配置向导 3



选择开发或生产模式，下一步

图 3-79 配置向导 4



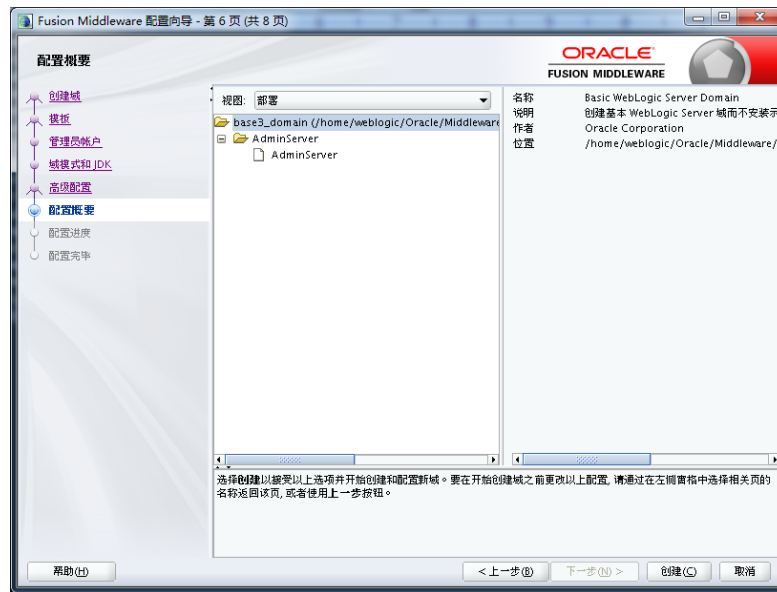
此处勾选可以设置端口和集群配置，默认下一步，端口使用7001

图 3-80 配置向导 5



单击创建开始创建新域

图 3-81 配置向导 6



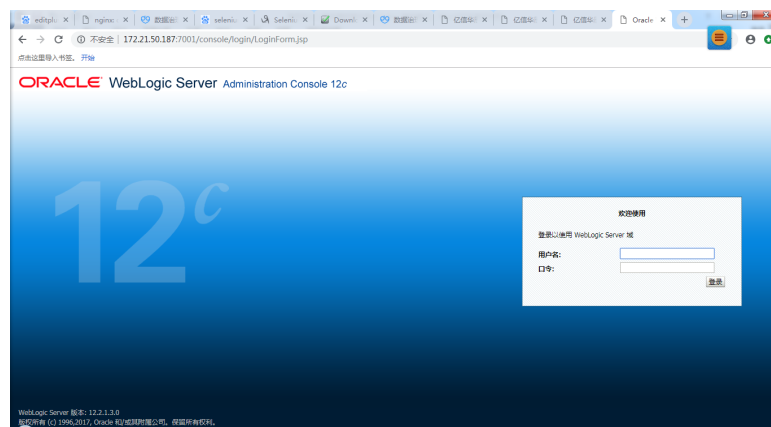
通过startWebLogic.sh(cmd)启动weblogic，直到窗口中出现listening on port 7001（端口号根据实际配置会不一样），表明服务已经成功启动。这时就可以在IE浏览器里键入http://localhost:7001/console，进入Weblogic控制台，输入您的管理员名和密码，即可进行控制台管理。

3.2.4.2.2 部署数据治理平台

Weblogic 12c中部署

Weblogic管理控制台访问地址：<http://服务器IP:7001/console>。在IE浏览器里键入该地址，进入Weblogic控制台登录页面，输入用户名和密码进行登录。

图 3-82 登录界面



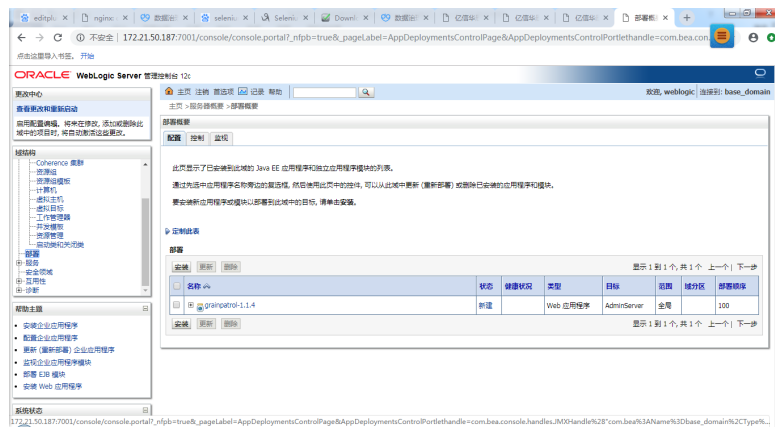
登录成功后，就出现Weblogic的控制台管理界面，如下：

图 3-83 控制台管理界面



单击“部署”节点，

图 3-84 部署



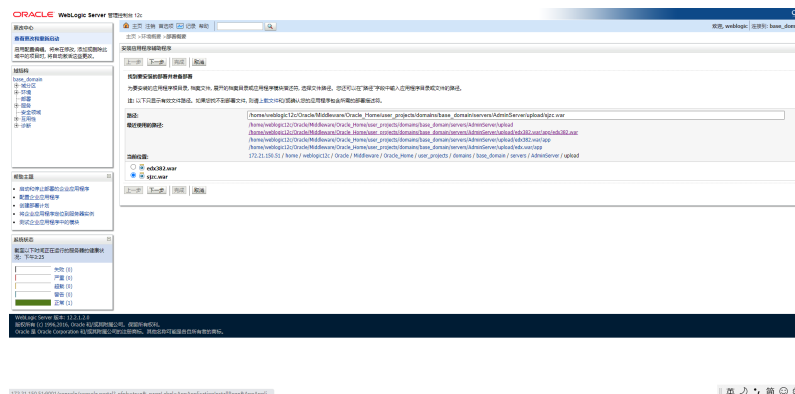
单击安装按钮

选择服务器的war包，选择下一步

说明

war包/WEB-INF/需要放一个web.xml文件“Attachment1.xml”

图 3-85 安装 1



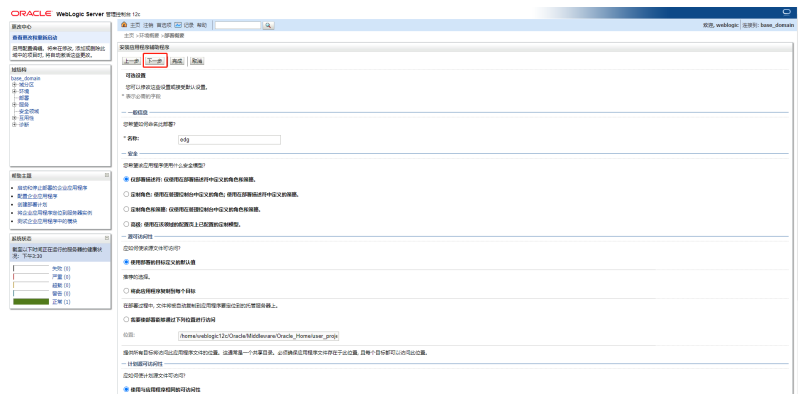
选择下一步

图 3-86 安装 2



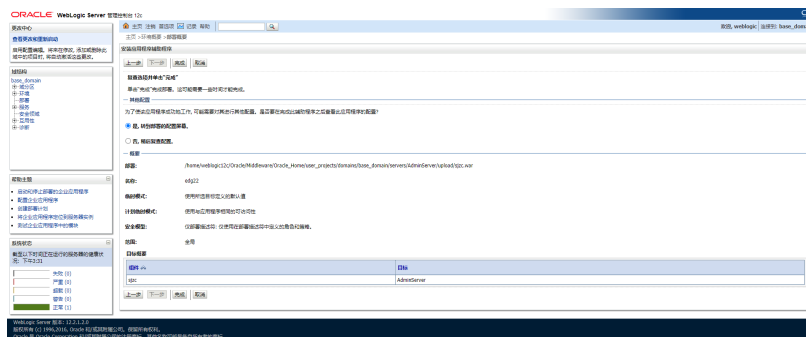
单击“下一步”按钮。

图 3-87 安装 3



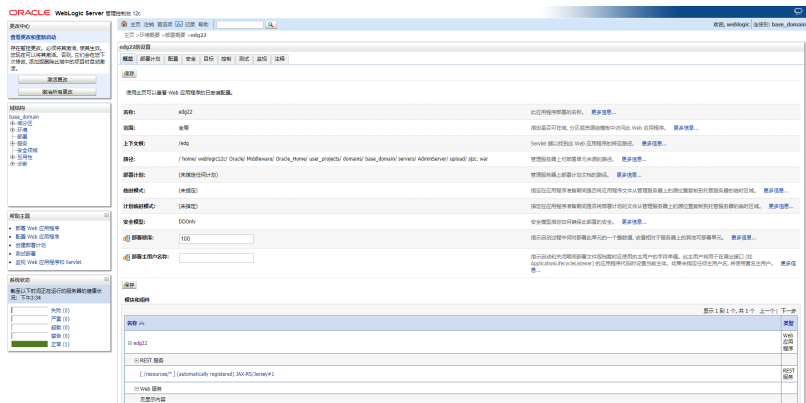
单击“完成”按钮。

图 3-88 安装 4



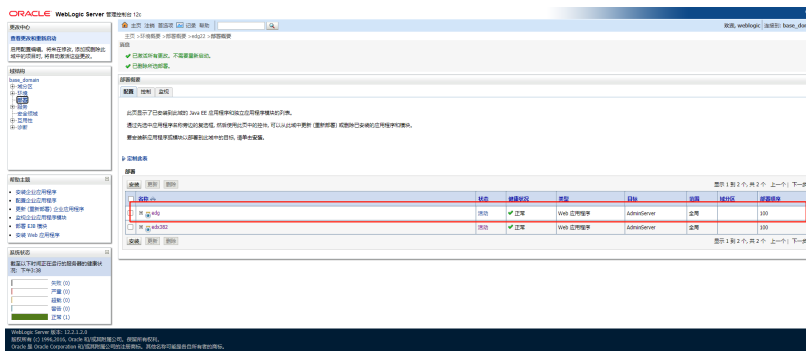
单击“保存”按钮

图 3-89 安装 5



部署完成

图 3-90 安装 6



参数设置

在startWebLogic.sh (cmd) 中, 增加

```
Set JAVA_OPTIONS=-Xmx2048m -Xms1025m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK ( Windows环境)  
JAVA_OPTIONS=" -Xmx2048m -Xms1025m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK ( Linux环境) "
```

其中, 设置内存不能低于上述值。

3.2.4.2.3 Weblogic 集群部署

说明

本章节介绍了Weblogic如何配置集群, 也介绍了在集群部署方式下, 数据治理平台应用配置集群。

集群 (Cluster) 是由两台或多台节点机 (服务器) 构成的一种松散耦合的计算节点集合, 为用户提供网络服务或应用程序(包括数据库、Web服务和文件服务等)的单一客户视图, 同时提供接近容错机的故障恢复能力。集群系统一般通过两台或多台节点服务器系统通过相应的硬件及软件互连, 每个群集节点都是运行其自己进程的独立服务器。这些进程可以彼此通信, 对网络客户机来说就像是形成了一个单一系统, 协同起来向用户提供应用程序、系统资源和数据。除了作为单一系统提供服务, 集群系统还

具有恢复服务器级故障的能力。集群系统还可通过在集群中继续增加服务器的方式，从内部增加服务器的处理能力，并通过系统级的冗余提供固有的可靠性和可用性。

数据治理平台支持集群部署，下面以Weblogic11g为例讲解如何部署集群，并在集群中部署服务。

创建新的Domain

选择“Toolà Configuration Wizard”，单击“下一步”按钮

图 3-91 创建新的 Domain1



选择安装域源

选择安装“Weblogic Server”，单击“下一步”按钮

图 3-92 选择域源



指定域名和位置

输入域的名称和位置

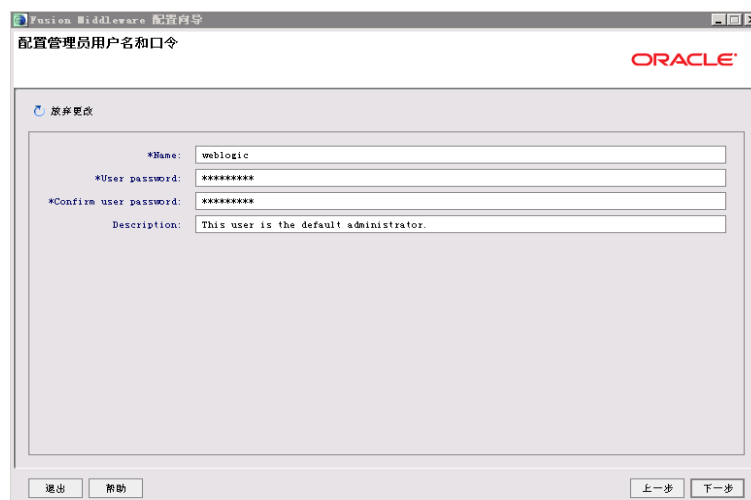
图 3-93 指定域名和位置



配置管理员用户名和密码

输入Weblogic管理员管理服务器（AdminServer）的用户名和密码，在此以“Weblogic”作用登录管理服务器的用户名和密码，单击“下一步”按钮

图 3-94 配置管理员用户名和密码



配置域启动模式和JDK

选择“生产模式”和“Sun SDK”后，单击“下一步”按钮

图 3-95 配置服务器启动模式和 JDK



选择管理服务器，受管服务器集群和计算机

选择“是”后，单击“下一步”按钮

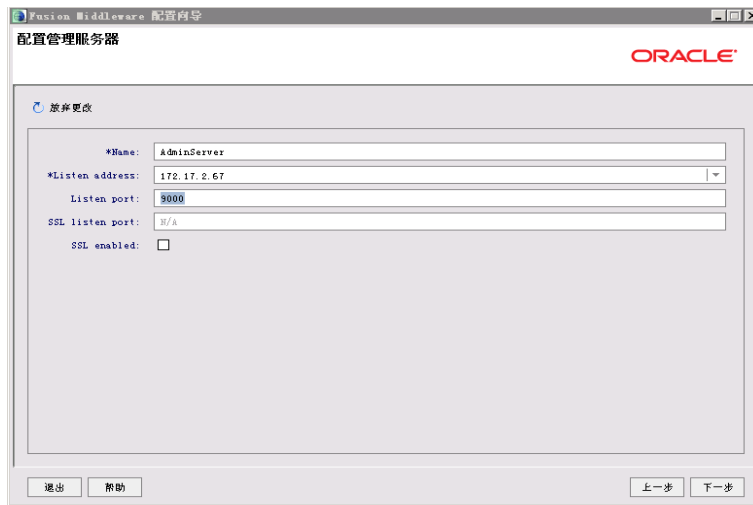
图 3-96 选择可选配置



配置管理服务器

输入Administrator Server的名称，监听地址，监听端口，如果需要SSL支持的话可以在“SSL enabled”后面的复选框上打勾，配置SSL监听端口，单击“下一步”按钮

图 3-97 配置管理服务器

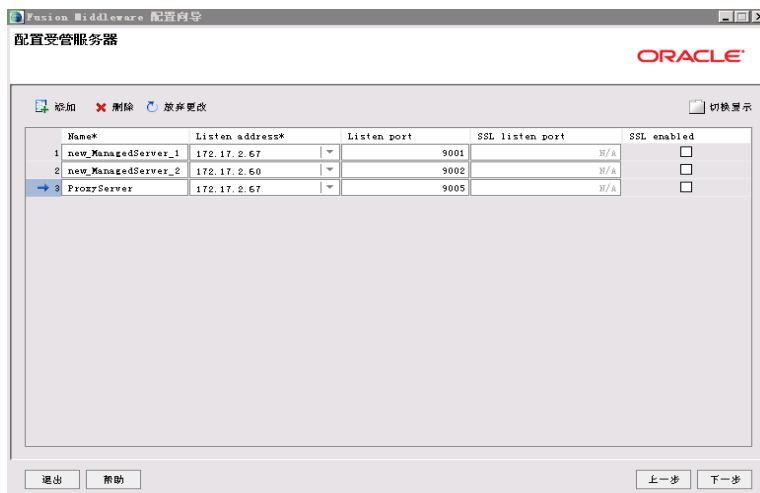


根据需要修改端口（默认为7001）

配置受管理服务器（代理服务器实际上也是一个受管理服务器）

输入Managed Server的名称，监听地址，监听端口，如果需要SSL支持的话，可以在“SSL enabled”复选框上打勾，配置SSL监听端口。可以配置多个Managed Server。单击“下一步”按钮

图 3-98 配置受管服务器



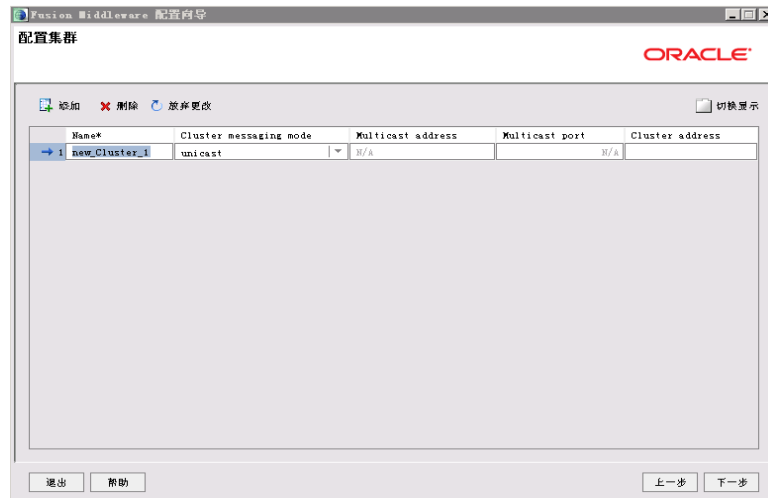
说明

我这台机器的IP是：172.17.2.67，另一台机上要安装受管理服务器的IP地址是：172.17.2.60

配置群集信息

单击“切换显示”，再单击“添加”，采用默认。单击“下一步”按钮。

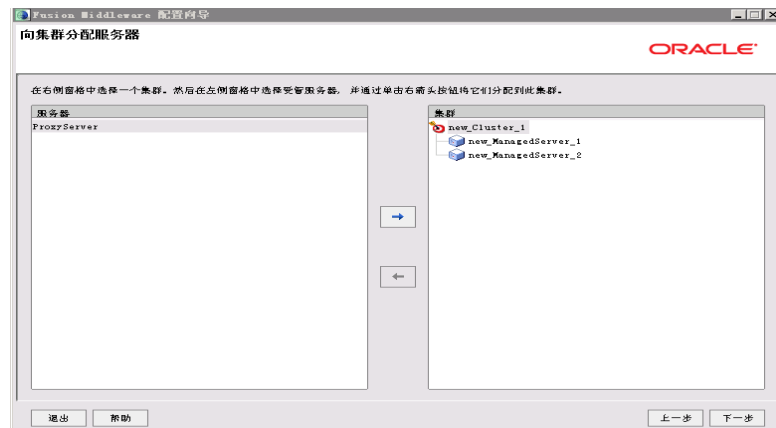
图 3-99 配置集群



将受管理服务器添加到集群中

选择左面列表中的Managed Server，将其添加到右面的列表中，这里代理服务器不要添加到右边，单击“下一步”按钮

图 3-100 向集群分配服务器



创建HTTP代理应用程序

选中“为群集创建Http代理”，myCluster代理服务器选择前面没有加入myCluster的ProxyServer，单击“下一步”按钮

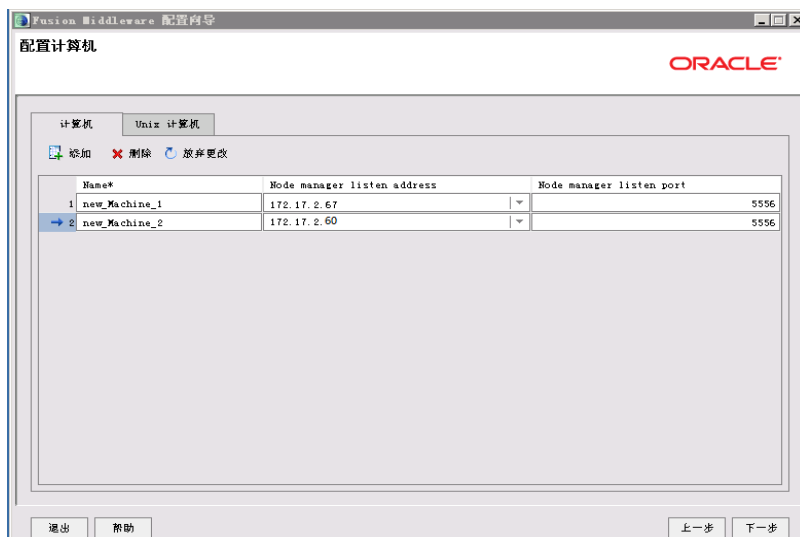
图 3-101 创建 HTTP 代理应用程序



配置计算机

添加两台计算机，按之前地址配置，端口采用默认，单击“下一步”按钮

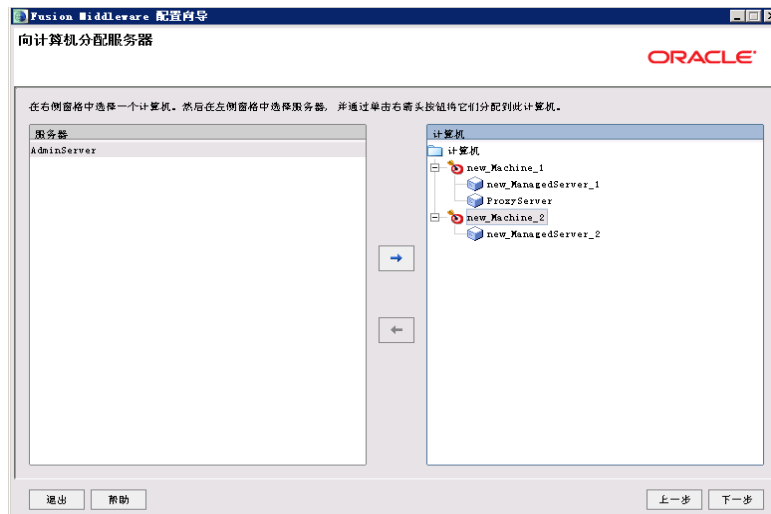
图 3-102 配置计算机



1. 向计算机分配服务器

按照之前规划向计算机分配服务器，如下图示

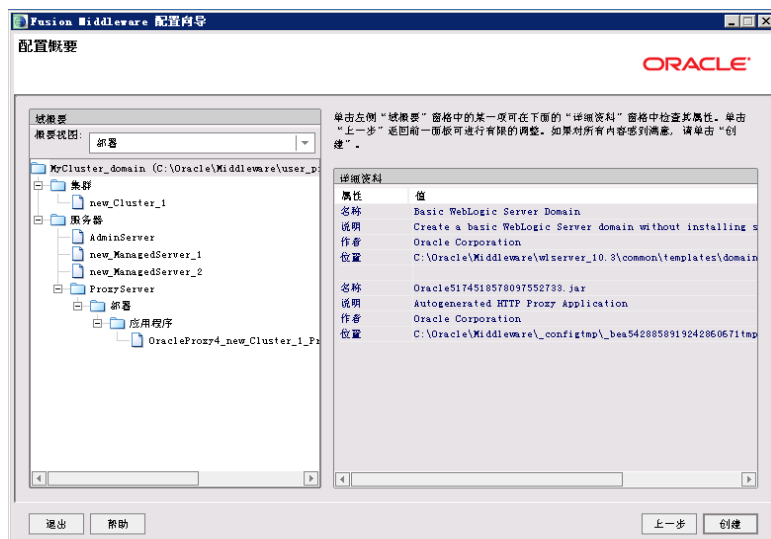
图 3-103 向计算机分配服务器



2. 检查Weblogic域

在此检查前面所做的配置，如果没有问题单击“下一步”按钮

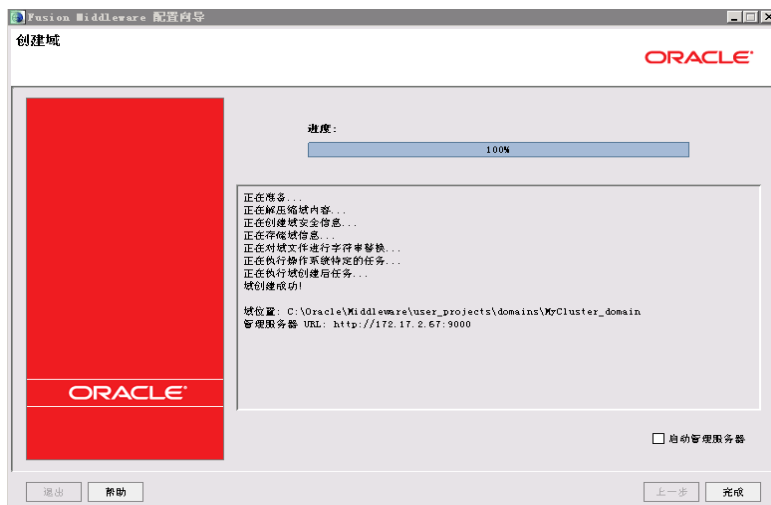
图 3-104 配置概要



3. 创建Weblogic域

单击“创建”按钮，创建无误后，单击“完成”按钮

图 3-105 创建域



4. 集群中辅节点操作

本例是weblogic双机集群，以上操作均在主节点中完成。

集群中的所有 Server 必须位于同一网段，并且必须是 IP 广播(UDP)可到达的；

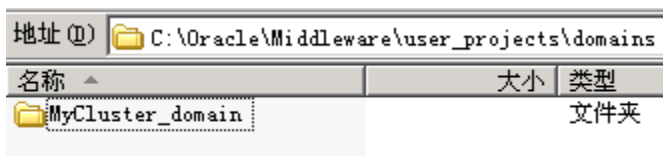
集群中的所有 Server 必须使用相同的版本，包括 Service Pack；

集群中的 Server 必须使用永久的静态 IP 地址。动态 IP 地址分配不能用于集群环境。如果服务器位于防火墙后面，而客户机位于防火墙外面，那么服务器必须有公共的静态 IP 地址，只有这样，客户端才能访问服务器；

辅节点weblogic安装和主节点相同版本，安装路径相同，过程略。辅节点具体操作如下：

- a. 将主节点创建好的domain文件夹复制到辅节点机器，如下图

图 3-106 复制



- b. 修改辅节点文件C:\Oracle\Middleware\wlserver_10.3\common\nodemanager\nodemanager.domains

添加如下内容并保存（实际环境中主要domain名称和路径）：

```
MyCluster_domain=C:\\Oracle\\Middleware\\user_projects\\domains\\MyCluster_domain
```

5. 启动集群服务

一般常用两种启动方式，一种是通过编辑脚本来启动，一种是启动nodemanager和主节点控制台后通过界面来启动，推荐使用后者。

- a. 编辑脚本来启动

编辑密码文件

在域目录servers/AdminServer/security、servers/ManagedServer_1/security、servers/ManagedServer_2/security、servers/proxyServer/security文件夹下（如果没有需新建）建立boot.properties文件，内容为：

```
username=Weblogic  
password=Weblogic1
```


编辑启动管理服务器的文件

在刚才创建的域目录下的bin目录中创建startAdminServer.cmd，内容为：

```
@ECHO OFF
@REM 启动管理服务器。
start .\startWeblogic
```

编辑启动代理服务器的文件

在创建的域目录下的bin目录中创建startProxyServer.cmd，内容为：

```
@ECHO OFF
@REM 启动代理服务器。
@REM 使用startManagedWeblogic命令启动Managed Server，它的命令格式是：
@REM startManagedWeblogic.cmd Managed Server的名称 Administrator Server地址
@REM startManagedWeblogic 受管理服务器名称 管理服务器URL
start .\startManagedWeblogic ProxyServer http://90.0.13.10:7001
```

编辑启动受管理服务器(ManagedServer_1)的文件

在刚才创建的域目录下的bin目录中创建startManagedServer_1.cmd，内容为：

```
@ECHO OFF
@REM 启动受管理服务器ManagedServer_1。
@REM 启动Managed Server
@REM 使用startManagedWeblogic命令启动Managed Server，它的命令格式是：
@REM startManagedWeblogic.cmd Managed Server的名称 Administrator Server地址
@REM startManagedWeblogic 受管理服务器名称 管理服务器URL
start .\startManagedWeblogic ManagedServer_1 http://90.0.13.10:7001
```

说明

使用startManagedWeblogic命令启动Managed Server，它的命令格式是：

startManagedWeblogic.cmd Managed Server的名称 Administrator Server地址

如果有多个受管理节点，可以做多个受管理服务器。

b. 启动nodemanager和主节点控制台通过weblogic控制台来启动

主节点操作如下：

i. 启动weblogic控制台，和前面介绍启动方式一样，此处略

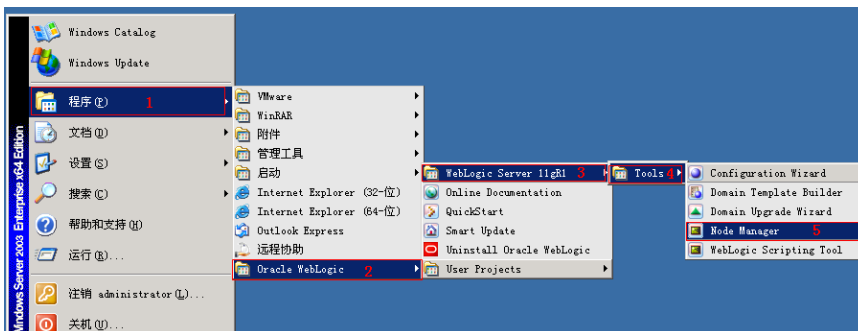
ii. 启动Node Manager

辅节点操作如下：（辅节点只需要启动Node Manager）

启动Node Manager

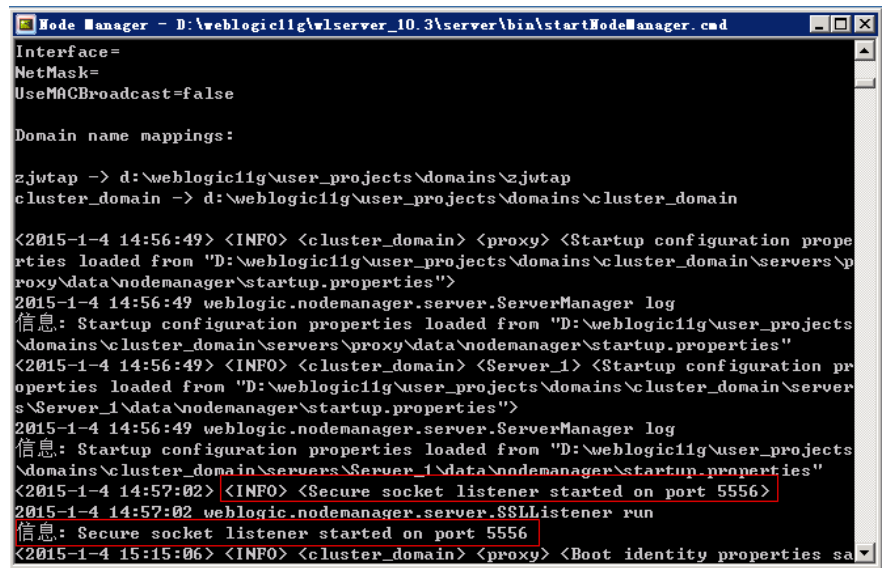
启动NodeManager方法如下：按如下图顺序启动

图 3-107 启动 Node Manager1



出现下图，5556端口处于监听表示启动成功。

图 3-108 启动 Node Manager2



此时，登录到weblogic控制台（<http://172.17.2.67:9000/console>）

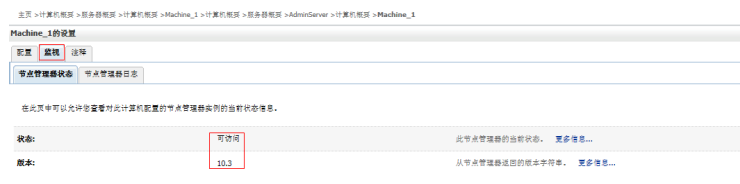
- a. 修改AdminServer的设置SSL 主机验证为无

图 3-109 AdminServer 的设置 1



修改完毕后重新启动weblogic控制台，查看计算机状态，确保为可访问

图 3-110 AdminServer 的设置 2



- b. 然后在服务器界面，依次启动代理和节点（即主控和受控节点）

切换到控制页面，一次勾选节点服务器和代理，单击“启动”，启动成功后如下图RUNNING状态，启动时间视机器配置而定，此时整个集群服务启动完毕。

图 3-111 启动代理和节点



6. 内存调整

修改域目录下\bin下的setDomainEnv.cmd (linux下commEnv.sh) 对应JDK的内存。

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，一般建议在32bit环境下最大内存不要超过2G，在64bit环境下不要超过系统可用内存的70%。

说明

某些JDK版本限制最大堆内存，不支持分配超过1.5G的内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

Weblogic建议最大最小内存分配一样大小。

7. Session设置

a. Application.properties

在war包中配置文件web-inf\classes\application.properties，添加如下内容：

```
server.session.timeout=1800
```

此例表示Session将在30分钟后过期

如果不配置，则默认30分钟

b. Weblogic.xml

在war包中找到web-inf\weblogic.xml，设置Weblogic特有部署描述符Weblogic.xml的<session-descriptor>元素的TimeoutSecs属性。这个值以秒为单位：

```
<session-descriptor>  
<session-param>  
  <param-name>TimeoutSecs</param-name>  
  <param-value>3600</param-value>  
</session-param>  
</session-descriptor>
```

默认值是3600秒

服务器端压力较大时建议减小此时间。

8. JNDI配置

登录weblogic控制台，进入“服务 > JDBC > 数据源”；单击【新建】按钮，填写JNDI名称，单击【下一步】；

默认下一步

默认下一步

填写数据库“实例名”、“数据库服务器ip”、“数据库端口”、“连接的数据库用户名”、“密码”，单击【下一步】按钮

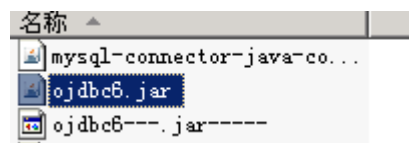
单击【测试配置】，确认连接成功，单击【下一步】按钮

选择需要使用该数据源的server节点，单击【下一步】按钮

配置好JNDI后需要添加jar包到weblogic的oracle\Middleware

\wlserver_10.3\server\lib的目录下，需添加：oracle_ojdbc14-1.0.0.jar，它是数据库驱动包,该包在数据治理平台 war包的\WEB-INF\lib路径下（jar包名视数据治理平台版本而定），添加后需要改名为ojdbc6.jar，并且注销weblogic自带的ojdbc6.jar文件。

图 3-112 名称



最后还需要到数据治理平台工作目录conf下修改jdbc.conf文件，在文件中增加第三方连接池参数datasource3=JDBC_QBI（其中JDBC_QBI为JNDI名称）。

配置完成启动weblogic，管理员admin登录查看数据库连接池基本信息。

9. JNDI数据库配置参数调优

登录weblogic控制台，单击进入“服务 > JDBC > 数据源”。单击需要调优的jdbc数据源名称。

进入数据源配置页面，选择“配置 > 连接池”选项卡。

在该页面可以修改“初始容量”（数据库空闲连接池）、“最大容量”（数据库最大连接数），单击下面的“高级”设置里面还有“最大等待连接数”。修改前需要单击“锁定并编辑”，本节不做详细讲解。

3.2.4.3 Tongweb6.1 上部署数据治理平台

部署环境配置

1. 安装前准备

安装JDK1.8或以上版本，具体参照第三章中1.1和2.1步骤

2. 开始安装

使用root用户登录，进入安装包文件所在的路径，先使用命令：chmod +x Install_TW6.1.5.17_Enterprise_Linux.bin，为安装包文件增加可执行权限。

使用命令执行文件：./Install_TW6.1.5.17_Enterprise_Linux.bin，安装过程简单，此处省略。

3. 启动和停止

安装成功后进入安装目录，如/data/TongWeb6.1/bin/，

通过nohup ./startserver.sh &来启动服务

服务启动后,通过命令./stopserver.sh来停止服务

4. 管理控制台

已经正确启动了Tongweb应用服务器后，在浏览器中输入以下URL：http://ip:9060/console

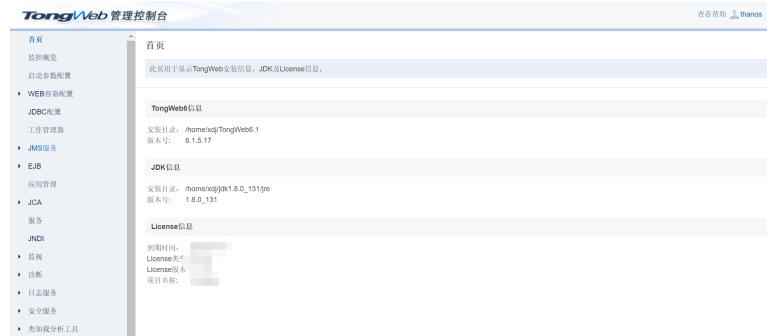
ip: 指安装Tongweb应用服务器软件的计算机的IP地址

9060: 是Tongweb应用服务器默认的http端口

部署数据治理

进入管理控制台，界面如下图：

图 3-113 进入管理控制台



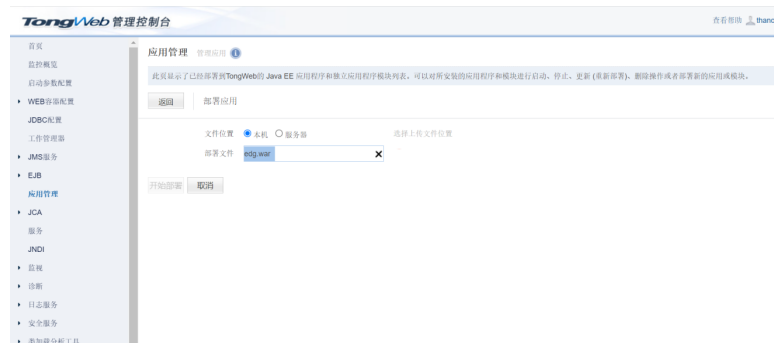
单击“应用管理”，进入应用部署界面，单击“部署应用”，进入下一界面

图 3-114 部署应用



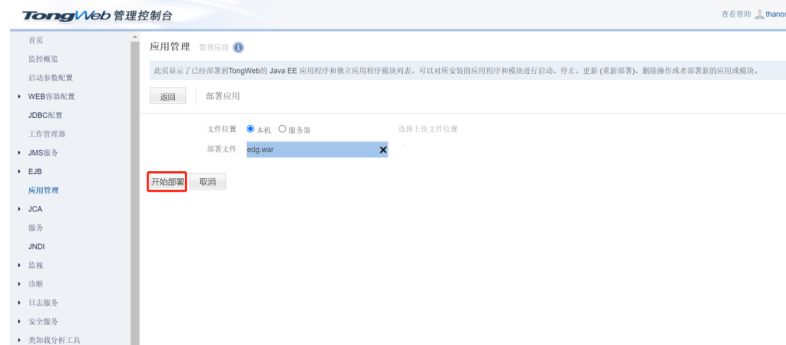
单击“选择文件”，选择所要部署的产品

图 3-115 选择文件



上传完毕后，单击“开始部署”

图 3-116 开始部署



单击下一步

图 3-117 下一步 1

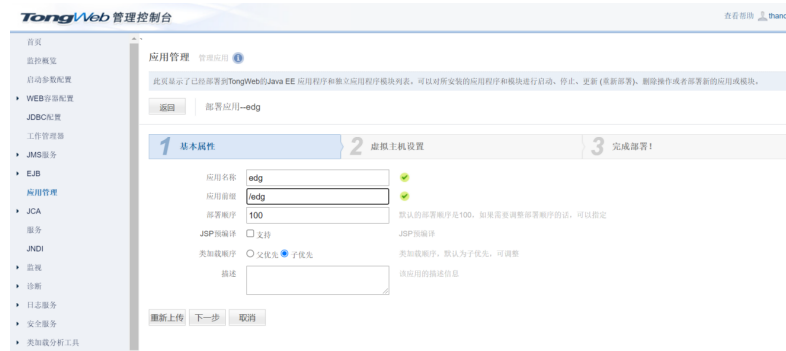
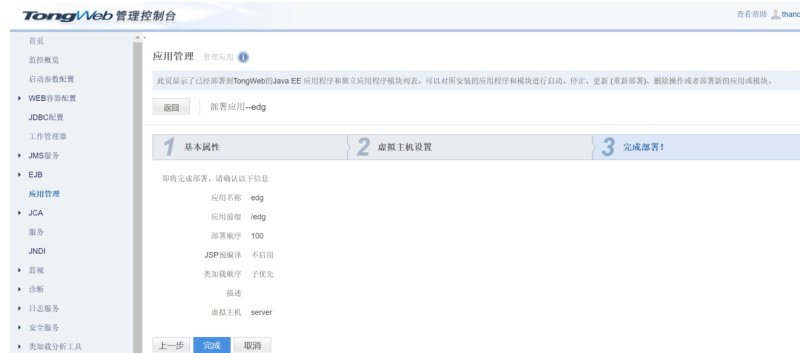


图 3-118 下一步 2



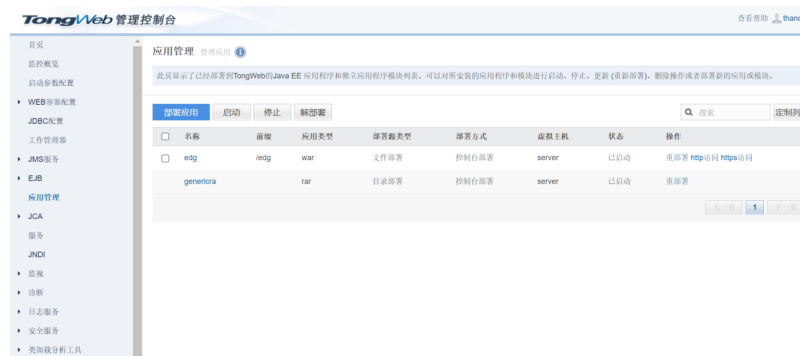
单击“完成”等待部署完毕

图 3-119 等待部署完毕



成功部署，状态为已启动，单击“http访问”即可进入产品访问地址

图 3-120 成功部署



3.2.4.4 金蝶 10 上部署数据治理平台

部署环境配置

本章节主要讲述在linux环境下如何安装及配置金蝶中间件。

安装应用服务器

确保所登录用户对AAS-V10.0.zip有可读写权限(在终端中用命令 `ls -l AAS-V10.0.zip` 查看), 可通过以下命令将AAS-V10.0.zip解压缩到安装目录:

可参考以下命令(具体以实际为准):

`unzip /data/AAS-V10.0.zip`其中: “/data/AAS-V10.0.zip” 表示金蝶10.0上传到服务器上的安装包

启动服务器

进入/data/AAS-V10/ApusicAS/aas/bin目录, 输入startserv命令并执行即可。

可参考以下命令(具体以实际为准):

```
cd /data/AAS-V10/ApusicAS/aas/bin
```

前台启动

```
sh ./ startserv
```

后台启动

```
nohup ./startserv &
```

后台启动可以tail -f nohup.out 查看输出的日志

管理控制台

已经正确的启动了Apusic应用服务器后，在浏览器中输入以下URL：https://ip:6848/

ip：指安装Apusic应用服务器软件的计算机的IP地址

6848：是Apusic应用服务器默认的http端口

访问成功后出现如下图所示的页面：

图 3-121 成功后界面



Apusic 管理控制台的初始用户名为 admin，密码为上面配置管理端密码。

停止服务器

```
cd /data/AAS-V10/ApusicAS/aas/bin, 进入安装目录下
```

```
执行./stopserv关闭Apusic, 如果无法关闭, 直接kill -9 PID
```

部署数据治理

说明

war包需要删除\WEB-INF\lib\eweb-1.3.1-SNAPSHOT.jar\META-INF\services\javax.servlet.ServletContainerInitializer文件，否则会提示“Cannot expose bind macro helper 'springMacroRequestContext' because of an existing model object of the same name”

进入控制台管理界面，单击右边“应用程序”，单击“部署”

图 3-122 部署 1

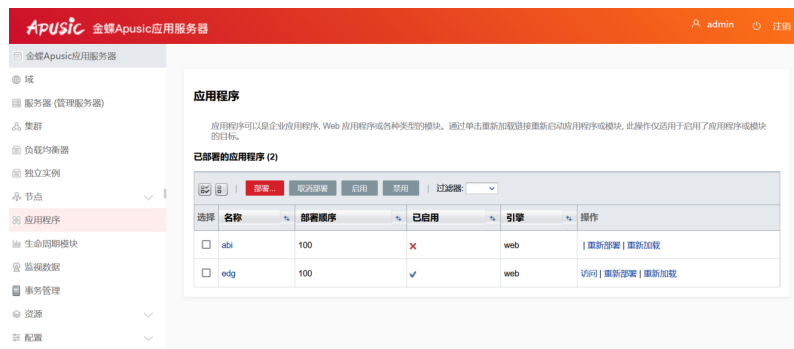


图 3-123 部署 2



图 3-124 部署 3



3.2.4.5 TAS2.8.5 上部署数据治理平台

部署环境配置

- 安装前准备
安装JDK1.8或以上版本，具体参照前面章节步骤
- 开始安装
使用root用户登录，进入安装包文件所在的路径，执行命令：`./java -jar tas-installer-2.8.5.jar -console`，安装过程简单，此处省略。
- 启动和停止
安装成功后进入安装目录，如/data/tas2.8.5/bin，通过./ StartTAS.sh来启动服务
服务启动后,通过命令./ StopTAS.sh来停止服务。
- 管理控制台
已经正确启动了Tas应用服务器后，在浏览器中输入以下URL：`http://ip:8081/tas-console`
ip：指安装Tas应用服务器软件的计算机的IP地址
8081：是Tas应用服务器默认的http端口

部署数据治理

进入管理控制台，单击左侧的“发布应用程序”

图 3-125 部署数据治理 1



第一种方式：单击“上传文件”，选择需要上传的文件，说明：文件小于200M时，选择上传war文件

图 3-126 部署数据治理 2



第二种方式：文件大于200M时，选择服务器上的war文件

图 3-127 部署数据治理 3



这里使用第二种方式

图 3-128 部署数据治理 4



单击确定，开始部署

图 3-129 部署数据治理 5



成功部署之后，应用程序列表中会出现刚刚部署的应用程序，单击访问即可进入应用访问地址

图 3-130 部署数据治理 6



3.2.4.6 Docker 上部署数据治理平台

制作 Dockerfile

Dockerfile是一个包含用于组合映像的命令的文本文件。可以使用在命令行中调用任何命令。Docker通过读取Dockerfile中的指令自动生成映像。可以参照以下方式制作Dockerfile。

```
FROM centos:7
ENV JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF8 -Xmx2048m -Xms1025m -XX:MaxMetaspaceSize=512m -XX:MetaspaceSize=256m -Duser.timezone=Asia/Shanghai -Desen.edg.workdir=/usr/local/workdir -Djava.awt.headless=true"
ENV JAVA_HOME /usr/local/java
ENV JRE_HOME /usr/local/java/jre
ENV PATH /usr/local/java/bin:$PATH
WORKDIR /usr/local/
ADD jdk-8u131-linux-x64.tar.gz /usr/local
ADD apache-tomcat-8.5.12.tar.gz /usr/local
RUN mv jdk1.8.0_131 java && mv apache-tomcat-8.5.12 tomcat
COPY sjzl.war /usr/local/tomcat/webapps
EXPOSE 6060
ENTRYPOINT ["/usr/local/tomcat/bin/catalina.sh","run"]
```

说明

jdk-8u131-linux-x64.tar.gz、apache-tomcat-8.5.12.tar.gz、sjzl.war、Dockerfile这几个均放在同一路径下。

构建镜像

1. 使用当前目录的 Dockerfile 创建镜像

```
docker build -t sjzl:v1 .
```

-t: 镜像的名字及标签

2. 通过 -f Dockerfile 文件的位置

```
docker build -f /opt/Dockerfile -t sjzl:v1 .
```

上下文路径，是指 docker 在构建镜像，有时候想要使用到本机的文件（比如复制），docker build 命令得知这个路径后，会将路径下的所有内容打包。

```
Successfully built 5d113977d755  
Successfully tagged sjzl:v1
```

构建过程最后出现上面显示，即代表镜像构建成功。

启动容器

```
docker run -d -p 6060:8080 -v /opt/sjzlworkdir:/usr/local/workdir sjzl:v1
```

-d: 服务形式启动

-p: 端口映射，宿主机6060端口: docker容器8080端口，6060可随意修改成未被占用的端口

-v: 工作目录路径,宿主机/opt/sjzlworkdir: 容器/usr/local/workdir

访问服务器

访问地址: <http://ip:6060/sjzl>

ip为宿主机ip地址

6060为宿主机映射的端口

sjzl为war包名称

根据实际修改

工作目录及注册说明

制作Dockerfile时，环境变量中的-Desen.edg.workdir=/usr/local/workdir是用来设置容器内部的产品工作目录，启动时再把工作目录映射到宿主机上。

注册时通过注册界面导入注册申请文件，不要直接把注册申请文件放在宿主机工作目录下。

相关问题

报错无法获取服务器硬件信息。

此问题是由于产品在注册时需要获取硬件信息，但是容器内部没有相关命令，须在构建镜像时加上一行

```
RUN yum install -y net-tools dmidecode
```

3.2.5 产品 WAR 包集群部署

3.2.5.1 集群场景说明、工作目录指定及 JVM 参数设置

集群场景说明

集群4个节点部署在两台服务器上，对应的服务器IP、端口、工作目录如下。Redis单独一台服务器部署。以双机双节点的配置数据治理平台集群为例，说明在数据治理平台中如何配置集群。

表 3-24 集群场景说明 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine1	Server1 (主控)	192.168.1.101	9001	/work_dir/rootdir1/
Machine1	Server2	192.168.1.101	9002	/work_dir/rootdir2/
Machine2	Server3	192.168.1.102	9003	/work_dir/rootdir3/
Machine2	Server4	192.168.1.102	9004	/work_dir/rootdir4/

表 3-25 集群场景说明 2

物理机器	安装软件	IP地址	端口号
Machine3	Redis	192.168.1.103	6379

工作目录指定

涉及到集群的工作目录指定，每个节点必须指定一个单独的工作目录，而且多节点不能同时使用同一个工作目录。

指定工作目录需要在中间件启动文件设置，在Server1的启动文件中设置：

```
-Desen.edg.workdir =/work_dir/rootdir1
```

在Server2的启动文件中设置：

```
-Desen.edg.workdir =/work_dir/rootdir2
```

在Server3的启动文件中设置：

```
-Desen.edg.workdir =/work_dir/rootdir3
```

在Server4的启动文件中设置：

```
-Desen.edg.workdir =/work_dir/rootdir4
```

JVM 参数设置

表 3-26 JVM 参数设置

节点名称	参数设置	备注
Server1	-Desen.edg.workdir=/work_dir/ rootdir1 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	-Xmx和-Xms根据服务器内存 大小设置，64位OS建议分配 8G-16G
Server2	-Desen.edg.workdir=/work_dir/ rootdir2 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上
Server3	-Desen.edg.workdir=/work_dir/ rootdir3 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

节点名称	参数设置	备注
Server4	-Desen.edg.workdir=/work_dir/ rootdir4 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

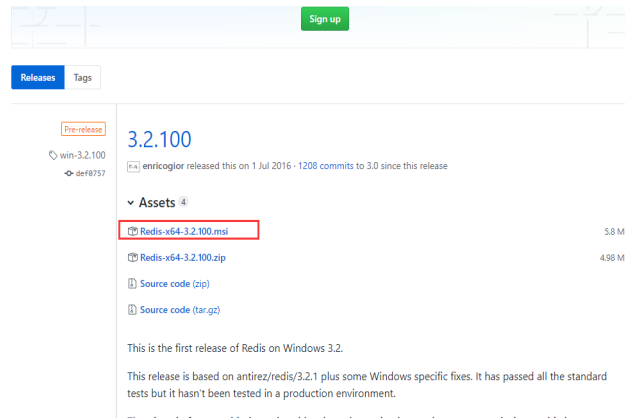
3.2.5.2 Redis 安装

Redis支持32位和64位的Windows和Linux操作系统。

Windows 下安装 redis

下载地址：<https://github.com/MSOpenTech/redis/releases>

图 3-131 下载



根据需要选择合适版本的Redis，下载zip包，下载完成后解压缩，会得到一个redis-server.exe。

下载完成后，双击安装，安装完成后，启动服务即可

使用客户端测试连接：redis-desktop-manager-0.9.3.817.exe

其实Redis在Windows上不需要安装，在命令行下执行这个redis-server.exe，就会启动redis服务，启动后不要关闭这个命令行窗口，关闭它服务就停止了。

Linux 下安装 redis

redis在linux下的安装使用包管理器来安装即可，对于没有外网的环境也可以复制压缩包，解压后手动安装。

安装

```
# wget http://download.redis.io/releases/redis-5.0.5.tar.gz  
# tar -zxf redis-5.0.5.tar.gz  
# cd redis-5.0.5  
# make  
# make install
```

启动

编译完成以后，启动 redis-server

```
# cd src  
# ./redis-server ../redis.conf &
```

访问redis

用自带的redis-cli客户端工具就可以访问Redis服务器了。

Redis的默认服务端口是6379

```
# redis-cli  
127.0.0.1:6379>
```

Redis配置文件说明

Redis的配置文件Redis.conf在Redis的安装目录下，里面有很多关于Redis的设置。用户可以通过直接修改文件更改配置，也可以通过命令像设置key-value数据库一样修改配置

```
config get xxx //获取某一个配置的值  
config set xxx xxx //设置某一项配置  
config get * 获取所有配置
```

建议直接修改配置文件，因为通过命令修改，重启Redis后就会失效。

主要配置参数列表：

表 3-27 配置参数

参数	值	描述
port	6379	redis的端口
bind	127.0.0.1	绑定的ip地址，默认是127.0.0.1。 如果本机有多个网卡，且限制只能通过其中一个网卡对应的IP才能访问redis时，可绑定该IP，例：bind IP
protected-mode	yes	保护模式，默认为yes，如果要其他网络能访问，改为no即可
requirepass	-	登录密码，默认是空
maxclients	0	同一时间能够连接到redis的客户端数量，默认是无限制，0表示不限制
maxmemory	-	redis能够使用的最大内存 单位byte, 默认无限制

3.2.5.3 部署包产品注册与集群文件配置

部署包产品注册

当在所有节点部署EDG包后，需要分别访问下列地址完成产品的注册。

<http://192.168.1.101:9001/edg>

<http://192.168.1.101:9002/edg>

<http://192.168.1.102:9003/edg>

<http://192.168.1.102:9004/edg>

在所有4个节点均完成注册后，分别登录所有节点，确认所有节点可以单独访问。

集群文件配置

停掉所有服务器，分别在各个工作目录的conf文件夹中放置cluster.properties，其中Server1为主控节点

表 3-28 集群文件配置

节点名称	配置内容
Server 1	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=true cluster.type=redis cluster.group=group1 cluster.nodeid=node1 cluster.name=测试集群节点1 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 cluster.redis.password=125 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server 2	<pre> #charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node2 cluster.name=测试集群节点2 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 cluster.redis.password=125 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false </pre>

节点名称	配置内容
Server 3	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node3 cluster.name=测试集群节点3 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码,则配置 cluster.redis.password=125 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server 4	<pre> #charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node4 cluster.name=测试集群节点4 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码,则配置 cluster.redis.password=125 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false </pre>

配置文件中参数说明:

表 3-29 参数说明

参数名	描述	例子
cluster.enabled	是否启用集群	默认不启用
cluster.nodeid	集群节点id	node1

参数名	描述	例子
cluster.name	集群节点名称	测试集群节点1
cluster.type	集群类型（redis： redis集群，netty： netty集群）	redis，默认使用redis
cluster.message.retry	消息发送失败后重发 次数	3
cluster.health.interval	心跳时间间隔（单 位：毫秒）	1500，默认1500
cluster.health.retry	集群节点离线时重试 次数，即连续几次没 有收到心跳则判定节 点离线	3
cluster.heath.rejoin	集群节点离线后重新 加入重试次数，即集 群节点连续几次都受 到心跳消息则判定节 点加入	3
cluster.redis.ip	redis ip	172.21.1.65
cluster.redis.port	redis端口	6379
cluster.redis.password	redis密码，没有密 码可不传	125
cluster.redis.nodes	redis本身的集群内 的其他节点（除上述 配置的其他redis集 群节点），没有可不 配置	172.21.1.66:6379;172.21.1.67:637 9

3.2.5.4 启动服务器、集群检查及注意事项

启动服务器

集群配置文件完成后，先启动server1，待server1启动完成后，可同时启动其它三个服务器。

集群检查

以admin身份登录数据治理平台，依次进入“系统设置 > 性能与维护 > 集群信息”，检查如下信息：

- 检查所有配置节点是否存在
- 检查所有节点在线状态

- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

注意事项

1. **集群内各服务器版本需保持一致：**只能同一版本的服务器做集群
2. **各服务器的数据库配置需一致：**各服务器数据库配置需要保持一致，其中url中ip地址也要一致
3. **各服务器redis配置需一致：**各服务器配置的redis相关参数需要是一致的，如redis ip、端口、密码

3.2.5.5 集群扩容说明

新增N个节点，与原集群各节点配置保持一致，连接同一个Redis，即可达到集群扩容目的。

例如新增物理机器Machine4，节点名称Server5,IP地址192.168.1.103，端口号9005

表 3-30 集群扩容说明 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine4	Server5	192.168.1.103	9005	/work_dir/rootdir5/

指定工作目录、设置JVM参数后，正常注册数据治理平台，然后停掉新增节点服务器，在工作目录的conf文件夹中放置cluster.properties，配置内容为：

表 3-31 集群扩容说明 2

节点名称	配置内容
Server 5	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node5 cluster.name=测试集群节点5 #需要二选一配置的属性: #集群连接配置方式1 cluster.redis.ip=172.21.1.65 cluster.redis.port=6379 #集群连接配置方式1 #cluster.redis.nodes=172.21.1.65:6379,172.21.1.65:6380,172.21.1.65:6381 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=125 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

📖 说明

该配置文件中cluster.redis.ip与原集群各节点保持一致

然后启动新增节点服务器, 启动成功后以admin身份登录数据治理平台, 依次进入“系统设置 > 性能与维护 > 集群信息”, 检查如下信息:

- 检查所有配置节点是否存在

- 检查所有节点在线状态
- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

3.2.6 实时处理服务部署

3.2.6.1 Linux 环境单机版部署

📖 说明

如果部署过集群版，需先停止所有服务并卸载所有组件才能继续完成安装。

这里以安装包在opt目录为例。

```
# cd /opt/edi-RTPS-el7/bin
# sh stop_all.sh
# cd /opt
# rm -rf edi-RTPS-el7
```

卸载完毕后：

下载edi-RTPS-el7.zip文件，解压后进入edi-RTPS-el7目录下的bin目录执行install_all.sh脚本。

假设把安装包放在opt目录下，命令如下：

```
# unzip -q /opt/edi-RTPS-el7.zip -C /opt (解压安装包)
# cd /opt/edi-RTPS-el7/bin
# sh install_all.sh
```

表 3-32 附录

文件路径	功能描述
edi-RTPS-el7/bin/start_all.sh	启动所有服务（默认安装后直接启动）
edi-RTPS-el7/bin/stop_all.sh	停止所有服务
edi-RTPS-el7/bin/status_all.sh	检查所有服务状态
edi-RTPS-el7/installldir	各服务安装目录

⚠️ 注意

数据工厂所在服务器需添加实时处理服务器IP和主机名映射，例：

172.17.3.252 test252

实时处理服务器也应添加数据工厂所在服务器主机名映射，例：

172.17.3.237 edi

3.2.6.2 Linux 环境集群版部署

环境准备

下面以三台操作系统为centos7.4的服务器为例。

表 3-33 环境准备

主机ip	主机名
172.17.3.252	test252
172.17.3.253	test253
172.17.3.254	test254

环境部署

说明

如果部署过单机版，需先停止所有服务并卸载所有组件才能继续完成安装。

这里以安装包在opt目录为例。

```
# cd /opt/edi-RTPS-el7/bin  
# sh stop_all.sh  
# cd /opt  
# rm -rf edi-RTPS-el7
```

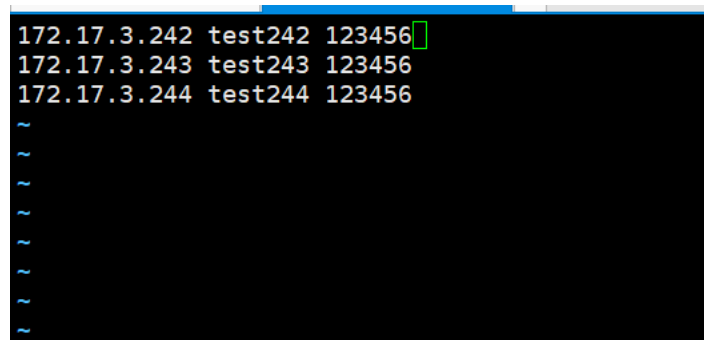
卸载完毕后：

下载edi-RTPS-el7.zip文件，解压后进入edi-RTPS-el7目录下的conf目录，配置slaves文件，然后执行install_all.sh文件进行部署。

假设把安装包放在opt目录下，命令如下：

```
# unzip -q /opt/edi-RTPS-el7.zip -C /opt (解压安装包)  
# cd /opt/edi-RTPS-el7/bin  
# vi /opt/edi-RTPS-el7/conf/slaves(然后按【ip 主机名 密码】格式编写，以空格分隔)
```

图 3-132 图示



```
# sh install_all.sh
```

表 3-34 附录

文件路径	功能描述
edi-RTPS-el7/bin/start_all.sh	启动所有服务（默认安装后直接启动）
edi-RTPS-el7/bin/stop_all.sh	停止所有服务
edi-RTPS-el7/bin/status_all.sh	检查所有服务状态
edi-RTPS-el7/installldir	各服务安装目录

⚠ 注意

数据工厂所在服务器需添加实时处理服务器IP和主机名映射，例：

172.17.3.252 test252

172.17.3.253 test253

172.17.3.254 test254

实时处理服务器（所有节点：test252 test253 test254）也应添加数据工厂所在服务器主机名映射。例：

172.17.3.237 edi

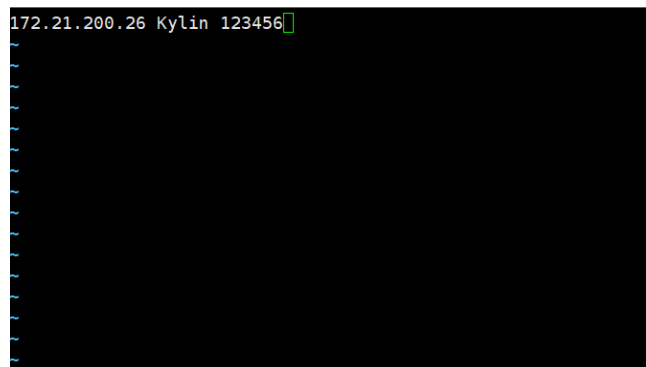
3.2.6.3 “银河麒麟”系统单机版部署

下载edi-RTPS-kylin.zip文件，解压后进入edi-RTPS-kylin目录下的conf目录，配置slaves文件，然后执行install_all.sh文件进行部署。

假设把安装包放在opt目录下，命令如下：

```
# unzip -q /opt/edi-RTPS-kylin.zip -C /opt (解压安装包)
# cd /opt/edi-RTPS-kylin/bin
# vi /opt/edi-RTPS-kylin/conf/slaves (然后按【ip 主机名 密码】格式编写，以空格分隔)
```

图 3-133 图示



```
# ./install_all.sh
```

表 3-35 附录

文件路径	功能描述
edi-RTPS-el7/bin/start_all.sh	启动所有服务（默认安装后直接启动）
edi-RTPS-el7/bin/stop_all.sh	停止所有服务
edi-RTPS-el7/bin/status_all.sh	检查所有服务状态
edi-RTPS-el7/installdir	各服务安装目录

注意

数据工厂所在服务器需添加实时处理服务器IP和主机名映射，例：
172.21.200.26 Kylin
实时处理服务器也应添加数据工厂所在服务器主机名映射。例：
172.17.3.237 edi

3.2.6.4 Hdfs、大数据相关驱动安装与 Standalone 模式 local 配置

配置驱动

hadoop依赖.zip。

影响数据源中hdfs数据源的连接。

压缩包内的jar包放入工作目录：lib\edi\hadoop目录下。

此目录会内自带edi-hadoop-deps-4.4.1-SNAPSHOT.jar，如果不存在，从数据工厂4.4.1war包中复制一份放入。

edataexchange-hadoop依赖的jar包.zip。

影响Hadoop文件输入组件，Hadoop迁移组件，Hadoop输出组件这些大数据组件。

压缩包内的jar包放入工作目录：lib\edi\cdh_hadoop目录下。

此目录会内自带edi-hadoop-deps-4.4.1-SNAPSHOT.jar，如果不存在，从数据工厂4.4.1war包中复制一份放入。

Standalone 模式 local 配置

图 3-134 Standalone 模式 local 配置信息

■ HDFS配置

*HDFS链接地址:

■ 计算引擎配置

*计算引擎:

*运行模式:

*Master地址:

*应用程序名:

参数配置:

3.2.6.5 Standalone 模式 spark 配置

数据工厂应用内配置

图 3-135 计算引擎配置

■ 计算引擎配置

*计算引擎:

*运行模式:

*Master地址:

*应用程序名:

参数配置:

参数配置中spark.executor.extraClassPath指定spark上需要额外加载的jar包的路径，jar包需要放在spark所在服务器上。如果spark是集群环境，则每个节点都需要放入jar包，且路径相同。

Spark 所需 jar 包

所需jar包在数据工厂4.4.1war包中lib目录下，复制即可。

表 3-36 jar 包

jar包	必须	说明
ecore-1.4.1-SNAPSHOT.jar	是	-
edi-core-4.4.1-SNAPSHOT.jar	是	-
edi-designer-api-4.4.1-SNAPSHOT.jar	是	-
edi-engine-spark-4.4.1-SNAPSHOT.jar	是	-
edi-widget-base-4.4.1-SNAPSHOT.jar	是	-
edi-widget-hadoop-4.4.1-SNAPSHOT.jar	是	-
eutil-1.3.1-SNAPSHOT.jar	是	-
edi-edatasource-api-4.4.1-SNAPSHOT.jar	是	-
ejdbc-1.3.1-SNAPSHOT.jar	是	-
esql-1.2.3-SNAPSHOT.jar	否	如果输出组件用到实时文件输出组件，则要加上此依赖。
guava-20.0.jar	是	-
kafka-clients-2.0.0.jar	否	如果输出组件用到kafka生产组件，则要加上此依赖。
rxjava-1.1.6.jar	是	-
spark-sql-kafka-0-10_2.11-2.4.0.jar	否	如果输出组件用到kafka生产组件，则要加上此依赖。
数据库驱动	否	根据实际需要确定，如果你的实时任务中使用了数据库，那么就需要添加对应数据库的驱动程序包如使用了oracle和mysql,那么就需要添加oracle和mysql的jdbc驱动。如果没使用数据库则不需要

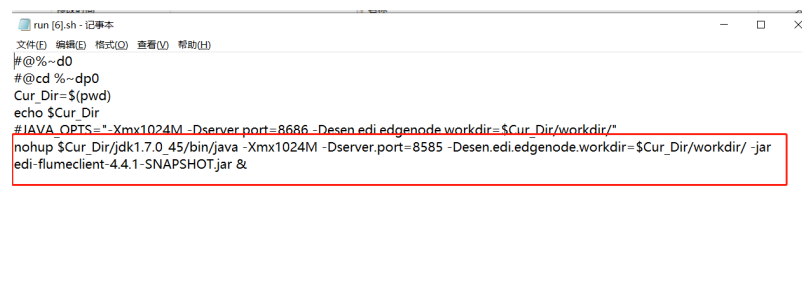
3.2.7 边缘采集节点部署

Linux 环境部署

下载edi-edgenodeagent-linux.zip文件，放在Linux服务器某个文件夹中，解压压缩，注意解压文件后可能没有权限需要重新执行chmod命令赋予权限，文件中包含jdk，jar包，flume、canal、run.sh。

修改run.sh文件中的端口(默认8686)、内存大小，然后执行run.sh启动，注意端口不要被占用，如果需要程序在后台运行，则在run.sh最后一行前后加nohup和&，如图。

图 3-136 Linux 环境部署



```
run [6].sh - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#@%~d0
#@cd %~-dp0
Cur_Dir=$(pwd)
echo $Cur_Dir
#JAVA_OPTS="-Xmx1024M -Dserver.port=8686 -Desen.edi.edgenode.workdir=$Cur_Dir/workdir/"
nohup $Cur_Dir/jdk1.7.0_45/bin/java -Xmx1024M -Dserver.port=8585 -Desen.edi.edgenode.workdir=$Cur_Dir/workdir/ -jar
edi-flumeclient-4.4.1-SNAPSHOT.jar &
```

注意

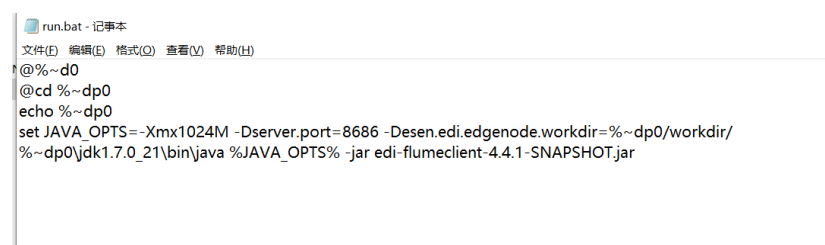
自研环境的操作系统中，根据不同的CPU使用不同架构的jdk，例如：银河麒麟环境，需要使用openjdk，而边缘采集节点程序自带的jdk为Oracle jdk。操作方式：使用对应的jdk替换程序原有的jdk，注意文件名需与原有的文件名一致。

Windows 环境部署

下载edi-edgenodeagent-win.zip文件，放在windows服务器某个文件夹中，解压压缩，注意解压文件后可能没有权限需要重新执行chmod命令赋予权限，文件中包含jdk，jar包，flume、canal、run.bat。

修改run.bat文件中的端口（默认8686）、内存大小，然后执行run.bat启动，注意端口不要被占用，如图。

图 3-137 Windows 环境部署

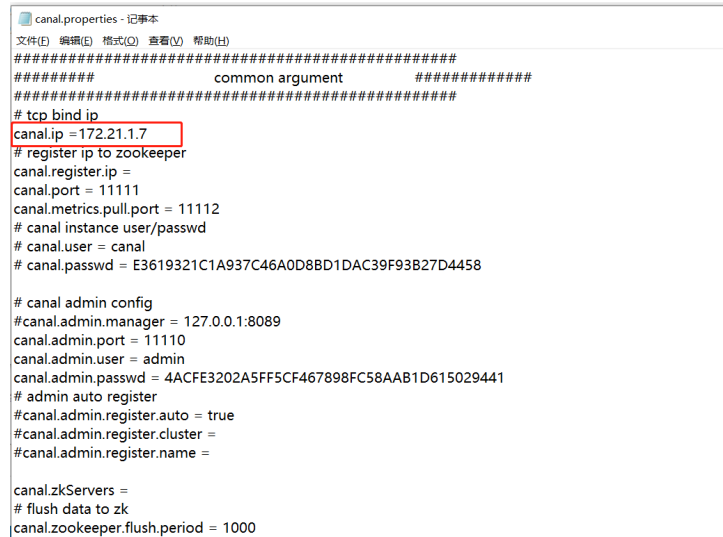


```
run.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#@%~d0
@cd %~-dp0
echo %~-dp0
set JAVA_OPTS=-Xmx1024M -Dserver.port=8686 -Desen.edi.edgenode.workdir=%~dp0/workdir/
%~dp0\jdk1.7.0_21\bin\java %JAVA_OPTS% -jar edi-flumeclient-4.4.1-SNAPSHOT.jar
```

canal 注意事项

目前canal需要手动执行canal\bin目录下startup.bat/startup来启动，并且修改压缩包下canal\conf\canal.properties配置canal.ip属性，否则无法获取canal地址，canal.ip为边缘采集节点服务器ip。

图 3-138 canal 注意事项



```
canal.properties - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#####
##### common argument #####
#####
# tcp bind ip
canal.ip = 172.21.1.7
# register ip to zookeeper
canal.register.ip =
canal.port = 11111
canal.metrics.pull.port = 11112
# canal instance user/passwd
# canal.user = canal
# canal.passwd = E3619321C1A937C46A0D8BD1DAC39F93B27D4458

# canal admin config
# canal.admin.manager = 127.0.0.1:8089
canal.admin.port = 11110
canal.admin.user = admin
canal.admin.passwd = 4ACFE3202A5FF5CF467898FC58AAB1D615029441
# admin auto register
# canal.admin.register.auto = true
# canal.admin.register.cluster =
# canal.admin.register.name =

canal.zkServers =
# flush data to zk
canal.zookeeper.flush.period = 1000
```

参数介绍

表 3-37 参数介绍

参数	说明
server.port	服务器端口，默认为8686
JAVA_OPTS=-Xmx5120M	JVM内存配置，推荐5G
esen.edi.edgenode.workdir	工作目录，默认为与jar包同级的workdir，日志文件存在工作目录中

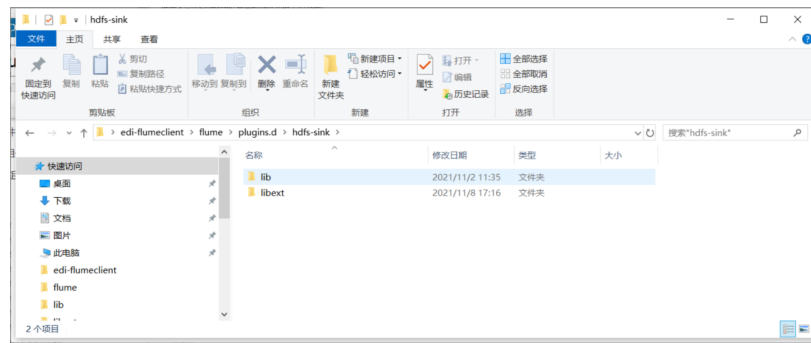
依赖 jar 包

使用不同的Flume组件需要额外添加相关的Jar包，详情见第二章依赖驱动清单。

如果系统已经安装Hadoop并配置了“HADOOP_HOME”环境变量就不需要再添加这些jar包。

可以直接将压缩包中的jar放到flume/lib目录下。建议用flume官方文档中拓展第三方插件的方式，在flume 安装目录下创建\plugins.d目录，jar包放到libext目录下，例如：

图 3-139 依赖 jar 包



3.2.8 初始化服务器

通过以上章节了解服务器如何部署在各web应用服务器上后，您可以通过如下步骤完成数据治理初始化工作。

1. 输入登录网址，如：<http://127.0.0.1:8080/edg>，进入数据治理服务器配置界面

图 3-140 配置界面



2. 勾选同意以上协议，单击“下一步”

图 3-141 下一步

The screenshot shows a configuration wizard with four steps: 1. 授权协议 (Authorization Agreement), 2. 工作目录设置 (Work Directory Settings), 3. 数据库设置 (Database Settings), and 4. 产品注册 (Product Registration). Step 2 is currently active. It features a text input field for '设置工作目录:' (Set work directory:), a green '检测' (Check) button, and a blue '下一步' (Next) button. A note below the input field states: '工作目录记录了服务器配置、数据库配置等内容，正确的设置工作目录是服务器正常自动的前提。' (The work directory records server configuration, database configuration, etc. Correctly setting the work directory is a prerequisite for the server to run normally automatically.)

设置工作目录，填写路径后单击检测按钮，保证工作目录可用

- 单击“下一步”，设置数据库

图 3-142 设置数据库

The screenshot shows the 'Database Settings' step (step 3) of the configuration wizard. It includes a '基本选项' (Basic Options) section with fields for '数据库:' (Database), '驱动类名:' (Driver class name), '链接地址:' (Connection string), '用户名:' (Username), and '密码:' (Password). A green '测试' (Test) button is located below the password field. There is also a collapsed '高级选项' (Advanced Options) section. A blue '下一步' (Next) button is at the bottom right.

此处可以单击测试，测试数据库链接。

- 单击“下一步”，注册服务器

图 3-143 注册服务器

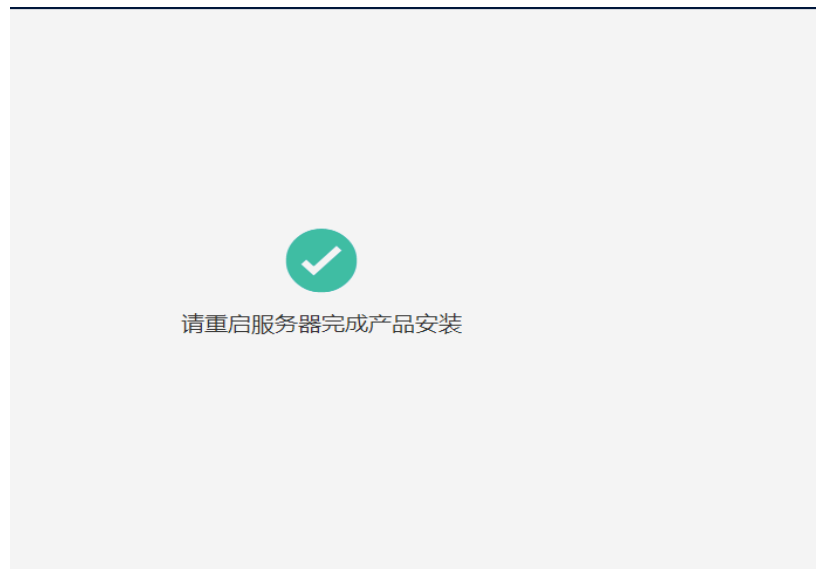


The screenshot shows a registration interface with four steps: 1. 授权协议 (Authorization Agreement), 2. 工作目录设置 (Working Directory Settings), 3. 数据库设置 (Database Settings), and 4. 产品注册 (Product Registration). The current step is 4. The form includes fields for: 生成注册申请 (Generate Registration Application) with sub-fields for *用户名称 (User Name), *项目名称 (Project Name), *联系电话 (Contact Phone), and *联系邮箱 (Contact Email). Below this is the 导入许可文件 (Import License File) section, which includes instructions to email the application to service@esensoft.com and a button to 导入许可文件 (Import License File). Navigation buttons for 上一步 (Previous Step) and 下一步 (Next Step) are at the bottom.

此处，需要填写用户名称，项目名称，联系电话和联系邮箱等，然后生成注册申请，将生成的注册申请文件导出，进行注册码申请，由具有审批权限的人员将生成的许可文件发给部署人员，然后导入许可文件。

5. 单击“下一步”，开始安装

图 3-144 开始安装



安装完成后请重启中间件

6. 进入服务器的登录界面

图 3-145 登录界面



管理员admin，初始密码为admin。

3.2.9 系统诊断

很多问题其实都是部署的时候没有正确设置某些参数，或者没有按要求修改配置文件，往往在遇到异常时很难定位到这些错误，数据治理系统能自动诊断这些文件，并加以提示。

在管理员登录界面后单击系统诊断提示的警告会出现如下诊断界面

图 3-146 系统诊断



系统进行了下列检测

表 3-38 检测

诊断项	检测内容（中文）	The detection content (English)
时区设置	必须是Asia/Shanghai，否则提示错误	必须是Asia/Shanghai，否则提示错误
file.encoding设置	必须是"UTF-8","UTF8","GB18030","GBK",否则提示错误	必须是"UTF-8",否则提示错误
sun.jnu.encoding设置	同上	同上

诊断项	检测内容 (中文)	The detection content (English)
操作系统字符集	检查文件名或文件内容是否支持GBK编码, 不支持提示错误	检查文件名或文件内容是否支持"UTF-8"编码, 不支持提示错误
java.awt.headless 设置	windows不检查, linux检查或未设置提示错误	windows不检查, linux检查或未设置提示错误
最大内存设置	32位要求大于1025M, 64位要求大于2048M	32位要求大于1025M, 64位要求大于2048M
可用内存	<%5 系统已几乎无可用内存 <30% 系统可用内存偏低 >=30% 通过	<%5 系统已几乎无可用内存 <30% 系统可用内存偏低 >=30% 通过
可用字体	必须有宋体字体, 否则不能通过	-
注册信息	系统剩余使用天数	-

3.2.10 疑难问题处理方案及数据库权限分配

如何指定工作目录

按照优先级由低到高排列

1. 通过界面初始化
windows环境在注册表下面, Linux环境在当前执行用户home目录下的.java下
2. 通过java环境变量指定
在启动参数中加上-Desen.edg.workdir=/opt/workdir/edg
3. 在war包中指定
在war包中的WEB-INF\classes\ application.properties,
指定esen.edg.workdir=/opt/workdir/edg

📖 说明

最前方的#代表注释

环境乱码如何配置

环境部署好后, 如果面包屑出现乱码现象, 将tomcat里面的参数修改即可。

路径为apache-tomcat-8.5.12/conf/server.xml, 在端口号后面加上URIEncoding="UTF-8", 改变tomcat的字符集, 如图所示;

图 3-147 环境乱码如何配置

```
<!-- Editable user database that can also be used by
      UsersDatabaseRealm to authenticate users
-->
<Resource name="UserDatabase" auth="Container"
          type="org.apache.catalina.UserDatabase"
          description="User database that can be updated and saved"
          factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
          pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>

<!-- A "Service" is a collection of one or more "Connectors" that share
      a single "Container" Note: A "Service" is not itself a "Container",
      so you may not define subcomponents such as "Valves" at this level.
      Documentation at /docs/config/service.html
-->
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
          maxThreads="150" minSpareThreads="4"/>
  -->

  <!-- A "Connector" represents an endpoint by which requests are received
      and responses are returned. Documentation at :
      Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
      Java AJP Connector: /docs/config/ajp.html
      APR (HTTP/AJP) Connector: /docs/ajp.html
      Define a non-SSL HTTP/1.1 Connector on port 8080
  -->
  <Connector port="8080" protocol="HTTP/1.1"
          connectionTimeout="30000"
          redirectPort="8443" URIEncoding="UTF-8"/>
  <!-- A "Connector" using the shared thread pool-->
  <!--
  <Connector executor="tomcatThreadPool"
          port="8080" protocol="HTTP/1.1"
          connectionTimeout="20000"
          redirectPort="8443" />
  -->

  <!-- Define a SSL HTTP/1.1 Connector on port 8443
      This connector uses the BIO implementation that requires the JSSE
      style configuration. When using the APR/native implementation, the
      OpenSSL style configuration is required as described in the APR/native
      documentation -->
  <!--
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
          maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
          clientAuth="false" sslProtocol="TLS" />
  -->

  <!-- Define an AJP 1.3 Connector on port 8009 -->
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
-->
```

Redis 部署相关

问题现象：由于某些项目服务器的操作系统是最小安装的，安装redis时会提示缺少gcc等rpm包

解决办法：

1. 如果处于外网环境（能ping通baidu.com），可直接执行命令进行安装
yum install -y gcc-c++
如果缺少其他依赖包，同样可以用yum进行安装
Make报错问题解决后，先清理上一次make产生的文件再次make
make distclean
make
2. 如果处于内网环境，则需要如下几步配置yum源实现yum安装：
 - a. 给服务器所在虚拟机或物理机插入相同版本的操作系统光盘
 - b. 在操作系统层面将光盘挂载在某一个目录下，如下图

图 3-148 挂载

```
[root@190 Packages]# df -h
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/rootvg-rootlv 50G  4.1G  46G   9% /
devtmpfs                 32G   0    32G   0% /dev
tmpfs                   32G  84K  32G   1% /dev/shm
tmpfs                   32G  553M  31G   2% /run
tmpfs                   32G   0    32G   0% /sys/fs/cgroup
/dev/sda1                494M  140M  355M  29% /boot
/dev/mapper/rootvg-lv0000 20G  862M  20G   5% /home
/dev/mapper/rootvg-workdir1v 30G  6.9G  24G  23% /workdir
/dev/mapper/rootvg-lv1440  80G  29G  52G  36% /weblogic
/dev/mapper/rootvg-tmplv  20G  42M  20G   1% /tmp
tmpfs                   6.3G  16K  6.3G   1% /run/user/0
tmpfs                   6.3G   0    6.3G   0% /run/user/1001
/dev/sr0                 3.8G  3.8G   0  100% /run/media/root/RHEL-7.2 Server.x86_64
```

mount /dev/sr0 /mnt/cdrom #/dev/sr0为光盘路径，/mnt/cdrom为挂载路径

3. 在/etc/yum.repos.d目录下备份原有文件，新建local.repo文件，后缀必须为.repo，增加如下内容：

```
[local]                #必须，唯一的repo名称
name=local            #自定义名称
baseurl=file:///mnt/cdrom    #光盘mount后的路径
enable=1              #启用该repo，0为禁用
gpgcheck=0           #不检查rpm包来源
```

睿治目前支持的系统库

Oracle库：Oracle11g、Oracle12c、Oracle19c

PG库：PostgreSQL 9.6、PostgreSQL 10.6 、PostgreSQL12.6

达梦数据库：达梦7.6、达梦8.1

Mysql数据库：Mysql8.0.27(需要支持窗口函数)

注意

1. mysql8作为驱动时url地址需要更新。参考：
jdbc:mysql://localhost:3306/test?
useUnicode=true&characterEncoding=utf8&useSSL=false&databaseTerm=SCHE
MA&nullDatabaseMeansCurrent=true
2. 一般不选数据库系统用户作为部署用户，如Oracle的system、mysql的root用户等。

睿治目前支持的中间件

Tomcat：tomcat版本需要用9.0.0.M18以上、8.5.12以上的版本

Weblogic：weblogic12c、weblogic14c

Tas：tas2.8.5

TongWeb：TongWeb6.1

睿治目前支持的浏览器

睿治目前支持的浏览器：

Chrome（v77以上）、ie11、360浏览器（v10以上）

测试通过的软硬件环境

表 3-39 测试通过的软硬件环境

总体测试环境 http://172.21.50.101:8081/ rz25linux	【数据库】Oracle19cg 【客户端】win7,win10 ie11, chrome
---	---

	【 中间件 】 tomcat9, jdk1.8
	【 中间件所在的操作系统 】 CentOS release 7
兼容性测试环境1 http://172.21.160.83:8081/rz25pg	【 数据库 】 postgresql12.6
	【 客户端 】 win7,win10 ie11, chrome
	【 中间件 】 tomcat9, jdk1.8
	【 中间件所在的操作系统 】 Windows10

元数据采集的相应配置

1. 元数据数量很大有20多万的情况，需要调整数据库的undo回滚空间至少分配30g；
2. 服务器数据很大，尽量调大服务器最大内存到4g及以上；
3. 采集的数据很大的情况，需要调整数据库的redo log 给3组，每组文件给10g；
4. 执行多个大数据量采集的计划任务，建议不要同时执行，尽量分开执行；
5. 入库的数据超过50w，服务器内存最大必须2g以上；
6. 备份恢复操作，如果元数据所有的数据超过20w以上，EDG25_EMET_METADATAHIS、EDG25_EMET_MCDATAS历史表的数据超过20w以上，建议采用数据备份来备份元数据环境，不推荐系统自带的备份恢复操作；

IPv6 部署

睿治251支持ip v6部署，此处以tomcat为例介绍如何配置

1. 修改tomcat/conf/server.xml文件，找到下图位置，添加address= ” ip v6地址 ” ，

图 3-149 修改

```
<Connector port="8080" protocol="HTTP/1.1" address="fd15:4ba5:5a2b:1008:d8b3:df29:fd2:7776"
connectionTimeout="20000"
redirectPort="8443" />
```

2. 其余配置和ip v4一样，可参考前面章节
3. 初始化睿治，在数据库设置这里，填写链接地址时，ipv6地址要用[]括起来，即标红部分。如使用oracle11g，则参照下面写法
jdbc:oracle:thin:@//[fd15:4ba5:5a2b:1008:d8b3:df29:fd2:7776]:1523/orcl
4. 注册成功后，ipv6访问地址也要用[]括起来，如 http://
[fd15:4ba5:5a2b:1008:d8b3:df29:fd2:7776]:8080/edg

数据库权限分配

表 3-40 权限分配

数据库	分配权限
-----	------

Oracle	connect、resource、create any view 说明 oracle12c及以上需要alter user xxx quota unlimited on users
--------	---

3.3 ABI 一站式数据分析平台

3.3.1 安装环境

注意事项

1. ABI系列的产品要求中间件支持servlet3.0，所以需要jdk1.7及以上。ABI5.3.1及以上版本需要jdk1.8 及以上。
2. 目前ABI5.3.1支持的中间件有：Tomcat 7.0.62及以上/weblogic12c/中创中间件/金蝶/东方通TongWeb V6.1中间件/华宇tas2.8.5中间件
3. 目前ABI5.3.1支持飞腾、龙芯、兆芯、鲲鹏等CPU，支持联想、长城等服务器
4. 为保证能正常获取注册信息，请使用管理员权限运行ABI
5. 为保证部署能顺利完成，linux下部署要求部署人员必须具备linux基础

服务器配置推荐列表

服务器推荐原则：

客户的需求多种多样，重点关注如下几个关键指标；

用户数：一般情况，用户量每增加3000户，建议增加一台中间件服务器；每增加6000户，建议增加一台数据库服务器；并发用户量按照用户量的10%估算；

数据量：数据量和报表复杂度越大，对数据库的要求就越高，按照不同数据量和报表复杂度对数据库配置进行了推荐；如果数据量超过亿行，建议使用分布式数据库进行部署；

操作：推荐配置按照常用使用推荐，如果用户有特殊操作（比如大数据量的导出等），建议提升一档推荐配置；

- **小型配置**

推荐客户：小范围试用和用户量较小的客户环境

用户数小于100，并发用户数小于10，后端待分析数据源单表数据量小于10万行，后端生成sql复杂度为连接个数+子查询个数 <=2

表 3-41 小型配置

配置项目	单项配置说明
应用/数据库服务器	DELL PowerEdge R240或其他同类产品 CPU: 英特尔® 至强® E-2224处理器 (4C/4T) 内存: 16G 硬盘: RAID5 1T 网络: 100M
操作系统及软件	操作系统: Windows、CentOS 中间件: Tomcat8 数据库: Oracle11

- **中等配置**

推荐客户: 中型企业、一般机构

用户数小于200, 并发用户数小于20, 后端待分析数据源单表数据量小于100万行;

用户数小于500, 并发用户数小于50, 后端待分析数据源单表数据量小于10万行;

后端生成sql复杂度为 $2 < \text{连接个数} + \text{子查询个数} \leq 5$

表 3-42 中等配置

配置项目	单项配置说明
应用服务器	DELL PowerEdge R340或其他同类产品 CPU: 英特尔® 至强® E-2234处理器 (4C/8T) 内存: 32G 硬盘: RAID5 1T 网络: 100M
数据库服务器	DELL PowerEdge R340或其他同类产品 CPU: 英特尔® 至强® E-2274G处理器 (4C/8T) 内存: 64G 硬盘: RAID5 1T 网络: 100M
操作系统及软件	操作系统: Windows、CentOS、Red Hat 中间件: Tomcat8、Weblogic12 数据库: Oracle11

- **大型配置**

推荐客户: 大型企业、大型机构

用户数小于2000, 并发用户数小于200, 后端待分析数据源单表数据量小于1000万行;

用户数小于5000，并发用户数小于500，后端待分析数据源单表数据量小于100万行；

后端生成sql复杂度为 $5 < \text{连接个数} + \text{子查询个数} \leq 10$ ；

表 3-43 配置 3

配置项目	单项配置说明
应用服务器 * 2	DELL PowerEdge R440或其他同类产品 CPU: 英特尔® 至强®银牌 4208处理器 (8C/16T) 内存: 32G 硬盘: RAID5 1T 网络: 1000M
数据库服务器	DELL PowerEdge R540或其他同类产品 CPU: 英特尔® 至强®金牌 6226处理器 (12C/24T) 内存: 128G 硬盘: 1.92TB SSD SATA 网络: 1000M
redis服务器 (推荐redis服务器单独部署, 如需和中间件服务器组合部署硬件不能低于两者的组合要求)	CPU: 英特尔® 至强® E-2224处理器 (4C/4T) 内存: 16G 硬盘: RAID5 100G 网络: 1000M
操作系统及软件	操作系统: Windows、CentOS、Red Hat 中间件: Tomcat8、Weblogic12 数据库: Oracle11

- **超大型配置**

推荐客户: 全国性机构、全国性企业

用户数小于10000，并发用户数小于1000，后端待分析数据源单表数据量小于1亿行, 后端生成sql复杂度为 $\text{连接个数} + \text{子查询个数} < 30$ ；

表 3-44 配置 4

配置项目	单项配置说明
应用服务器 * 4	DELL PowerEdge R740或其他同类产品 CPU: 英特尔® 至强®金牌 5215处理器 (10C/20T) 内存: 64G 存储: RAID5 1T 网络: 1000M以上
数据库服务器 * 2	DELL PowerEdge R740或其他同类产品 CPU: 英特尔® 至强®金牌 5218N处理器 (16C/32T) 内存: 256G 存储: 光纤存储 网络: 1000M以上
redis服务器 * 3 (推荐redis服务器单独部署, 如果需和中间件服务器组合部署硬件不能低于两者的组合要求)	CPU: 英特尔® 至强® E-2224处理器 (4C/4T) 内存: 16G 硬盘: RAID5 100G 网络: 1000M
推荐操作系统及软件	操作系统: Windows、CentOS、Red Hat 中间件: Weblogic12 数据库: Oracle11

• 其他情况说明

- a. 用户量少, 数据量大: 按照对应用户量的应用服务器配置提升一档, 数据库服务器以数据量为准;
- b. 更多用户, 更大数据量: 以全国性机构推荐配置为基准, 每多3000用户增加一台应用服务器, 每多6000用户再增加一台数据库服务器; 更大数据量建议咨询大数据产品部;

客户端配置

表 3-45 ABI 客户端

所属项	说明
CPU	不低于酷睿i3
内存	不小于4G
网络	不低于100KB/s传输速度

表 3-46 客户端浏览器

操作系统	浏览器
Win7、Win10	IE11、Chrome

📖 说明

使用IE时要用正常模式，不要用兼容模式。

名词解释

ABI5.3.1解释：

ABI5.3.1是指ABI5.3.1报表服务器的版本，对应的在服务器登录界面上的version信息。

3.3.2 导出截图功能要求

ABI5.3.1中结果导出用到phantomjs进行截图，这个工具对操作系统有要求，因此，需要使用导出功能的小伙伴们，ABI5.3.1要部署在下面环境中：

Windows

版本要求：Windows server 2008及以上，Win7及以上

Linux

操作系统要求有图形化界面。

版本要求：Linux内核版本2.6.32-358（CentOS 6.4或RHEL 6 Update 4）及以上

3.3.3 TOMCAT 中部署 ABI

3.3.3.1 Windows 操作系统

下面讲述的是在Windows环境下安装部署ABI的整个过程。

安装 JDK

安装JAVA运行环境（jdk1.8或以上版本），如下图所示：

图 3-150 安装 JDK1



图 3-151 安装 JDK2



安装过程中，可以默认设置，直接NEXT完成安装，具体过程略。

安装 TOMCAT

安装TOMCAT有两种方法。一种是使用TOMCAT的安装版程序，一种是使用TOMCAT的解压版文件。这两种方法的区别在于，通过安装版程序安装TOMCAT，不需要设置JAVA环境变量，使用解压版文件需要设置环境变量。

现在基本都使用TOMCAT的解压版程序，只需将TOMCAT压缩文件解压到某一目录即可。

配置环境变量

如果是解压版的TOMCAT，需要在startup.bat中设置JAVA环境变量。

Set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_221,红色路径改为实际JDK安装路径。

参数优化

修改内存大小一般在启动文件startup.bat中设置，设置内存不能低于如下数值：

说明

jdk1.8开始取消了PermGen，取而代之的是Metaspace

```
Set JAVA_OPTS=-Xmx2048m -Xms2048m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -
Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Duser.language=zh -
Duser.country=CN
```

部署 ABI

找到TOMCAT安装目录中的webapps文件夹，删除除ROOT文件夹外的其他文件和文件夹。

将ABI服务器的WAR包文件分别复制到上述的webapps文件夹中。假设ABI服务器的WAR包文件名为ABI。

启动服务器

运行TOMCAT解压目录中的bin\startup.bat，启动TOMCAT服务。

在IE地址栏中输入报表服务器地址，输入地址时，应带上TOMCAT的端口和应用的目录，即WAR包的目录名，完成服务器的初始化工作。

3.3.3.2 Linux 操作系统

在Linux中部署与windows中基本相同，下面以TOMCAT为例讲解在linux下部署的步骤。

Linux 下使用“变量名=变量值”设置变量，并使用 export 命令将其导出为环境变量。下面是安装并配置jdk、tomcat的环境变量。

安装 JDK

```
#cp /path/to/ jdk-8u131-linux-x64.tar.gz /usr/local/ (把jdk复制到要安装的位置，如：usr/local下面)
#cd /usr/local (到usr/local目录下)
#chmod a+x jdk-8u131-linux-x64.tar.gz (执行命令)
#tar -xzf jdk-8u131-linux-x64.tar.gz (执行命令)
```

然后会提示是否确认安装，这是输入yes，确认即可。

安装 TOMCAT

同上，把tomcat (apache-tomcat-7.0.70.zip) 复制到你想要安装的位置，如：usr/local下

```
# unzip apache-tomcat-7.0.70.zip (解包) 到此既可以完成安装了。
```

配置环境变量

在startup.sh或catalina.sh的开头位置设置JAVA_HOME

```
JAVA_HOME=/usr/local/jdk1.8.0_131
```

参数优化

修改内存大小一般在启动文件startup.sh中设置，设置内存不能低于如下数值：

📖 说明

jdk1.8开始取消了PermGen，取而代之的是Metaspace

```
export JAVA_OPTS=" -Xmx2048m -Xms2048m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -  
Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -  
Duser.language=zh -Duser.country=CN"
```

部署 ABI

将ABI(此名可以更改).war包复制到\Tomcat \webapps文件夹下

修改tomcat/bin 目录下的catalina.sh

启动服务器

启动tomcat

```
# ./startup.sh
```

访问报表服务器，形如：<http://127.0.0.1:8080/ABI>，完成服务器的初始化工作。后缀ABI就是部署时war包的名称，如有更改，请对应更改。

3.3.3.3 TOMCAT 常用配置

修改端口

TOMCAT安装时默认的端口8080，TOMCAT安装版可以在安装中修改。对于安装完成后和解压版TOMCAT如何修改呢？

找到TOMCAT安装目录下的\conf\server.xml文件，用记事本打开，搜索安装时设置的端口值或默认的值8080，将其替换为要设置的端口值。

在修改端口时要注意，先要停止TOMCAT服务，再来修改。修改完成后，重启TOMCAT服务，即可通过修改后的端口访问WEB应用。

JVM 设置

TOMCAT默认内存最大内存为128M，在大用户量情况下容易出现内存不够的情况，所以需要调整。

修改内存大小一般在启动文件startup.bat或startup.sh中设置，设置内存不能低于如下数值：

```
Set JAVA_OPTS= -Xmx2048m -Xms2048m -XX:MaxMetaspaceSize=512m ( Windows )  
JAVA_OPTS=" -Xmx2048m -Xms2048m -XX: MaxMetaspaceSize =512m" ( Linux )
```

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，在64bit环境下分配8G内存。

注意：某些JDK版本或操作系统限制最大堆内存，不支持分配超过1.5G的内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

线程调整

表 3-47 线程调整

属性名	描述
maxThreads	Tomcat使用线程来处理接收的每个请求。这个值表示Tomcat可创建的最大的线程数。
acceptCount	指定当所有可以使用的处理请求的线程数都被使用时，可以放到处理队列中的请求数，超过这个数的请求将不予处理。
minSpareThreads	Tomcat初始化时创建的线程数。
maxSpareThreads	一旦创建的线程超过这个值，Tomcat就会关闭不再需要的socket线程。

上述参数在conf/ server.xml中进行设置，TOMCAT上述默认值如下：

```
maxThreads: 150  
acceptCount: 100  
minSpareThreads: 25  
maxSpareThreads: 75
```

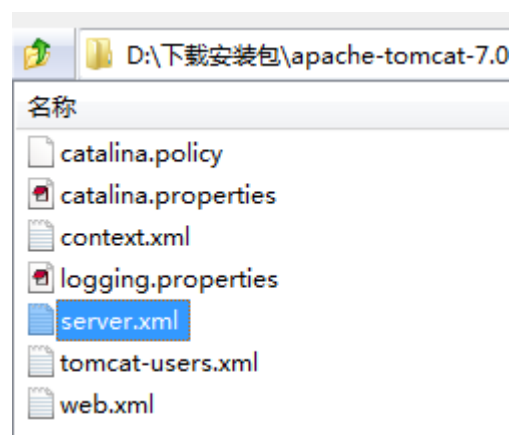
在一般情况下够用，如果同时并发用户数量很大，需要调整maxThreads参数，调整原则为每秒用户请求数*请求操作执行时间，比如每秒有100个请求，每个请求执行时间为1秒，maxThreads设置为100即可。同时要适当调整maxSpareThreads参数，不要关闭socket线程后又马上生成的操作。

Tomcat 修改应用上下文根为空

因为tomcat控制台http://IP:8080/默认上下文根为空，所以在修改应用的上下文根为空时需要修改tomcat控制台上上下文根不为空。

进入tomcat\ conf路径下，修改server.xml文件，

图 3-152 修改



在文件中找到Host name="localhost"代码段，在该代码下增加如下代码。

```
<Context path="/ROOT" docBase="E:\apache-tomcat-7.0.70\webapps\ROOT" reloadable="true"/>  
<Context path="" docBase="E:\apache-tomcat-7.0.70\webapps\ABI" reloadable="true"/>
```

图 3-153 代码

```
121 <Realm className="org.apache.catalina.realm.UserDatabaseRealm"  
122     resourceName="UserDatabase"/>  
123 </Realm>  
124  
125 <Host name="localhost" appBase="webapps"  
126     unpackWARs="true" autoDeploy="true">  
127 <Context path="/ROOT" docBase="E:\apache-tomcat-7.0.70\webapps\ROOT" reloadable="true"/>  
128 <Context path="" docBase="E:\apache-tomcat-7.0.70\webapps\pmeta" reloadable="true"/>  
129  
130 <!-- SingleSignOn valve, share authentication between web applications  
131     Documentation at: /docs/config/valve.html -->  
132 <!--
```

上面红色字体为修改tomcat控制台上下文根。蓝色字体为修改应用上下文根。上文参数解释：

Context path= 后面跟上下文根（应用的上下文根为""）。

docBase= 后面跟应用路径（ROOT为控制台，ABI为解压后的war包名）

根据上面的讲解自己修改路径和上下文根，以上可做参考。

配置完成后保存，重启tomcat，访问http://IP:8080/ROOT控制台及http://IP:8080/应用，如果发现依然访问控制台，请重启tomcat。

3.3.4 Weblogic 中部署 ABI

3.3.4.1 在 Weblogic 中创建 Domain

本章节是以Weblogic 12.2.1版本为例，介绍整个配置过程。

执行安装目录下/home/weblogic/Oracle/Middleware/Oracle_Home/wlserver/common/bin/config.sh

图 3-154 创建 1



按照实际情况修改路径

图 3-155 创建 2



设置账户密码，下一步

图 3-156 创建 3



选择开发或生产模式，下一步

图 3-157 创建 4



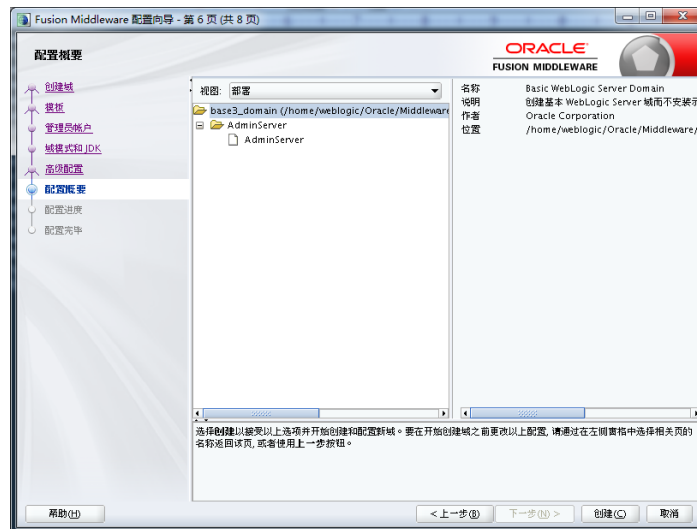
此处勾选可以设置端口和集群配置，默认下一步，端口使用7001

图 3-158 创建 5



单击创建开始创建新域

图 3-159 创建 6



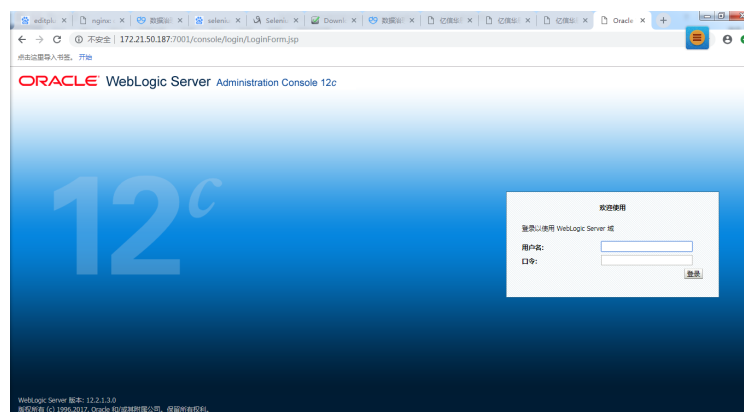
通过startWebLogic.sh(cmd)启动weblogic，直到窗口中出现listening on port 7001（端口号根据实际配置会不一样），表明服务已经成功启动。这时就可以在IE浏览器里键入http://localhost:7001/console，进入Weblogic控制台，输入您的管理员名和密码，即可进行控制台管理。

3.3.4.2 部署 ABI

Weblogic 12c 中部署

Weblogic管理控制台访问地址：<http://服务器IP:7001/console>。在IE浏览器里键入该地址，进入Weblogic控制台登录页面，输入用户名和密码进行登录。

图 3-160 部署 1



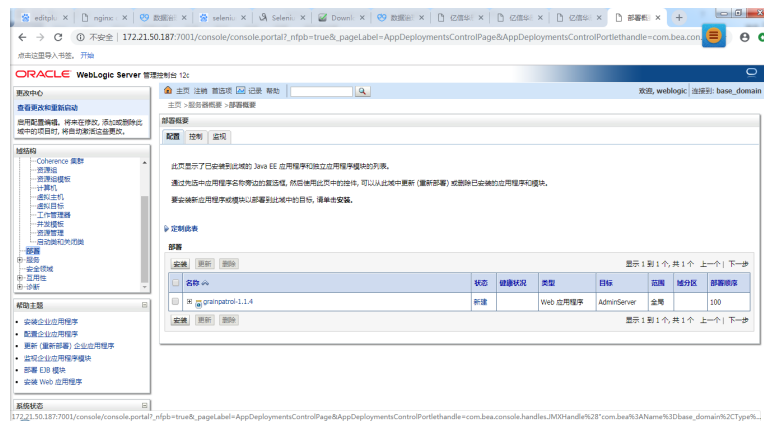
登录成功后，就出现Weblogic的控制台管理界面，如下：

图 3-161 部署 2



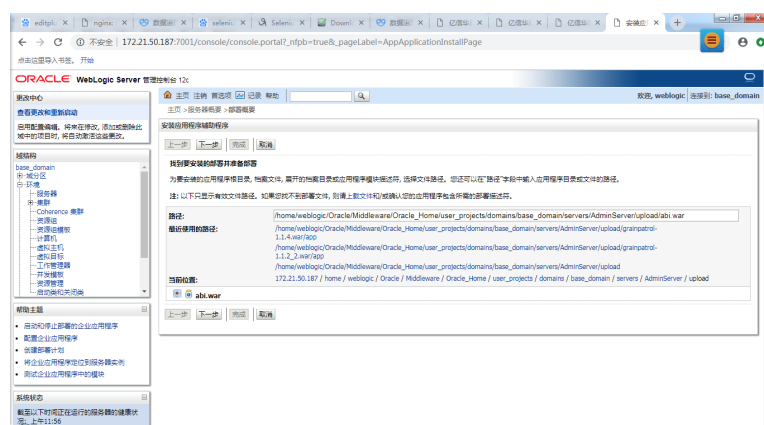
单击“部署”节点

图 3-162 部署 3



单击安装按钮

图 3-163 部署 4



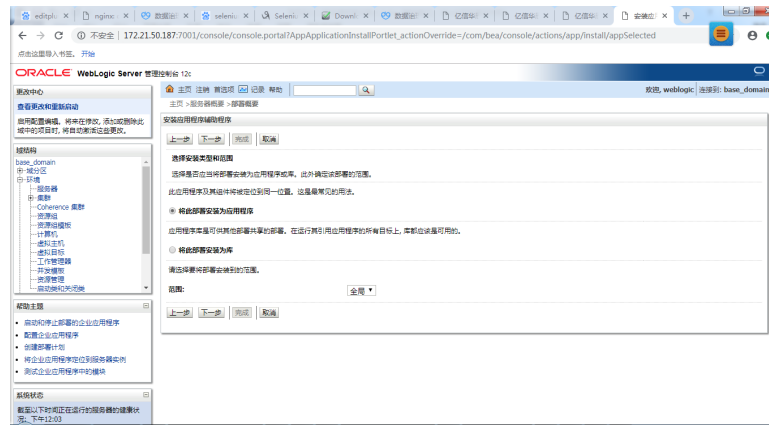
选择服务器的war包，选择下一步

图 3-164 部署 5



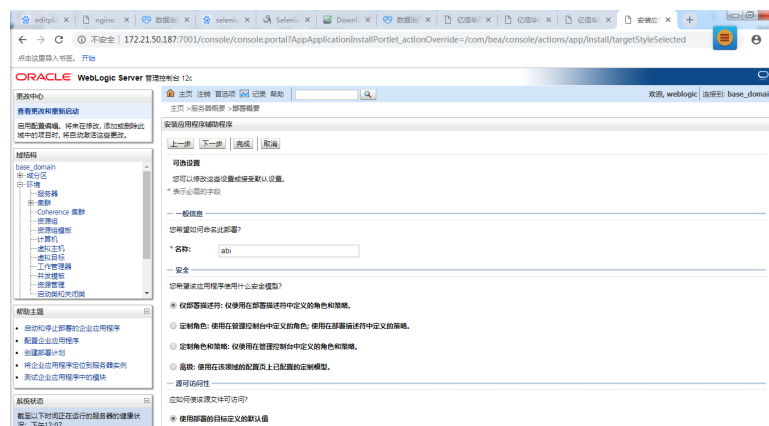
选择下一步

图 3-165 部署 6



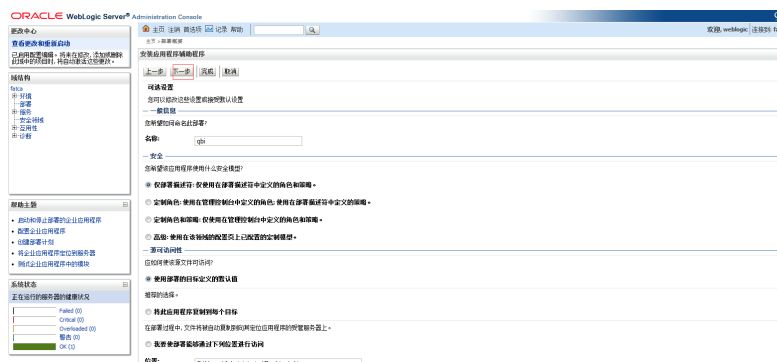
单击“下一步”按钮。

图 3-166 部署 7



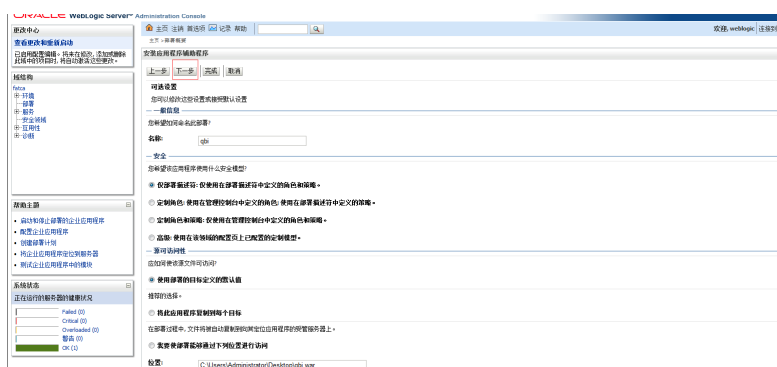
选择完成

图 3-167 部署 8



单击“下一步”按钮。

图 3-168 部署 9



单击“完成”按钮。

参数设置

在startWebLogic.sh (cmd) 中, 增加

说明

jdk1.8开始取消了PermGen, 取而代之的是Metaspace

```
Set JAVA_OPTIONS=-Xmx1024m -Xms1024m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK (Windows环境)  
JAVA_OPTIONS=" -Xmx1024m -Xms1024m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK" (Linux环境)
```

其中, 设置内存不能低于上述值。

3.3.4.3 Weblogic 单机双节点集群部署

说明

本章节介绍了Weblogic如何配置集群, 也介绍了在集群部署方式下, AB15.3.1应用配置集群。部署weblogic集群是为了使用weblogic的proxy代理, 并不使用weblogic集群的任何特性

集群 (Cluster) 是由两台或多台节点机 (服务器) 构成的一种松散耦合的计算节点集合, 为用户提供网络服务或应用程序(包括数据库、Web服务和文件服务等)的单一客

户视图，同时提供接近容错机的故障恢复能力。集群系统一般通过两台或多台节点服务器系统通过相应的硬件及软件互连，每个群集节点都是运行其自己进程的独立服务器。这些进程可以彼此通信，对网络客户机来说就像是形成了一个单一系统，协同起来向用户提供应用程序、系统资源和数据。除了作为单一系统提供服务，集群系统还具有恢复服务器级故障的能力。集群系统还可通过在集群中继续增加服务器的方式，从内部增加服务器的处理能力，并通过系统级的冗余提供固有的可靠性和可用性。

ABI5.3.1支持集群部署，下面以Weblogic12c为例讲解如何部署集群，并在集群中部署服务。

创建新 Domain

执行安装目录下/home/weblogic/Oracle/Middleware/Oracle_Home/wlserver/common/bin/config.sh

图 3-169 创建 1



选择模板

图 3-170 模板



配置管理员用户名和密码

输入Weblogic管理员管理服务器（AdminServer）的用户名和密码，在此以“weblogic”作为登录管理服务器的用户名和密码，单击“下一步”按钮

图 3-171 配置 1



配置域启动模式和 JDK

选择“生产模式”和“JDK”后，单击“下一步”按钮

图 3-172 JDK



选择管理服务器，节点管理器和拓扑

选择“是”后，单击“下一步”按钮

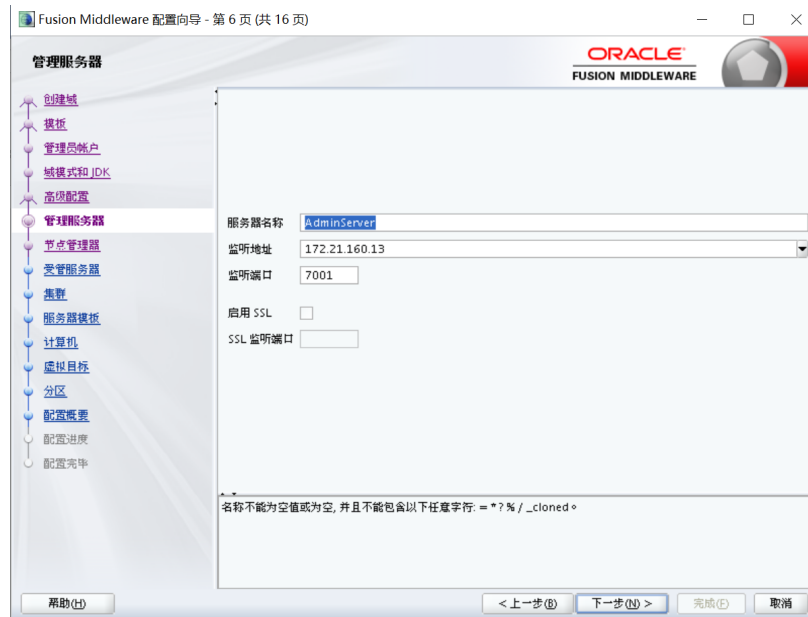
图 3-173 拓扑



配置管理服务器

输入Administrator Server的名称，监听地址，监听端口，如果需要SSL支持的话可以在“SSL enabled”后面的复选框上打勾，配置SSL监听端口，单击“下一步”按钮

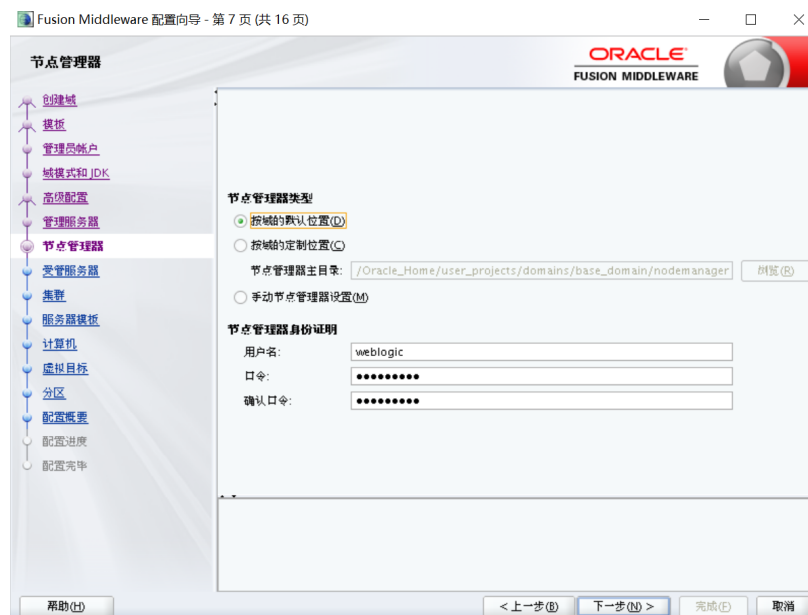
图 3-174 配置管理服务器



根据需要修改端口（默认为7001）

设置节点管理器

图 3-175 设置节点管理器



配置受管服务器

添加3个server，设置监听地址、监听端口

图 3-176 配置受管服务器



配置集群

添加集群，设置集群地址

图 3-177 集群



直接下一步

图 3-178 下一步 1



直接下一步

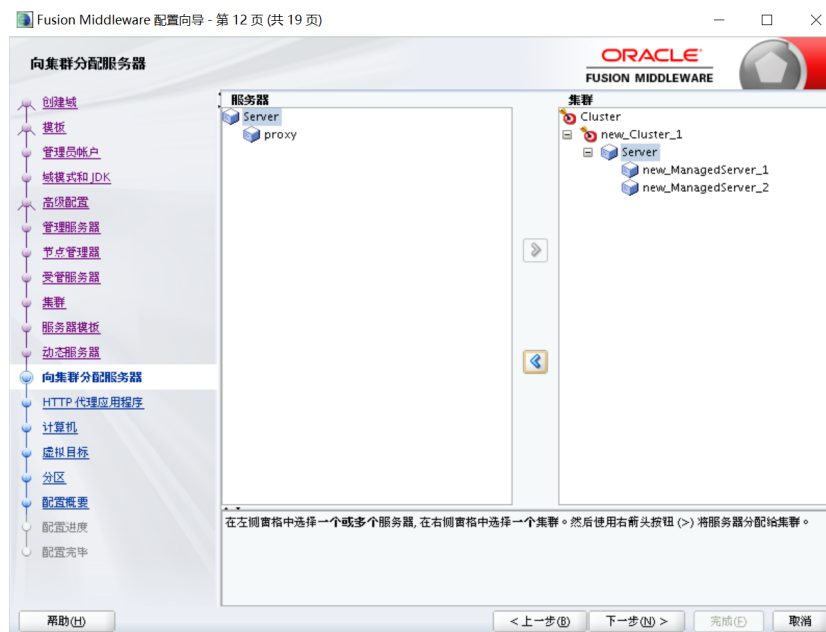
图 3-179 下一步 2



向集群分配服务器

选择左面列表中的Managed Server，将其添加到右面的列表中，这里代理服务器不要添加到右边，单击“下一步”按钮

图 3-180 向集群分配服务器



创建 HTTP 代理应用程序

勾选“创建Http代理”，代理服务器选择前面没有加入集群的Proxy，单击“下一步”按钮

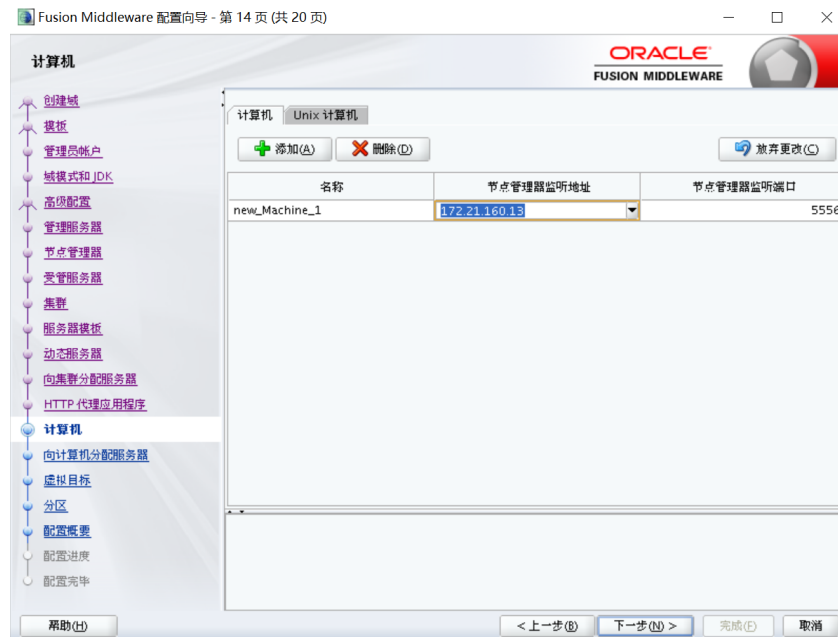
图 3-181 创建 HTTP 代理应用程序



配置计算机

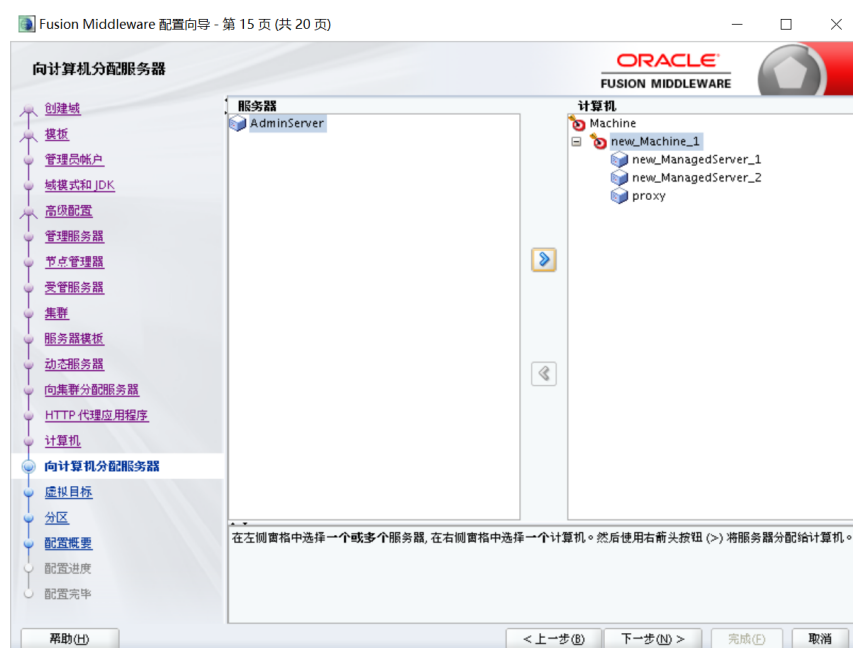
添加1台计算机，按之前地址配置，端口采用默认，单击“下一步”按钮

图 3-182 配置计算机



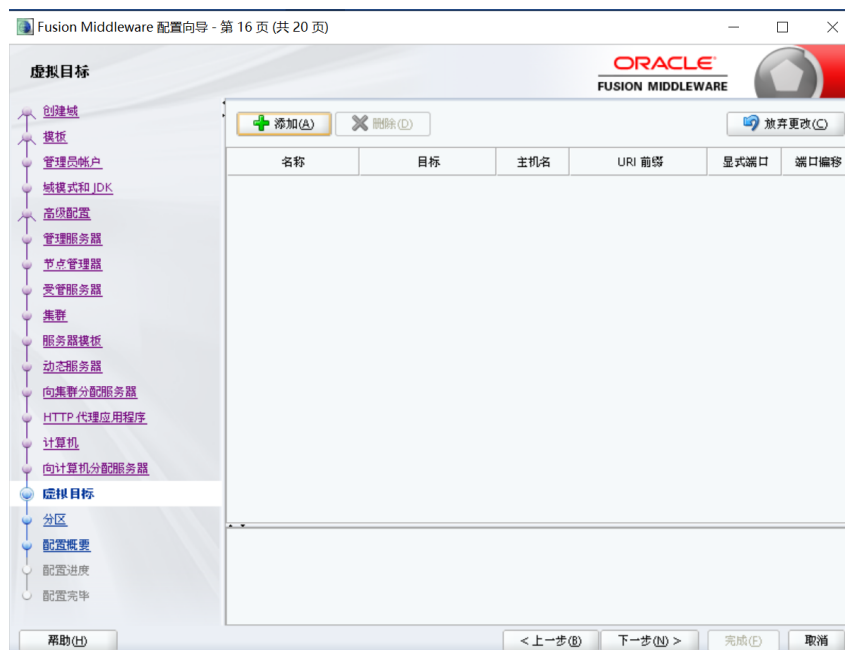
向计算机分配服务器

图 3-183 分配 1



直接下一步

图 3-184 分配 2



直接下一步

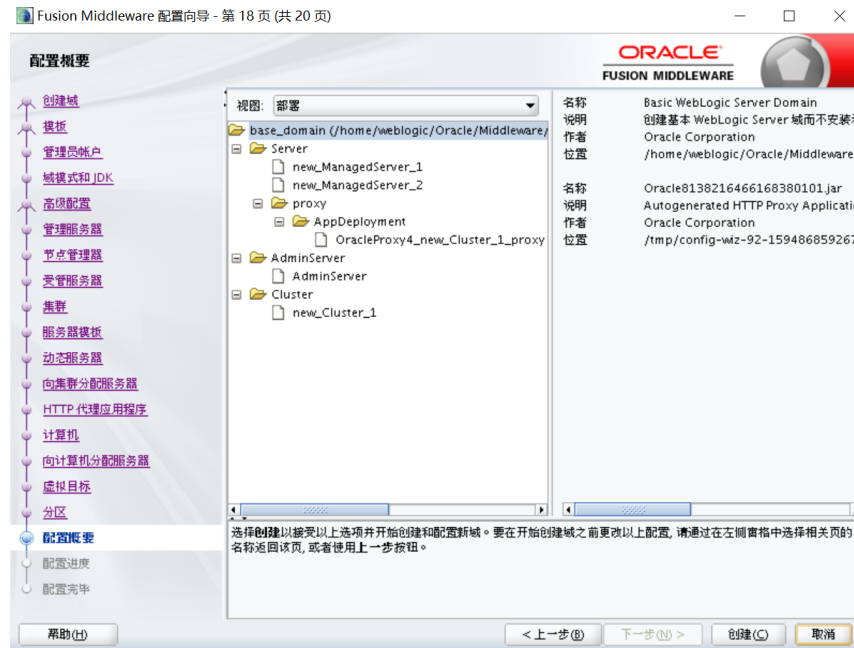
图 3-185 分配 3



检查配置概要

在此检查前面所做的配置，如果没有问题单击“下一步”按钮

图 3-186 检查



创建 Weblogic 域

单击“创建”按钮，创建无误后，单击“完成”按钮

图 3-187 创建 1

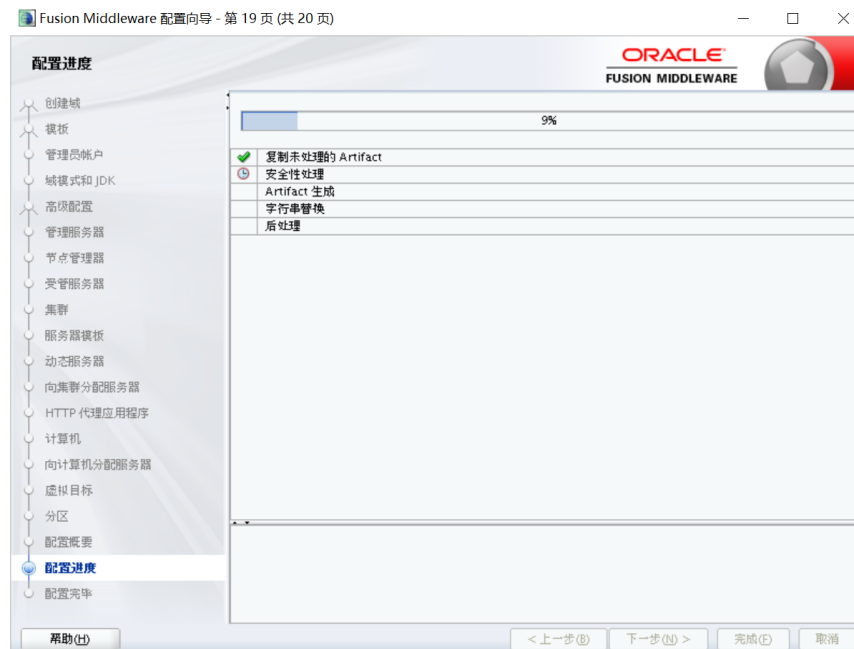


图 3-188 创建 2

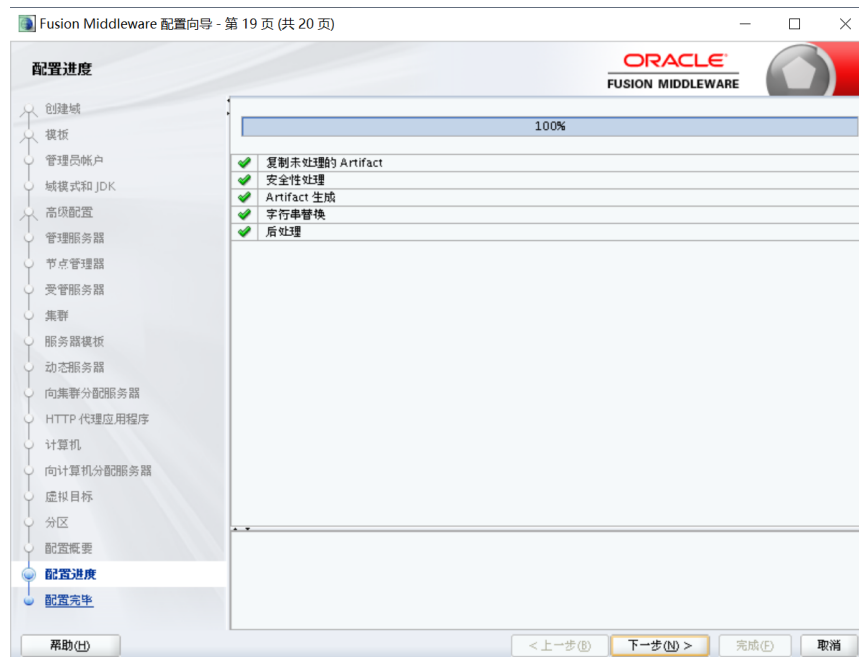


图 3-189 创建 3



3.3.4.4 Weblogic 双机集群部署

📖 说明

本章节介绍了Weblogic如何配置集群，也介绍了在集群部署方式下，ABI5.3.1应用配置集群。部署weblogic集群是为了使用weblogic的proxy代理，并不使用weblogic集群的任何特性；

集群（Cluster）是由两台或多台节点机（服务器）构成的一种松散耦合的计算节点集合，为用户提供网络服务或应用程序(包括数据库、Web服务和文件服务等)的单一客户视图，同时提供接近容错机的故障恢复能力。集群系统一般通过两台或多台节点服

务器系统通过相应的硬件及软件互连，每个群集节点都是运行其自己进程的独立服务器。这些进程可以彼此通信，对网络客户机来说就像是形成了一个单一系统，协同起来向用户提供应用程序、系统资源和数据。除了作为单一系统提供服务，集群系统还具有恢复服务器级故障的能力。集群系统还可通过在集群中继续增加服务器的方式，从内部增加服务器的处理能力，并通过系统级的冗余提供固有的可靠性和可用性。

ABI5.3.1支持集群部署，下面以Weblogic12c为例讲解如何部署集群，并在集群中部署服务。

创建新 Domain

执行安装目录下/home/weblogic/Oracle/Middleware/Oracle_Home/wlserver/common/bin/config.sh

图 3-190 创建新 Domain



选择模板

图 3-191 选择模板



配置管理员用户名和密码

输入Weblogic管理员管理服务器（AdminServer）的用户名和密码，在此以“weblogic”作为登录管理服务器的用户名和密码，单击“下一步”按钮

图 3-192 配置管理员用户名和密码



配置域启动模式和 JDK

选择“生产模式”和“JDK”后，单击“下一步”按钮

图 3-193 配置域启动模式和 JDK



选择管理服务器，节点管理器和拓扑

图 3-194 选择管理服务器，节点管理器和拓扑



配置管理服务器

输入Administrator Server的名称，监听地址，监听端口，如果需要SSL支持的话可以在“SSL enabled”后面的复选框上打勾，配置SSL监听端口，单击“下一步”按钮

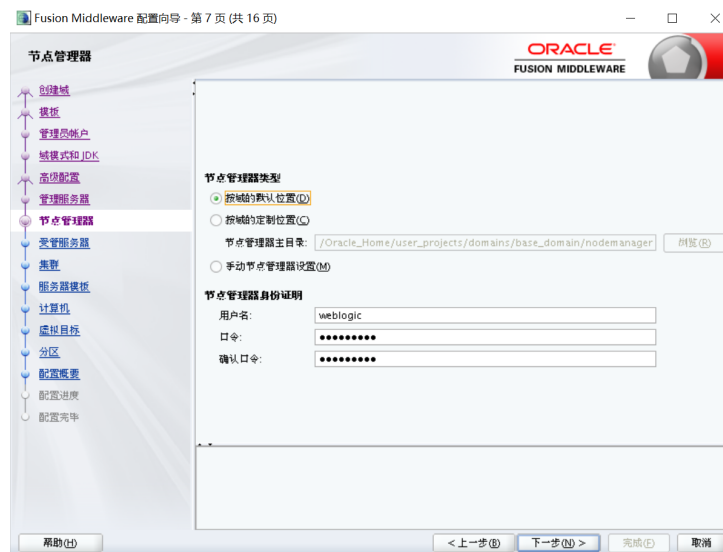
图 3-195 配置管理服务器



根据需要修改端口（默认为7001）

设置节点管理器

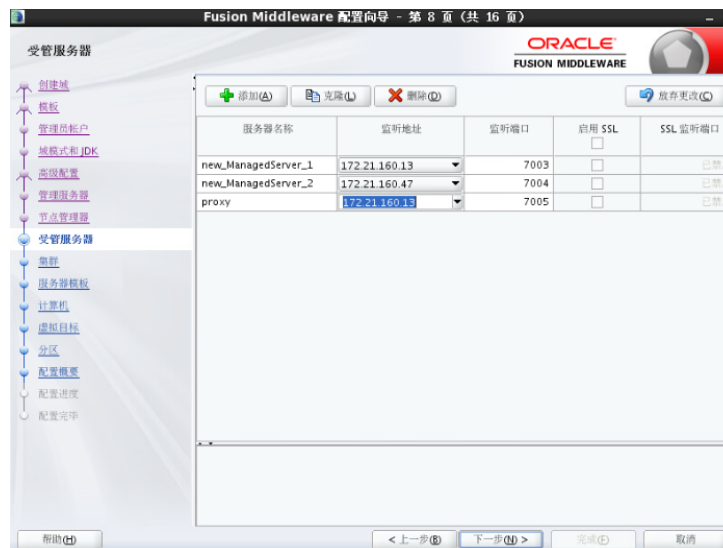
图 3-196 设置节点管理器



配置受管服务器

添加3个server，设置监听地址、监听端口

图 3-197 配置受管服务器



配置集群

添加集群，设置集群地址

图 3-198 配置集群 1



直接下一步

图 3-199 配置集群 2



直接下一步

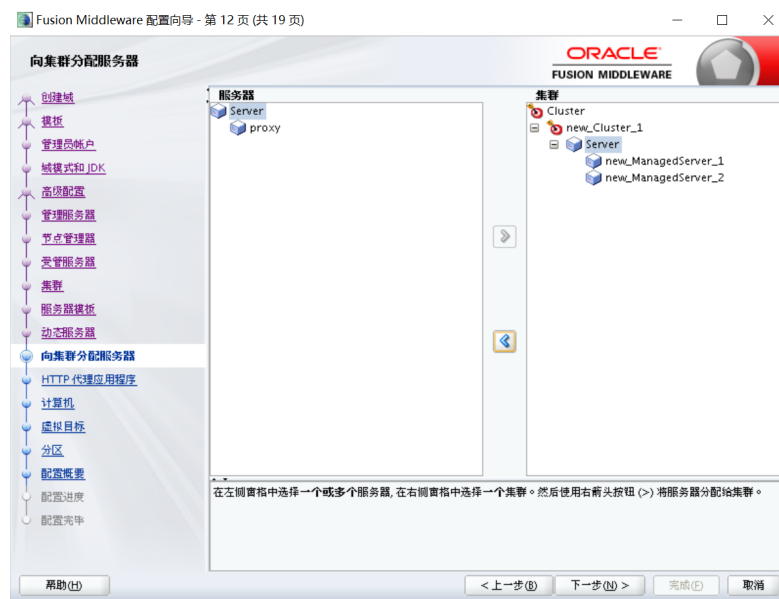
图 3-200 配置集群 3



向集群分配服务器

选择左面列表中的Managed Server，将其添加到右面的列表中，这里代理服务器不要添加到右边，单击“下一步”按钮

图 3-201 向集群分配服务器



创建 HTTP 代理应用程序

勾选“创建Http代理”，代理服务器选择前面没有加入集群的Proxy，单击“下一步”按钮

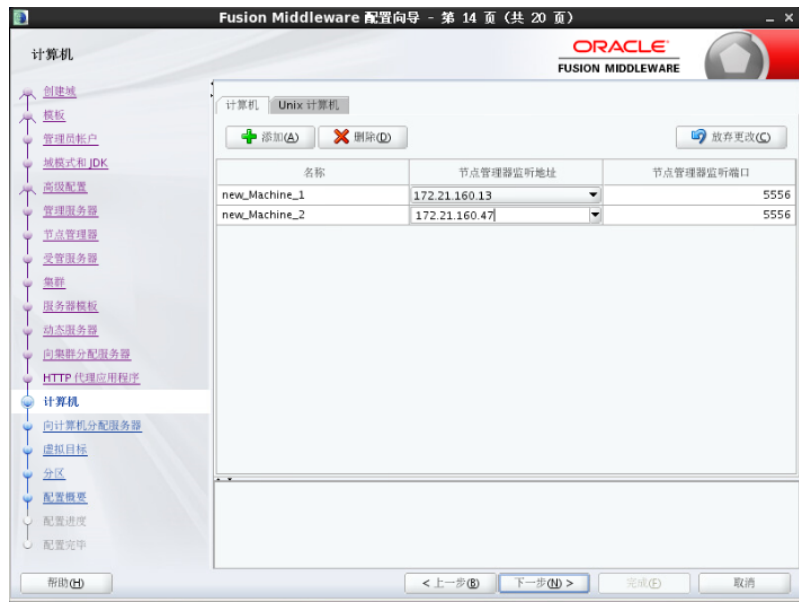
图 3-202 创建 HTTP 代理应用程序



配置计算机

添加2台计算机，端口采用默认，单击“下一步”按钮

图 3-203 配置计算机



3.3.4.5 向计算机分配服务器、检查配置概要

向计算机分配服务器

图 3-204 分配服务器 1

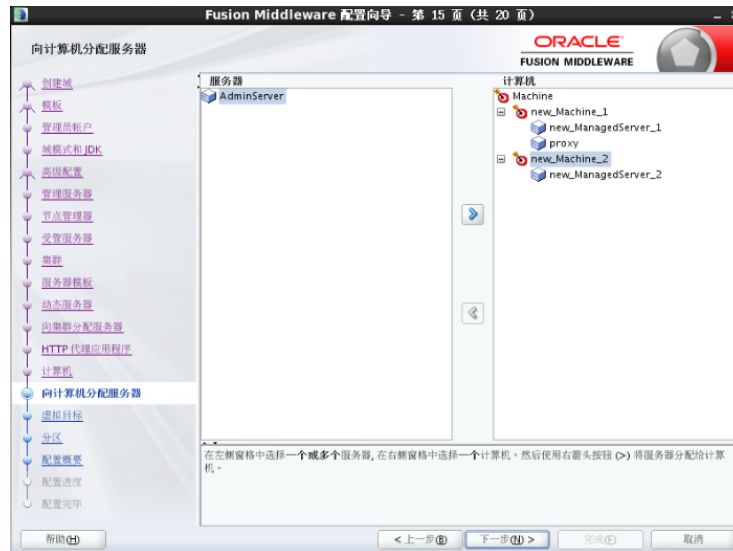


图 3-205 分配服务器 2

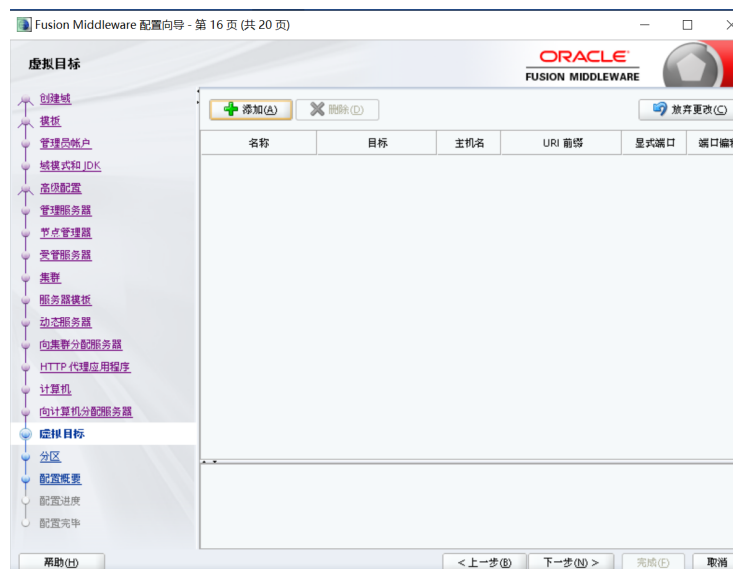
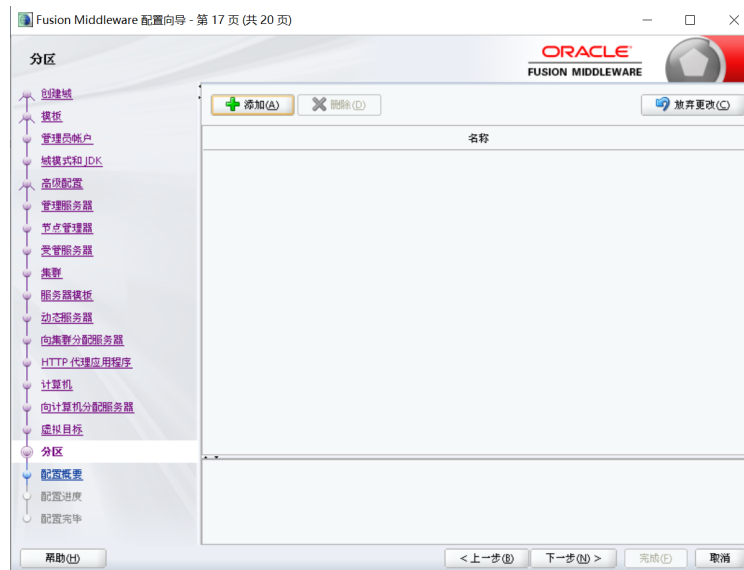


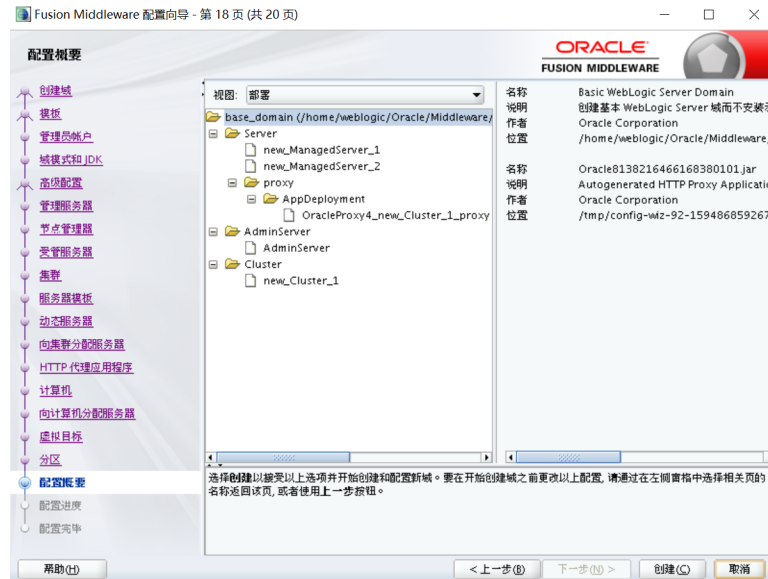
图 3-206 分配服务器 3



检查配置概要

检测前配置。

图 3-207 检查配置概要



配置完成后如果没有问题单击“下一步”按钮。

3.3.4.6 创建 Weblogic 域、集群中辅节点操作

创建 Weblogic 域

单击“创建”按钮

图 3-208 创建 Weblogic 域 1

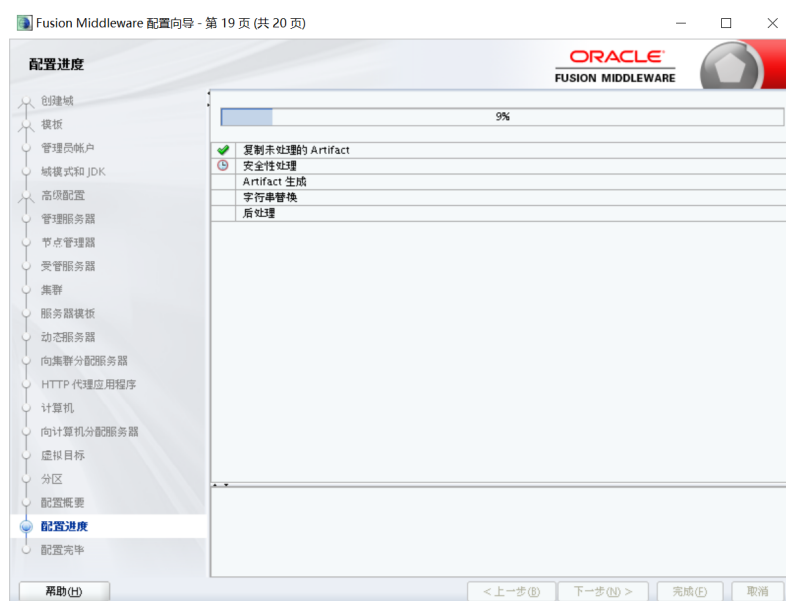


图 3-209 创建 Weblogic 域 2

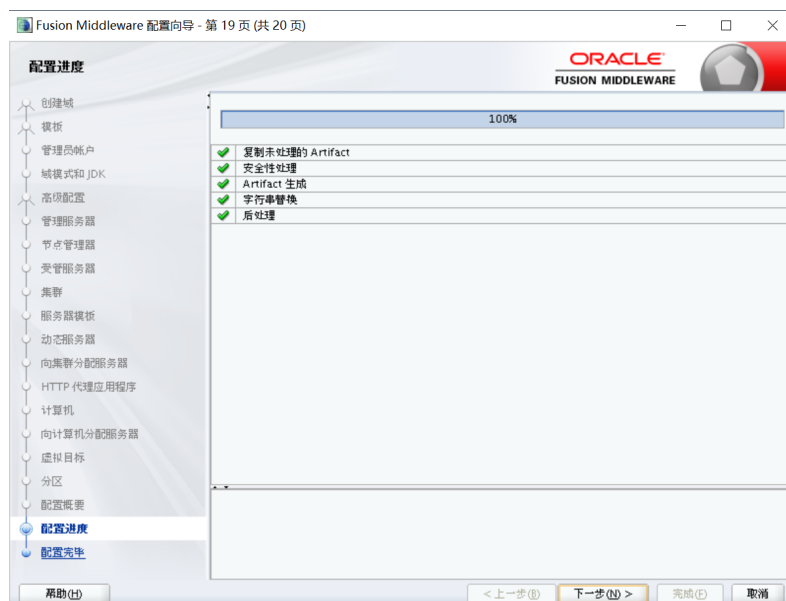


图 3-210 创建 Weblogic 域 3



创建无误后，单击“完成”按钮。

集群中辅节点操作

本例是weblogic双机集群，以上操作均在主节点中完成。

集群中的所有 Server 必须位于同一网段，并且必须是 IP 广播(UDP)可到达的；

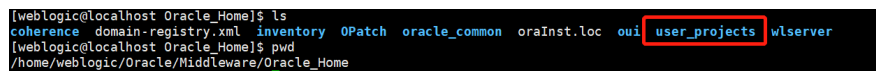
集群中的所有 Server 必须使用相同的版本，包括 Service Pack；

集群中的 Server 必须使用永久的静态 IP 地址。动态 IP 地址分配不能用于集群环境。如果服务器位于防火墙后面，而客户机位于防火墙外面，那么服务器必须有公共的静态 IP 地址，只有这样，客户端才能访问服务器；

辅节点weblogic安装和主节点相同版本，安装路径相同，过程略。辅节点具体操作如下：

将主节点创建好的user_projects文件夹复制到辅节点机器

图 3-211 复制



3.3.4.7 启动集群服务

一般常用两种启动方式，一种是通过命令行来启动，一种是启动nodemanager和主节点控制台后通过界面来启动，推荐使用后者。

1. 来启动

编辑密码文件

在域目录servers/AdminServer/security、servers/ManagedServer_1/security、servers/ManagedServer_2/security、servers/proxyServer/security文件夹下（如果没有需新建）建立boot.properties文件，内容为：

```
username=weblogic  
password=weblogic1
```

启动代理服务器

```
nohup ./startManagedWebLogic.sh 受管理服务器名称 管理服务器URL  
nohup ./startManagedWebLogic.sh Proxy http://172.21.160.13:7001 &
```

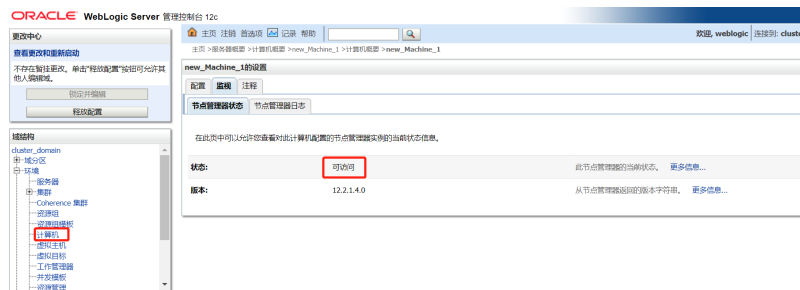
启动受管理服务器ManagedServer_1.

```
nohup ./startManagedWebLogic.sh 受管理服务器名称 管理服务器URL  
nohup ./startManagedWebLogic.sh new_ManagedServer_1 http://172.21.160.13:7001 &
```

如果有多个受管理节点，可以做多个受管理服务器。

2. 启动nodemanager和主节点控制台通过weblogic控制台来启动；主节点操作如下：
 - a. 启动weblogic控制台，和前面介绍启动方式一样，此处略
 - b. 启动Node Manager，启动命令：nohup ./startNodeManager.sh &
 - c. 启动成功之后进入控制台，确保节点管理器状态为可访问

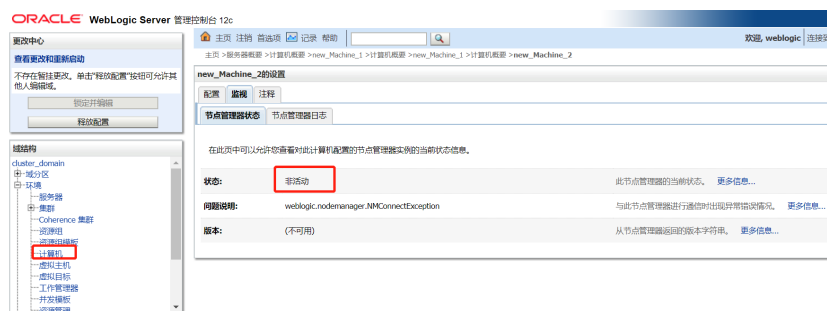
图 3-212 操作 1



说明

后台查看进程，NodeManager确实已经启动成功，但是控制台查看状态为非活动，如下所示：

图 3-213 操作 2



需要修改2个地方：

图 3-214 操作 3



编辑nodemanager.properties文件

图 3-215 编辑 1

```
[root@localhost nodemanager]# pwd
/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager
[root@localhost nodemanager]# ls
nodemanager.domains nodemanager.log.lck nodemanager.process.lck stopnodemanager.log
nodemanager.log nodemanager.process.id nodemanager.properties
[root@localhost nodemanager]# vi nodemanager.properties
```

图 3-216 编辑 2

```
Thu Jul 16 15:06:16 CST 2020
#Node manager properties
#Thu Jul 16 14:46:02 CST 2020
DomainsFile=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager/nodemanager.
s
LogLimit=0
PropertiesVersion=12.2.1.4.0
AuthenticationEnabled=true
NodeManagerHome=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager
JavaHome=/opt/jdk/jdk1.8.0_131
LogLevel=INFO
DomainsFileEnabled=true
ListenAddress=172.21.160.13
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
weblogic.StartScriptName=startWebLogic.sh
SecureListener=false
LogCount=1
QuitEnabled=false
LogAppend=true
weblogic.StopScriptEnabled=false
StateCheckInterval=500
CrashRecoveryEnabled=false
weblogic.StartScriptEnabled=true
LogFiles=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
```

辅节点操作如下：（辅节点只需要启动Node Manager）

启动Node Manager，启动命令：nohup ./startNodeManager.sh &，此处需要修改一个文件

图 3-217 编辑 3

```
[root@localhost nodemanager]# pwd
/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager
[root@localhost nodemanager]# ls
nodemanager.domains nodemanager.log.lck nodemanager.process.lck
nodemanager.log nodemanager.process.id nodemanager.properties
[root@localhost nodemanager]# vi nodemanager.properties
```


图 3-218 编辑 4

```
Thu Jul 16 15:24:31 CST 2020
#Node manager properties
#Thu Jul 16 14:46:02 CST 2020
DomainsFile=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager/nodemanager.do
s
LogLimit=0
PropertiesVersion=12.2.1.4.0
AuthenticationEnabled=true
NodeManagerHome=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager
JavaHome=/opt/jdk/jdk1.8.0_131
LogLevel=INFO
DomainsFileEnabled=true
ListenAddress=172.21.160.47
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
weblogic.StartScriptName=startWebLogic.sh
SecureListener=false
LogCount=1
QuitEnabled=false
LogAppend=true
weblogic.StopScriptEnabled=false
StateCheckInterval=500
CrashRecoveryEnabled=false
weblogic.StartScriptEnabled=true
LogFile=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
```

需要改为这个server的ip

出现下图，5556端口处于监听表示启动成功。

图 3-219 编辑 5

```
ListenPort=5556
LogToStderr=true
weblogic.StartScriptName=startWebLogic.sh
SecureListener=false
LogCount=1
LogAppend=true
weblogic.StopScriptEnabled=false
StateCheckInterval=500
CrashRecoveryEnabled=false
weblogic.StartScriptEnabled=true
LogFile=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
coherence.StartScriptEnabled=false
ListenBacklog=50
NodeManagerHome=/home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain/nodemanager
RestartEnabled=false
isLoggingEnabled=true
weblogic.startup.JavaHome=/opt/jdk/jdk1.8.0_131
weblogic.startup.MW_Home=
coherence.startup.JavaHome=/opt/jdk/jdk1.8.0_131
coherence.startup.MW_Home=

Domain name mappings:

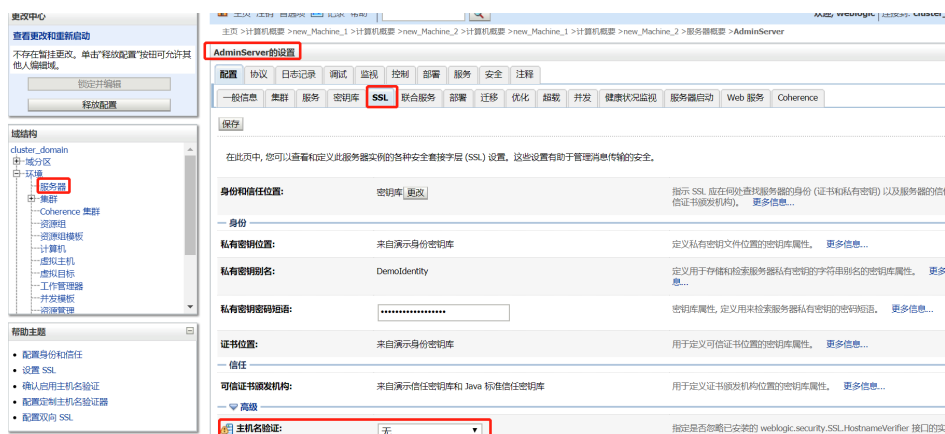
cluster_domain -> /home/weblogic/Oracle/Middleware/Oracle_Home/user_projects/domains/cluster_domain

<Jul 16, 2020 3:26:25 PM CST> <INFO> <12.2.1.4.0>
<Jul 16, 2020 3:26:25 PM CST> <INFO> <Server Implementation Class: weblogic.nodemanager.server.NMServer$ClassicServer.>
<Jul 16, 2020 3:26:25 PM CST> <INFO> <Plain socket listener started on port 5556, host /172.21.160.47>
```

此时，登录到weblogic控制台（http://172.21.160.13:7001/console）

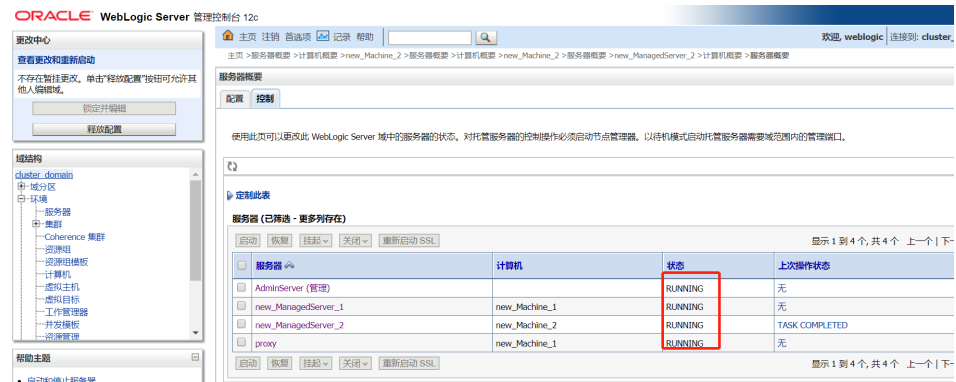
3. 修改AdminServer的设置SSL主机验证为无

图 3-220 修改 AdminServer 的设置



4. 然后在服务器界面，依次启动代理和节点（即主控和受控节点）
切换到控制页面，一次勾选节点服务器和代理，单击“启动”，启动成功后如下
图RUNNING状态，启动时间视机器配置而定，此时整个集群服务启动完毕。

图 3-221 启动代理和节点



在集群中部署 ABI 应用

应用部署：进入Console控制台（<http://172.21.160.13:7001/console>）。

图 3-222 登录



进入console控制台后，在左边域结构树中单击“环境”前的“+”，

图 3-223 环境



在展开的节点中单击“服务器”，在右边的列表中可以看到各服务器是否正常启动。

图 3-224 服务器



如果正常启动则单击“部署”

图 3-225 部署



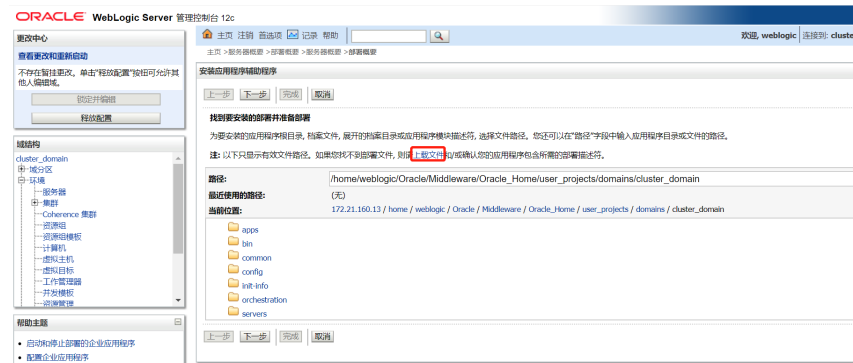
再单击“安装”按钮，选择要部署的WEB应用程序

图 3-226 选择



单击“上传文件”

图 3-227 上传文件



单击“选择文件”

图 3-228 选择文件

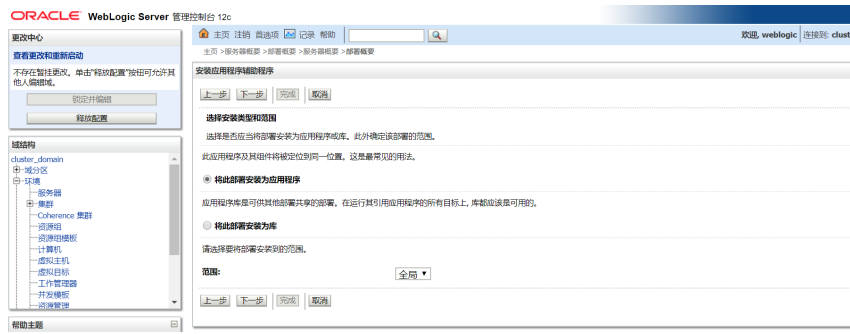


选择好之后等待上传到服务器，单击“下一步”

图 3-229 下一步 1

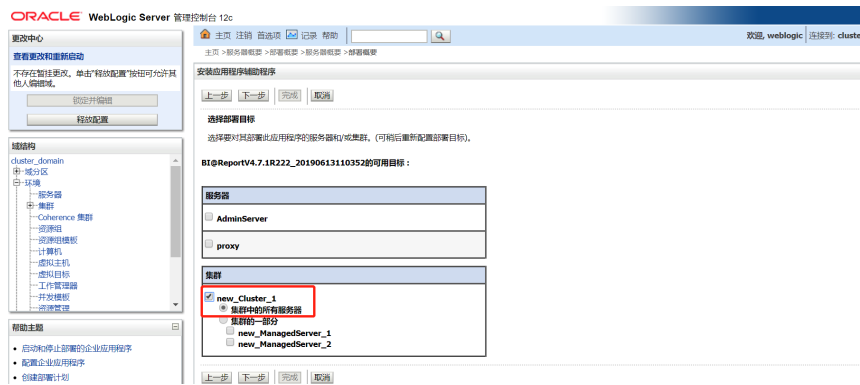


图 3-230 下一步 2



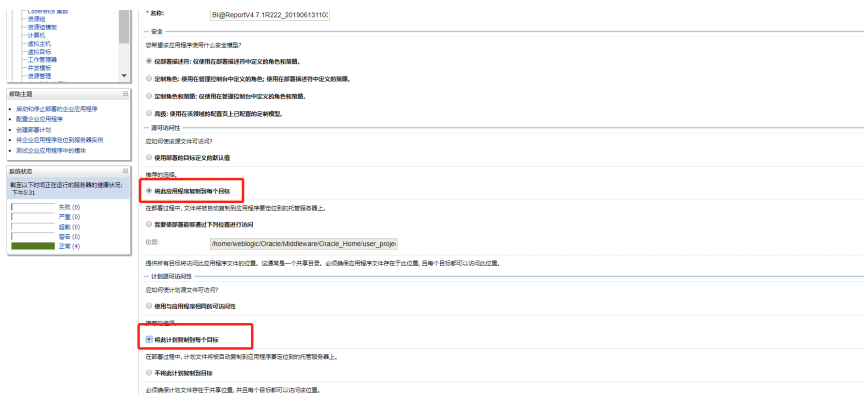
选择“群集中的所有服务器”，之后单击“下一步”

图 3-231 下一步 3



选择“将此目标复制到每一个目标”，再单击“下一步”

图 3-232 下一步 4



再单击“完成”。（部署s的是生产环境还需要再单击左上角的“激活更改”）。

图 3-233 单击完成 1

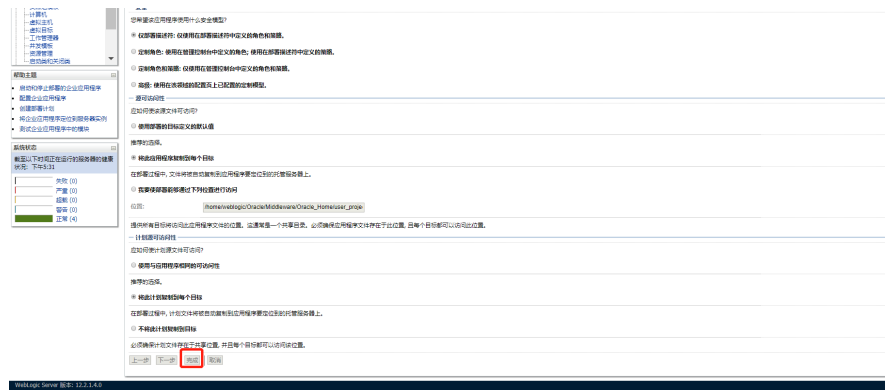
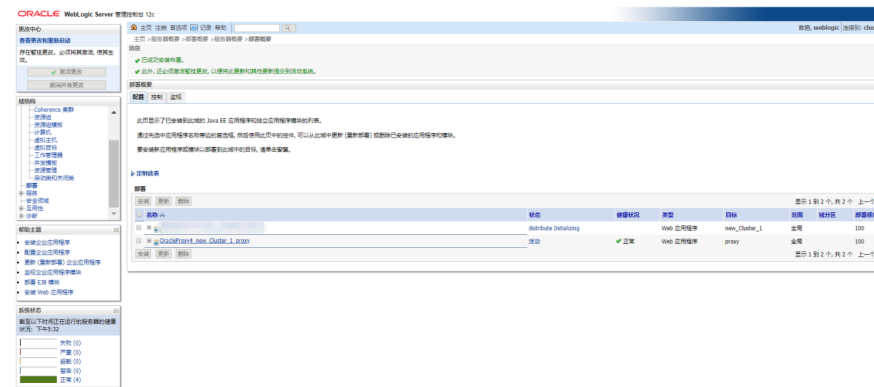


图 3-234 单击完成 2



启动应用程序

启动WEB应用

进入Console控制台（<http://172.21.160.13:7001/console>）。

进入console控制台后，单击“部署”

图 3-235 部署



选中列表中的应用，然后选择“为所有请求提供服务”，再单击“是”。

图 3-236 选择



看到状态是“活动”就表示WEB应用启动完成。

图 3-237 完成



内存调整

修改域目录下\bin下的setDomainEnv.cmd (linux下setDomainEnv.sh) 对应JDK的内存。

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，一般建议在32bit环境下最大内存不要超过2G，在64bit环境下不要超过系统可用内存的70%。

⚠ 注意

- 某些JDK版本限制最大堆内存，不支持分配超过1.5G的内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。
- Weblogic建议最大最小内存分配一样大小。

Session 设置

1. Application.properties

在war包中配置文件web-inf\classes\application.properties，添加如下内容：

```
server.session.timeout=1800
```

此例表示Session将在30分钟后过期

如果不配置，则默认30分钟

2. Weblogic.xml

在war包中找到web-inf\weblogic.xml，设置Weblogic特有部署描述符Weblogic.xml的<session-descriptor>元素的TimeoutSecs属性。这个值以秒为单位：

```
<session-descriptor>  
  <session-param>  
    <param-name>TimeoutSecs</param-name>  
    <param-value>3600</param-value>  
  </session-param>  
</session-descriptor>
```

默认值是3600秒

服务器端压力较大时建议减小此时间。

3.3.5 Websphere 中部署 ABI

3.3.5.1 在 Websphere8.5 中部署

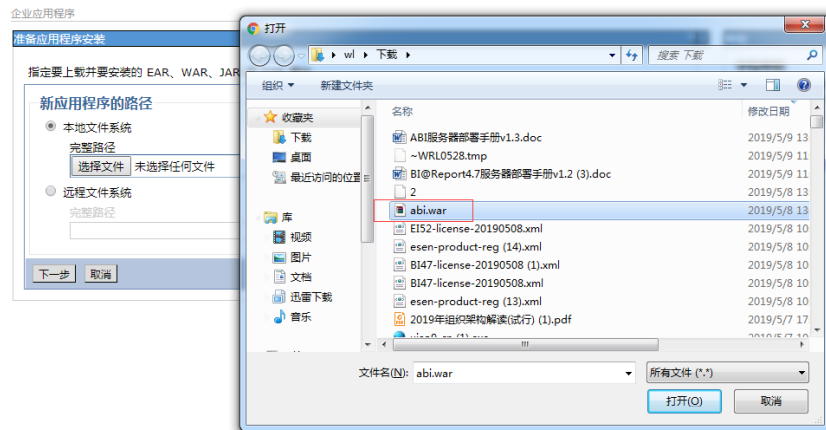
选择“新建应用程序 > 新建企业应用程序”

图 3-238 选择



在如下界面下，选择浏览按钮，在本地找到需要部署的包

图 3-239 部署的包



此处注意，长文件名websphere不支持，需要修改文件名。

选择下一步

图 3-240 下一步 1



选择下一步

图 3-241 下一步 2



继续下一步

图 3-242 下一步 3



继续下一步

图 3-243 下一步 4



继续下一步，指定上下文根为（这里指定为abi）

图 3-244 下一步 5



继续下一步

图 3-245 下一步 6



单击完成按钮

图 3-246 下一步 7



选择保存到主配置, 保存一定要做, 否则会出现问题

保存完成后选择“全部应用程序”, 可看到刚才部署的包还没有启动, 选择名称前的checkbox, 操作为启动选项, 然后单击提交操作按钮,

图 3-247 提交



3.3.5.2 Websphere 常用调配

部署调优配置

Web容器传输链

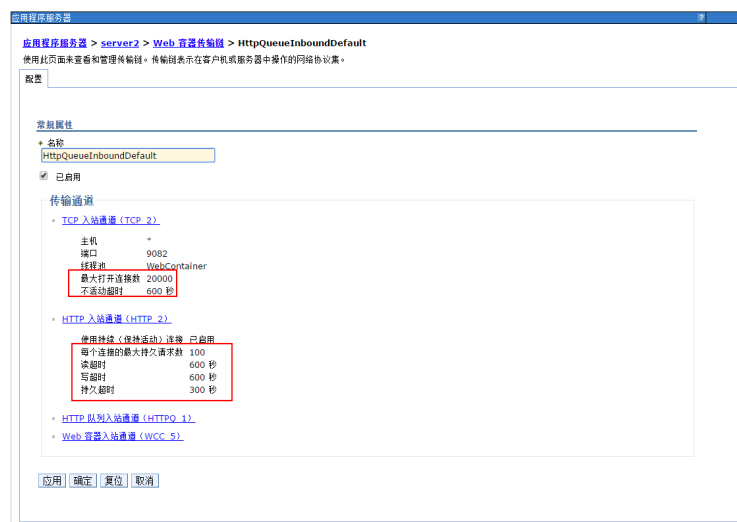
进入WAS控制台，依次进入页面：“应用程序服务器 > server2 > Web 容器传输链”

图 3-248 部署调优配置 1



修改以上各资源的相关参数，可以默认参数基础上，将连接数和超时时间扩大10倍：

图 3-249 部署调优配置 2



线程池最大大小扩大10倍

图 3-250 部署调优配置 3



定制属性设置

在“服务器 > 服务器类型 > WebSphere Application Server > your_server_name > Web容器设置 > Web容器 > 定制属性”中，单击“新建”，加入一个自定义属性：

名称：com.ibm.ws.webcontainer.invokeFiltersCompatibility

值：true

图 3-251 定制属性设置



内存设置

图 3-252 内存设置 1



WebSphere默认内存最大内存为256M，在大用户量情况下容易出现内存不够的情况，所以需要进行调整。在如图所示地方进行调整，直接输入数字即可。

图 3-253 内存设置 2



在这块配置内存参数（数值不能低于此值）。jdk1.8开始取消了PermGen，取而代之的是Metaspace-Xmx1024m -Xms1024m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，在64bit环境下建议分配8G-16G内存。

注意

某些JDK版本或操作系统限制最大堆内存，不支持分配超过较大内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

线程池调整

该参数在管理控制台里的“服务器→应用程序服务器→server1→线程池”的“WebContainer”中进行设置，不要勾选“允许线程分配超过最大线程大小”。此处建议根据压力测试实际情况进行调整。

图 3-254 线程池调整 1



“服务器→应用程序服务器→server1→Web容器设置→Web容器”里勾选“启用servlet高速缓存”，提高访问速度。

图 3-255 线程池调整 2



服务器端 SSL 配置

Websphere WAS中间件产品在安装部署时已经自带SSL访问方式。见下图。

图 3-256 服务器端 SSL 配置

<input type="checkbox"/>	WC_adminhost	*	9060
<input type="checkbox"/>	WC_adminhost_secure	*	9043
<input type="checkbox"/>	WC_defaulthost	*	9080
<input type="checkbox"/>	WC_defaulthost_secure	*	9443

WC adminhost 为控制台端口

WC_adminhost_secure 为https访问控制台端口

WC defaulthost 应用访问端口

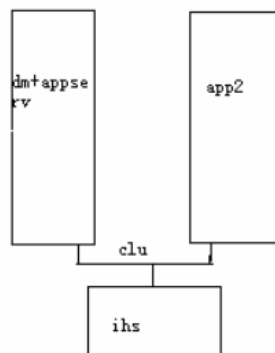
WC defaulthost_secure 为https应用访问端口

3.3.5.3 Websphere 集群部署

3.3.5.3.1 实例说明

本例中的部署如下所示：

图 3-257 部署示意



Dm 和一个应用安装在主机WEBSPHEREDM，一个应用单独安装主机名为 WEBSPHERECLU2，组成一个集群，然后通过一个IHS 来完成负载均衡,主机名为 WEBSPHEREIHS,今后可以对IHS扩容成一个集群来完成IHS的高可用。

需提供两台服务器，必须在每个服务器中的网络中添加其他两台主机，可修改hosts文件如下所示：

图 3-258 hosts 文件



192.168.84.92 是DM+app1机器

192.168.84.110是app2的机器

备注:

1. 选一台WAS作为主，其它几台作为辅。
su -c 'date -s 时:分:秒'
2. 修改时间，几台服务器的时间必须同步，时间相差尽量不要超过2分钟。
(date -s 加时间，语言转换命令: export LANG=)
3. 修改hosts文件，命令: vi/opt/soft i键编辑 WQ 保存。
4. 修改好之后 ping 主机名 看是否通。

3.3.5.3.2 创建概要文件

概要文件是WEBSHERE的重要概念，相当于一个独立的配置文件，每个应用都必须有一个单独的概要文件。

先在AppServer目录下创建profiles文件夹

创建概要文件的方法通过命令行:

图 3-259 命令

```
[root@wasdm bin]# pwd  
/opt/IBM/WebSphere/AppServer/bin
```

```
./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/dmgr -  
profilePath /opt/IBM/WebSphere/AppServer/profiles/Dmgr01 -nodeName Dmgr01Node -cellName  
Dmgr01NodeCell -enableAdminSecurity true -adminUserName wasadmin -adminPassword wasadmin -  
profileName Dmgr01 -hostName wasdm
```

注意

-hostName wasdm 此处务必写主机名,不能用localhost

成功后提示:

图 3-260 提示

```
[root@wasdm bin]# ./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/dmgr -profilePath /opt/IBM/WebSphere/AppServer/profiles/Dmgr01 -nodeName Dmgr01Node -cellName Dmgr01NodeCell -enableAdminSecurity true -adminUserName wasadmin -adminPassword wasadmin -profileName Dmgr01 -hostName wasdm
INSTCONFSUCCESS: 成功: 概要文件 Dmgr01 现在存在。有关此概要文件的更多信息, 请参阅 /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/logs/AboutThisProfile.txt。
```

3.3.5.3.3 服务启动

启动DM，进入下图中的bin目录下，输入命令 `./startmanager.sh` 启动DM服务；

图 3-261 服务启动 1

```
[root@wasdm bin]# pwd
/opt/IBM/WebSphere/AppServer/bin
```

创建单独的应用服务器。

仍然使用上面的方法，通过命令行创建，如下：

图 3-262 服务启动 2

```
[root@wasdm bin]# pwd
/opt/IBM/WebSphere/AppServer/bin
```

```
./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/managed -profileName AppSrv01 -profilePath /opt/IBM/WebSphere/AppServer/profiles/AppSrv01 -nodeName Default01Node -hostName wasdm -cellName Default01Cell -dmgrHost 192.168.84.92 -dmgrPort 8879 -dmgrAdminPassword wasadmin -dmgrAdminUserName wasadmin
```

成功后提示：

图 3-263 服务启动 3

```
[root@wasdm bin]# ./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/managed -profileName AppSrv01 -profilePath /opt/IBM/WebSphere/AppServer/profiles/AppSrv01 -nodeName Default01Node -hostName wasdm -cellName Default01Cell -dmgrHost 192.168.84.92 -dmgrPort 8879 -dmgrAdminPassword wasadmin -dmgrAdminUserName wasadmin
INSTCONFSUCCESS: 成功: 概要文件 AppSrv01 现在存在。有关此概要文件的更多信息, 请参阅 /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/AboutThisProfile.txt。
```

以下安装在WEBSPHERECLU2上安装(192.168.84.110)。

在主机WEBSPHERECLU2仅需要安装应用

重复步骤(1),(2)

创建NODE,命令行如下：

```
./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/managed -profileName AppSrv04 -profilePath /opt/IBM/WebSphere/AppServer/profiles/AppSrv04 -nodeName Default04Node -hostName qbwebclu4 -cellName Default04Cell -dmgrHost 10.148.63.104 -dmgrPort 8879 -dmgrAdminPassword wasadmin -dmgrAdminUserName wasadmin
```

DM控制台登录页面可看到两个节点，确保状态都是已同步

图 3-264 登录



3.3.5.3.4 创建集群

在DM控制中选择集群,然后新建

图 3-265 创建集群 1



添加成员1

图 3-266 创建集群 2

步骤 1: 输入基本集群信息

→ 步骤 2: 创建第一个集群成员

步骤 3: 创建其他集群成员

步骤 4: 总结

创建第一个集群成员

第一个集群成员确定集群成员的服务器设置。服务器配置模板是从第一个成员创建的并存储为集群数据的一部分。其他集群成员都是从此模板中复制的。

* 成员名
app1

选择节点
Default01Node(ND 6.1.0.0)

* 权重
2 (0..20)

生成唯一的 Http 端口

选择第一个集群成员的基础:

- 使用应用程序服务器模板创建成员。
default
- 使用现有应用程序服务器作为模板创建成员。
(无)
- 通过转换现有应用程序服务器创建成员。
(无)
- 无。创建空的集群。

添加成员2,单击下图添加成员按钮

图 3-267 创建集群 3

步骤 1: 输入基本集群信息

步骤 2: 创建第一个集群成员

→ 步骤 3: 创建其他集群成员

步骤 4: 总结

创建其他集群成员

输入有关此新集群成员的信息，然后单击添加成员以将此集群成员添加至成员列表。服务器配置模板是从第一个成员创建的并存储为集群数据的一部分。其他集群成员都是从此模板中复制的。

* 成员名
app2

选择节点
Default02Node(ND 6.1.0.0)

* 权重
2 (0..20)

生成唯一的 Http 端口

添加成员

使用“编辑”功能来编辑已包含在此列表中的集群成员的属性。使用“删除”功能从此列表中除去集群成员。不允许您编辑或删除第一个集群成员或已经存在的集群成员。

编辑	删除			
选择	成员名	节点	版本	权重
<input type="checkbox"/>	app1	Default01Node	ND 6.1.0.0	2

注意此处的权重,代表负载的权重比例

图 3-268 创建集群 4

群信息

步骤 2: 创建第一个集群成员

步骤 3: 创建其他集群成员

→ 步骤 4: 总结

操作总结:

选项	值
集群名	testclu
核心组	DefaultCoreGroup
节点组	DefaultNodeGroup
首选本地	true
配置 HTTP 会话内存到内存复制	false
服务器名称	app1
节点	Default01Node(ND 6.1.0.0)
权重	2
克隆模板	default
克隆类型	default
生成唯一的 Http 端口	true
服务器名称	app2
节点	Default02Node(ND 6.1.0.0)
权重	2
克隆模板	default
克隆类型	default
生成唯一的 Http 端口	true

上一步 完成 取消

图 3-269 创建集群 5

服务器集群

服务器集群

使用此页面来更改集群的配置设置。一个服务器集群由一组应用程序服务器组成。如果其中一个成员服务器发生故障，则请求会被路由到集群的其他成员。在 [guided activity](#) 中学习有关此任务的更多内容。指导性活动提供任务步骤列表和关于主题的更多一般信息。

☑ 首选项

新建 删除 启动 停止 波动启动 立即停止

选择	名称	状态
<input type="checkbox"/>	testclu	✖

总数 1

启动群集服务。

3.3.5.3.5 IHS 的部署

注意

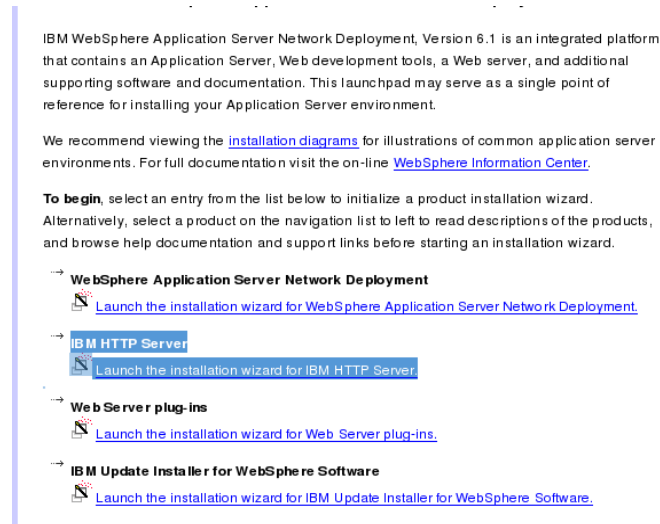
以下操作在WEBSHEREIHS主机上执行。

之前的操作仅仅是部署了一个应有到一个集群上,但是这个时候负载均衡未时实现,所以需要配置HIS来实现负载均衡。

解压C88TLML_WASND_61_LINUX_X86_64_SUPPLEMENT.tar.gz

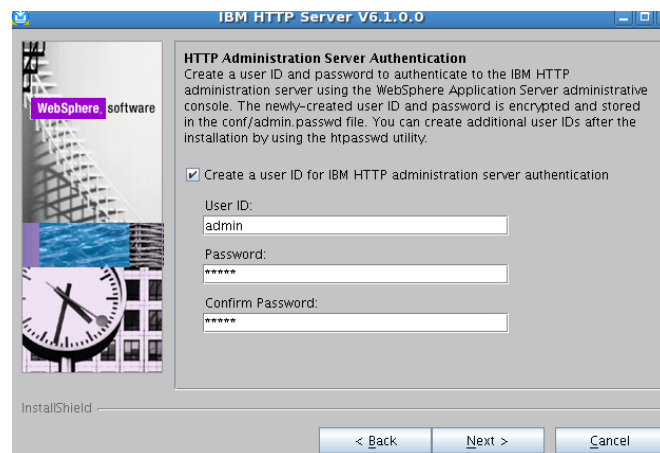
解压后执行./launchpad.sh

图 3-270 IHS 的部署 1



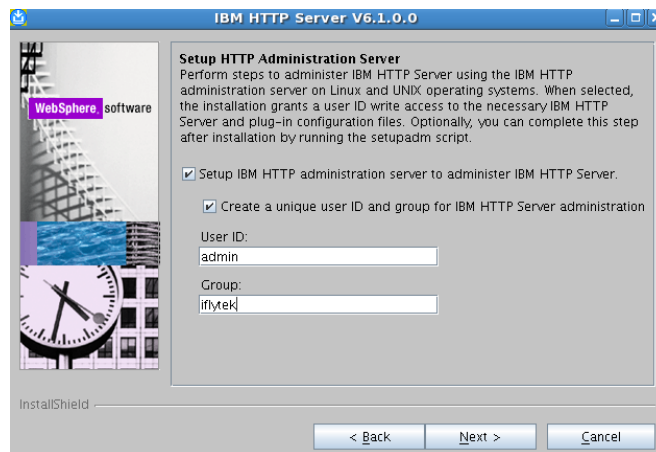
选择安装http Server

图 3-271 IHS 的部署 2



安装过程中询问是否创建IHS的组、用户，得创建上。按照提示创建即可，注意管理用户和密码

图 3-272 IHS 的部署 3



按照过程中都是默认选项.

安装后可以使用如下命令启动和停止IHS

目录: /websphereND/gzwlyw/IBM/HTTPServer/bin

./apachectl start stop IHS的启动和停止

3.3.5.3.6 安装插件

依然在下面界面中选择

图 3-273 安装插件 1

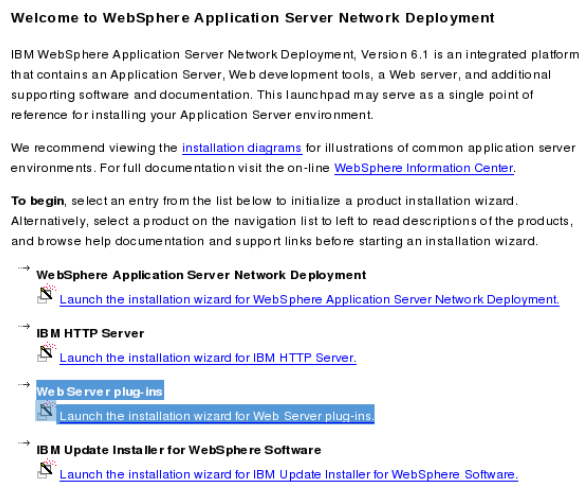


图 3-274 安装插件 2

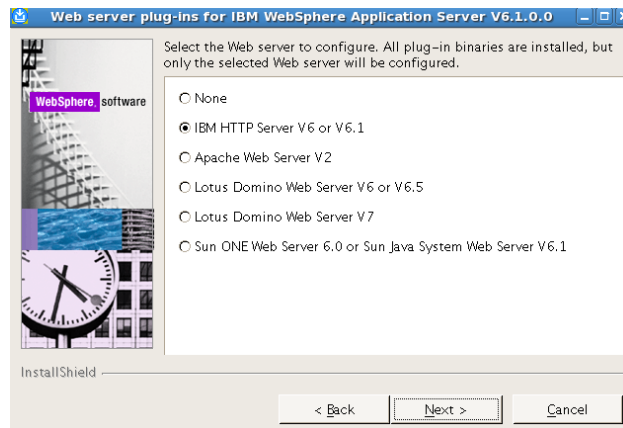


图 3-275 安装插件 3

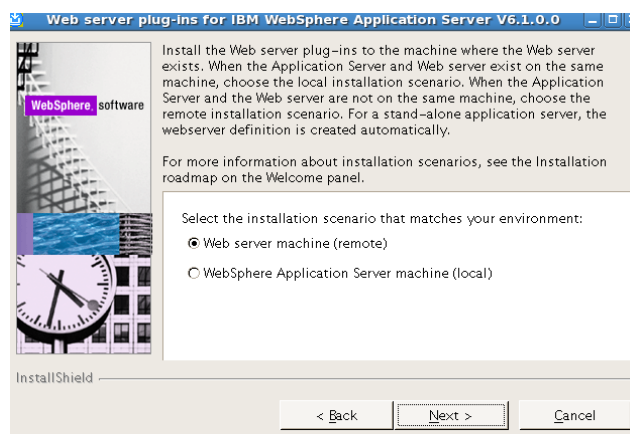
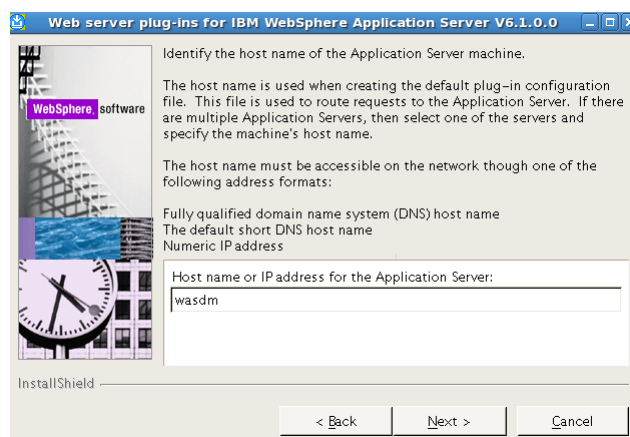


图 3-276 安装插件 4



其它默认安装.

安装后把/opt/IBM/HTTPServer/Plugins1/bin下面的configurewebserver1.sh复制到/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/bin下，然后执行./configurewebserver1.sh -user wasadmin -password wasadmin.

图 3-277 安装插件 5



图 3-278 安装插件 6



3.3.5.3.7 应用部署与启动应用程序

应用部署

需注意事项，在部署的时候，需选择集群和webserver1两个，选中checkbox后应用，如下图

图 3-279 应用部署 1



之后的操作都是默认。

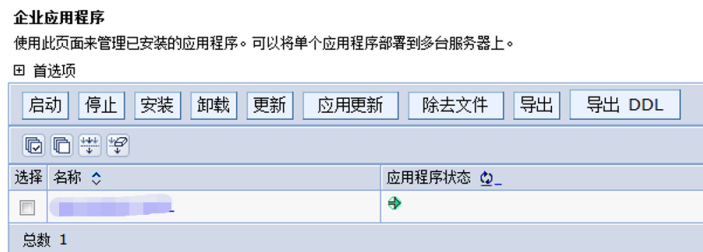
设置虚拟主机中的default_host的主机名，添加一个9081端口

图 3-280 应用部署 2



启动应用程序

图 3-281 启动应用程序



全部完成了，集群访问地址：<http://192.168.84.92/jsec-zdsjyj>;

was会根据权重自动负载到92:9080和110:9080两个节点上，同时满足实现故障转移。

3.3.5.4 启动命令

启动步骤如下

1. 启动DM，进入bin目录下，输入命令 startmanager 启动DM服务
2. 启动ND01，进入app01的bin目录下，输入命令startnode 启动ND服务
3. 启动ND02，进入app02的bin目录下，输入命令startnode 启动ND服务

3.3.6 金蝶 AAS V9.0 中部署 ABI

3.3.6.1 部署环境配置

本章节主要讲述在linux环境下如何安装及配置金蝶中间件。

安装应用服务器

确保所登录用户对AAS-V9.0.zip有可读写权限(在终端中用命令 `ls -l AAS-V9.0.zip`查看)，可通过以下命令将AAS-V9.0.zip解压缩到安装目录：

可参考以下命令(具体以实际为准)：

`unzip -d /home/AASV9/ /home/AAS-V9.0.zip`其中：

`/home/AASV9`：表示指定文件解压后的（安装）目录

`/home/AAS-V9.0.zip`：表示金蝶9.0上传到服务器上的安装包

启动服务器

进入`/home/AASV9/domains/mydomain/bin`目录，输入`startapusic`命令并执行即可。

可参考以下命令(具体以实际为准)：

```
cd /home/AASV9/domains/mydomain/bin
```

前台启动

```
sh ./startapusic
```

后台启动

```
nohup ./startapusic &
```

后台启动可以`tail -f nohup.out` 查看输出的日志

查看版本号

```
./startapusic -v
```

首次启动服务器需要配置管理端密码，输入默认密码可以自定义但是长度要超过8位，包含大写小写字母，数字和特殊符号，另外输入密码时是不显示的，所以不要以为你没输入进去。

图 3-282 启动服务器 1

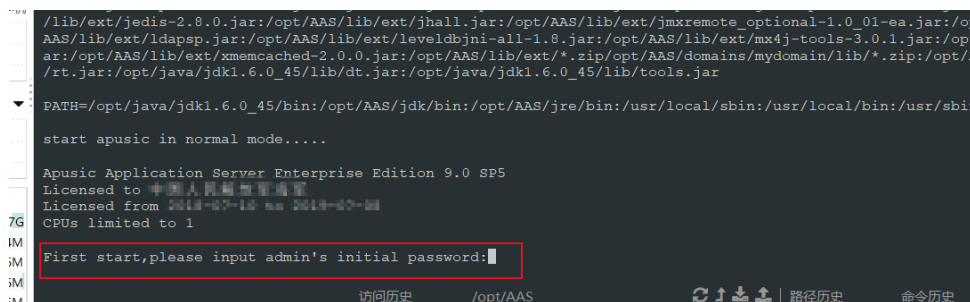
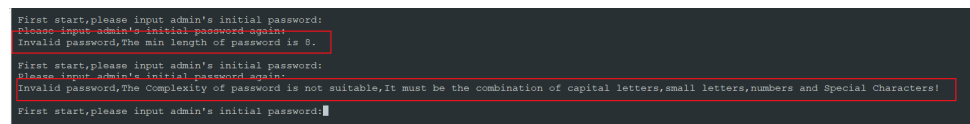


图 3-283 启动服务器 2



启动过程，看到“服务器就绪”字样，启动成功。

图 3-284 启动服务器 3

```
2020-06-29 15:34:50 信息 [apusic.service.JCA-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.EJB-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.EJBTimer-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.MUXEndpoint-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.ACPEndpoint-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.Web-Thread 1] apusic.web.io.type = nio
2020-06-29 15:34:50 信息 [apusic.service.Web-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.HttpSessionStore-Thread 1] 成功启动
2020-06-29 15:34:50 信息 [apusic.service.HttpSession-Thread 1] 成功启动
2020-06-29 15:34:51 信息 [apusic.web.webtool.war./admin-Thread 1] Initializing Spring root WebApplicationContext
2020-06-29 15:34:52 信息 [apusic.web.webtool.war./admin-Thread 1] Context Root [/admin]
2020-06-29 15:34:52 信息 [apusic.application.webtool.war-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.ejb.mejb.jar.MEJB-Thread 1] EJB home 'ejb/mgmt/MEJB' 已注册。
2020-06-29 15:34:53 信息 [apusic.application.mejb.jar-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.web.default./-Thread 1] Context Root [/]
2020-06-29 15:34:53 信息 [apusic.application.default-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.service.J2EEDeployer-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.service.RMIConnector-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.service.admin-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.service.ClassLoaderViewer-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.bridge.JNDIResourceService-Thread 1] 成功启动
2020-06-29 15:34:53 信息 [apusic.server.Main-Thread 1] 服务器就绪。
```

管理控制台

已经正确的启动了Apusic应用服务器后，在浏览器中输入URL：`http://hostname:6888/admin`

hostname：指安装Apusic应用服务器软件的计算机的IP地址或主机名；

6888：是Apusic应用服务器默认的http端口。

访问成功后出现如下图所示的页面：

图 3-285 管理控制台



Apusic 管理控制台的初始用户名为 admin，密码为上面配置管理端密码。

停止服务器

`jps -lv` 查找对应进程号，使用`kill -9 pid`命令停止进程。

图 3-286 停止服务器

```
[root@localhost bin]# jps -lv
353 com.apusic.server.Main -Djava.endorsed.dirs=/home/AASV9/lib/endorsed -Dcom.apusic.domain.home=/home/AASV9/domains/mydo
bin -Djava.net.preferIPv4Stack=true -Djava.security.egd=file:/dev/./urandom -Xms1024m -Xmx2048m -XX:MaxPermSize=512m
1822 sun.tools.jps.Jps -Denv.class.path=./home/jdk1.8.0_201/lib/dt.jar:/home/jdk1.8.0_201/lib/tools.jar -Dapplication.home=
/home/jdk1.8.0_201 -Xms8m
```

如果是前台启动，可以通过Ctrl+c进行停止服务

3.3.6.2 部署 ABI

注意

如果金蝶9.0上正常部署ABI5.3.1，需要做以下处理。增加一个web.xml文件在war包的WEB-INF目录下。

如果金蝶9.0上部署启动报错，需要删除WEB-INF\lib下的eweb-1.4.1-SNAPSHOT.jar中的META-INF\services下的javax.servlet.ServletContainerInitializer文件后重启。

- 第一种方式：进入控制台管理界面，单击右边“应用管理”，单击“部署应用”

图 3-287 部署 ABI1



图 3-288 部署 ABI2



在部署应用对话框中，输入应用名称，选择“本地应用包”，单击“选择文件”按钮，在本地找到ABI包，单击完成。等待上传和部署，直到部署应用对话框的消失。

至此，部署操作就完成了。

- 第二种方式：将war包放入applications目录下并启动

图 3-289 启动

```
[root@localhost applications]# pwd
/home/AASV9/domains/mydomain/applications
[root@localhost applications]# ls
abihyjw.war  default
```

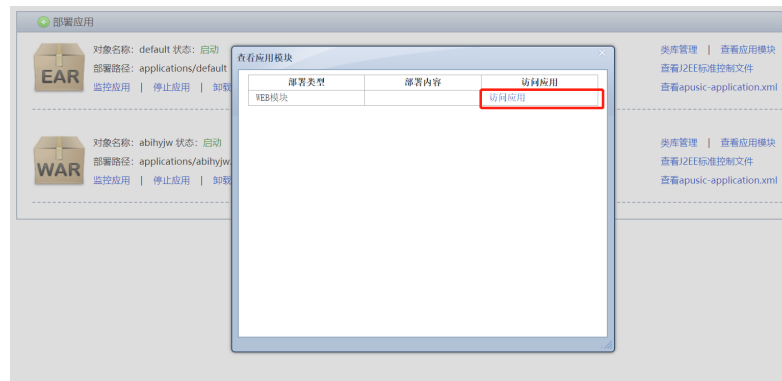
访问测试：找到部署的应用，单击右边的查看应用模块，

图 3-290 查看应用模块



单击访问应用

图 3-291 访问应用



3.3.7 中创 InforSuite V10 中部署 ABI

部署环境配置

安装前准备：安装JDK1.8，具体参照第三章中1.1和2.1步骤

1. 开始安装

支持图形界面和命令行安装，在此介绍命令行安装：

使用root用户登录，进入安装包文件所在的路径，先使用命令：`chmod +x AS_V10.bin`，为安装包文件增加可执行权限。

使用命令执行文件：`./AS_V10.bin -i console`，安装过程简单，此处省略。

2. 启动和停止

安装成功后进入安装目录，如AppServer/as/bin，通过startserv来启动服务；服务启动后,通过命令sh stopserv或./stopserv来停止服务。

3. 管理工具

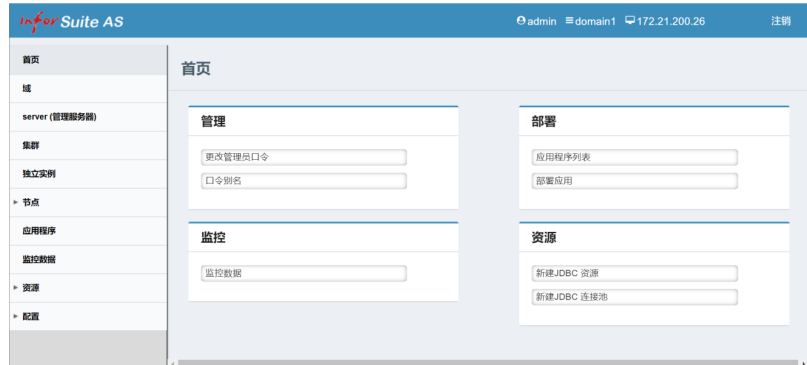
应用服务器初次安装启动后，管理工具默认没有用户名密码，本地访问可在浏览器输入URL 地址，如http://localhost:port进入V10管理工具，其中localhost 为应用服务器的本地地址，port 为管理工具监听端口，默认值为8060。

InforSuite AS支持远程访问管理工具。V10产品安装后，如果不是通过本机访问，需要使用https方式远程访问管理工具，具体的操作步骤如下：需设置登录用户名和密码并开启远程管理功能，可通过管理工具或命令行工具实现。

部署 ABI

进入管理工具，界面如下图所示：

图 3-292 进入管理工具



单击左侧“应用程序”，进入应用部署界面，单击Deploy，进入下一界面

图 3-293 应用程序 1

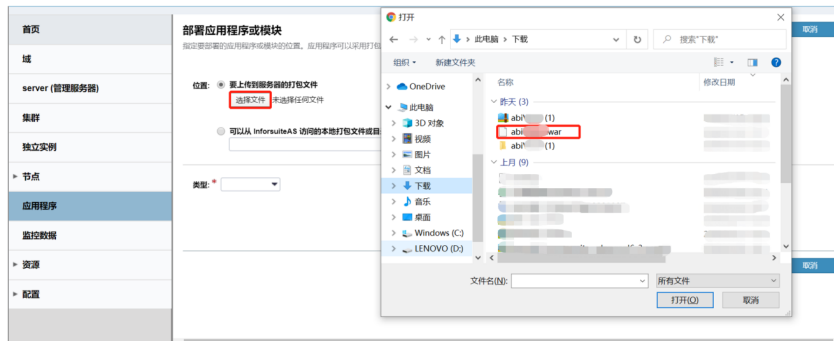


图 3-294 应用程序 2



单击“选择文件”，选择所要部署的产品

图 3-295 选择文件



单击“确定”，开始产品部署

图 3-296 产品部署



成功部署，单击访问

图 3-297 访问



单击访问地址即可进入，非本机访问时需要将AS换成ip地址，如：<http://172.21.200.26:8085/abi>

3.3.8 Tongweb V6.1 中部署 ABI

部署环境配置

1. 安装前准备

安装JDK1.8或以上版本，具体参照第三章中1.1和2.1步骤

2. 开始安装

使用root用户登录，进入安装包文件所在的路径，先使用命令：`chmod +x Install_TW6.1.5.8_Enterprise_Linux.bin`，为安装包文件增加可执行权限。

使用命令执行文件：`./ Install_TW6.1.5.8_Enterprise_Linux.bin`，安装过程简单，此处省略。

3. 启动和停止

安装成功后进入安装目录，如/data/TongWeb6.1/bin/，通过./startserver.sh来启动服务

服务启动后,通过命令./stopserver.sh来停止服务

4. 管理控制台

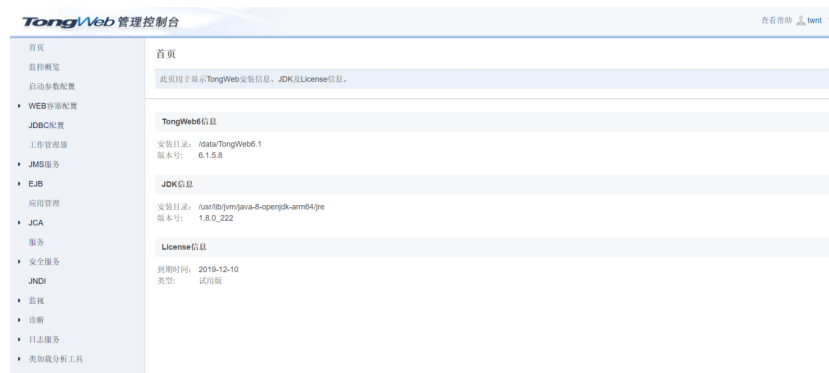
已经正确启动了Tongweb应用服务器后，在浏览器中输入URL “`http://hostname:9060/console`”

hostname：指安装Tongweb应用服务器软件的计算机的IP地址或主机名
9060：是Tongweb应用服务器默认的http端口

部署 ABI

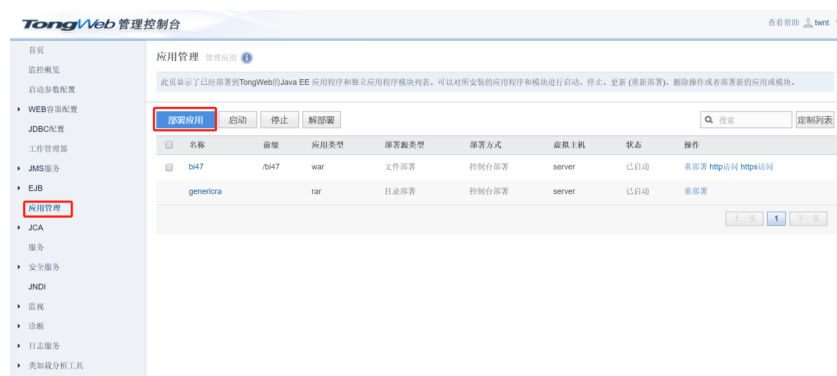
进入管理控制台，界面如下图：

图 3-298 部署 ABI1



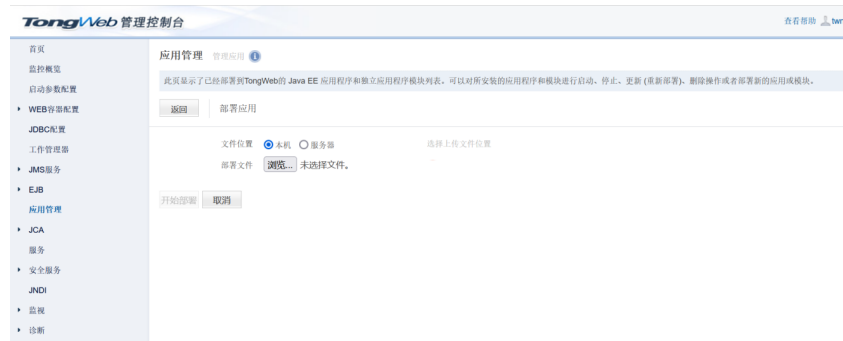
单击“应用管理”，进入应用部署界面，单击“部署应用”，进入下一界面

图 3-299 部署 ABI2



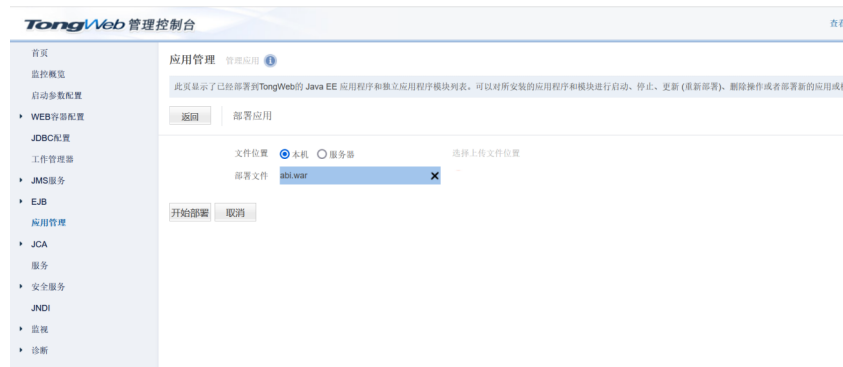
单击“浏览”，选择所要部署的产品

图 3-300 部署 ABI3



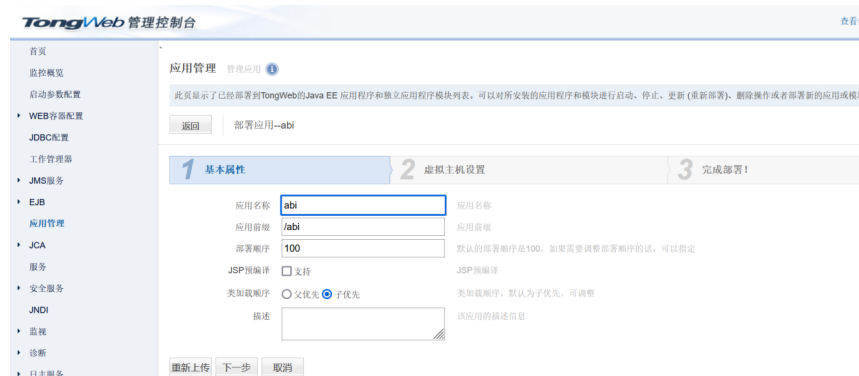
上传完毕后，单击“开始部署”

图 3-301 部署 ABI4



单击“下一步”

图 3-302 部署 ABI5



单击“下一步”

图 3-303 部署 ABI6



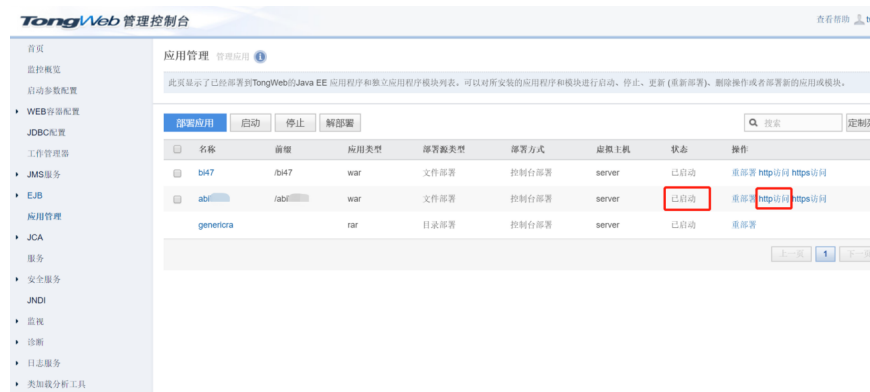
单击“完成”等待部署完毕

图 3-304 部署 ABI7



成功部署，状态为已启动，单击“http访问”即可进入产品访问地址

图 3-305 部署 ABI8



3.3.9 华宇 tas2.8.5 中部署 ABI

下面的tas3.0.0其实就是tas2.8.5

部署环境配置

1. 安装前准备
安装JDK1.8或以上版本，具体参照第三章中1.1和2.1步骤
2. 开始安装
使用root用户登录，进入安装包文件所在的路径，执行命令：`./java -jar tas-installer-3.0.0.jar -console`，安装过程简单，此处省略。
3. 启动和停止
安装成功后进入安装目录，如/data/tas3.0/bin，通过`./ StartTAS.sh`来启动服务；服务启动后,通过命令`./ StopTAS.sh`来停止服务。
4. 管理控制台
已经正确启动了Tas应用服务器后，在浏览器中输入URL “`http://172.21.200.26:8081/tas-console`”
172.21.200.26：指安装Tas应用服务器软件的计算机的IP地址
8081：是Tas应用服务器默认的http端口

部署 ABI

进入管理控制台，界面如下图：

图 3-306 部署 ABI1



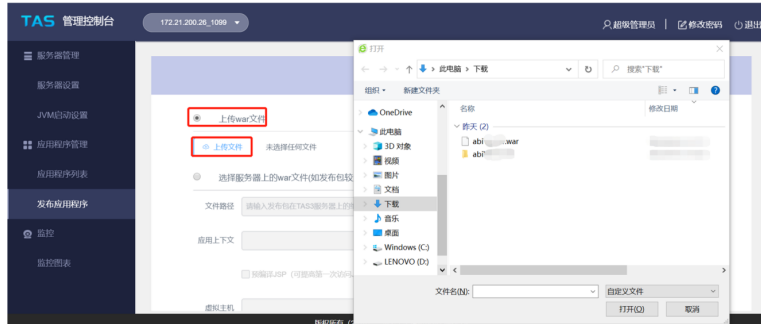
单击左侧的“发布应用程序”

图 3-307 部署 ABI2



单击“上传文件”，选择需要上传的文件，说明：文件小于200M时，选择上传war文件

图 3-308 部署 ABI3



文件大于200M时，选择服务器上的war文件

图 3-309 部署 ABI4



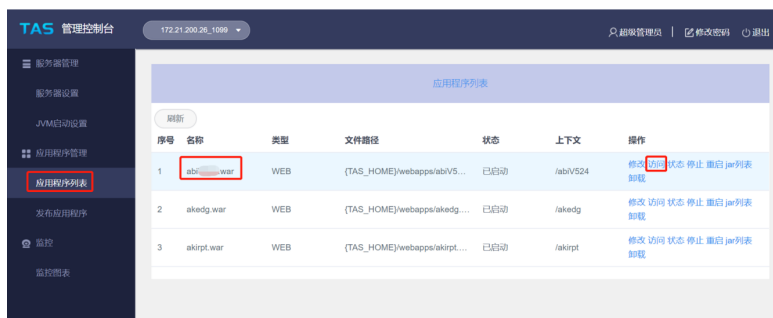
单击“确定”，开始部署

图 3-310 部署 ABI5



成功部署之后，应用程序列表中会出现刚刚部署的应用程序，单击访问即可进入应用访问地址

图 3-311 部署 ABI6



3.3.10 Docker 上部署 ABI

1. 制作Dockerfile

Dockerfile是一个包含用于组合映像的命令的文本文件。可以使用在命令行中调用任何命令。 Docker通过读取Dockerfile中的指令自动生成映像。可以参照以下方式制作Dockerfile。

```
FROM centos:7
ENV JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF8 -Xmx2048m -Xms256m -
XX:MaxMetaspaceSize=512m -XX:MetaspaceSize=256m -Duser.timezone=Asia/Shanghai
-Desen.abi.workdir=/usr/local/workdir -Djava.awt.headless=true"
ENV JAVA_HOME /usr/local/java
ENV JRE_HOME /usr/local/java/jre
ENV PATH /usr/local/java/bin:$PATH
WORKDIR /usr/local/
ADD jdk-8u131-linux-x64.tar.gz /usr/local
ADD apache-tomcat-7.0.76.tar.gz /usr/local
RUN mv jdk1.8.0_131 java && mv apache-tomcat-7.0.76 tomcat
COPY abi.war /usr/local/tomcat/webapps
EXPOSE 6060
ENTRYPOINT ["/usr/local/tomcat/bin/catalina.sh","run"]
```

📖 说明

jdk-8u131-linux-x64.tar.gz、apache-tomcat-7.0.76.tar.gz、abi.war、Dockerfile这几个均放在同一路径下。

2. 构建镜像

- a. 使用当前目录的 Dockerfile 创建镜像

```
docker build -t abi:v1 .
-t: 镜像的名字及标签
```

- b. 通过 -f Dockerfile 文件的位置

```
docker build -f /opt/Dockerfile -t abi:v1 .
```

上下文路径，是指 docker 在构建镜像，有时候想要使用到本机的文件（比如复制），docker build 命令得知这个路径后，会将路径下的所有内容打包。

```
Successfully built 5d113977d755
Successfully tagged abi:v1
```

构建过程最后出现上面显示，即代表镜像构建成功。

3. 启动容器

```
docker run -d -p 6060:8080 -v /opt/workdir:/usr/local/workdir abi:v1
-d: 服务形式启动
```

- p: 端口映射, 宿主机6060端口: docker容器8080端口, 6060可随意修改成未被占用的端口
- v: 工作目录路径, 宿主机/opt/workdir: 容器/usr/local/workdir
- 4. 访问服务器
访问地址: http://ip:6060/abi, ip为宿主机ip地址, 6060为宿主机映射的端口, abi为war包名称, 根据实际修改。
- 5. 工作目录及注册说明
制作Dockerfile时, 环境变量中的-Desen.abi.workdir=/usr/local/workdir是用来设置容器内部的产品工作目录, 启动时再把工作目录映射到宿主机上。
注册时通过注册界面导入注册申请文件, 不要直接把注册申请文件放在宿主机工作目录下。
- 6. 相关问题
报错无法获取服务器硬件信息, 此问题是由于产品在注册时需要获取ip及mac地址, 但是容器内部没有相关命令, 须在构建镜像时加上一行RUN yum install -y net-tools dmidecode

3.3.11 K8S 上部署 ABI

说明: 采用K8S部署abi531时, pod数只能是1个。

有管理平台

说明

此处以华宇容器云平台作为部署工具, 华宇容器云平台是为在生产环境中管理Docker和K8s设计的全栈化容器部署与管理平台。它以Docker及K8s为底层, 以应用为发布单元的企业级容器云平台。

采用开源的rancher管理平台, 部署步骤基本一致。

1. 准备工作

安装nfs

安装参考网址 https://blog.csdn.net/sinat_41075146/article/details/80800812

作用: 允许网络中的计算机之间通过网络共享资源。

说明: 需要搭建服务端和客户端, 服务端nfs共享目录设置为产品工作目录。

镜像准备

提前把所需要的镜像推送到镜像仓库。

2. 部署

添加项目和命名空间: 选择集群, 单击项目/命名空间, 单击添加项目, 输入项目名称, 单击创建, 完成后找到该项目, 单击添加命名空间, 输入名称, 单击创建。

图 3-312 部署 1

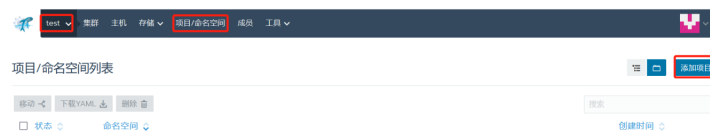


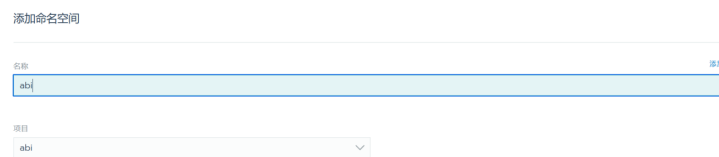
图 3-313 部署 2



图 3-314 部署 3



图 3-315 部署 4



添加PV和PVC 添加PV

图 3-316 添加 PV1



图 3-317 添加 PV2

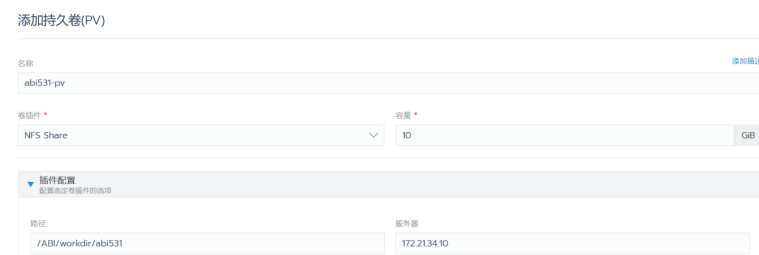


图 3-318 添加 PV3



添加PVC

图 3-319 添加 PVC1

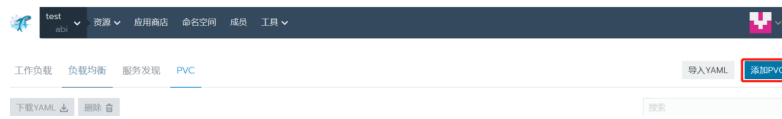


图 3-320 添加 PVC2



部署服务

图 3-321 添加 PVC3



图 3-322 添加 PVC4



无管理平台

说明

无管理平台，即采用yaml文件，用k8s命令部署

1. 准备工作

安装nfs

安装参考网址 https://blog.csdn.net/sinat_41075146/article/details/80800812

作用：允许网络中的计算机之间通过网络共享资源。

说明：需要搭建服务端和客户端，服务端nfs共享目录设置为产品工作目录。

镜像准备

提前把所需要的镜像推送到镜像仓库。

2. 部署

总共4个yaml文件，用k8s命令部署即可，如kubectl apply -f abi-pv.yaml
abi-pv.yaml abi-pvc.yaml abi531-deployment.yaml abi531-service.yaml

图 3-323 abi-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: abi531-pv
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 10Gi
  nfs:
    path: /ABI/workdir/abi531
    server: 172.21.1.10
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
```

图 3-324 abi-pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: abi531-pvc
  namespace: abi
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: ""
  volumeMode: Filesystem
  volumeName: abi531-pv
```

图 3-325 abi531-deployment.yaml1

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: abi531
  namespace: abi
  labels:
    workload.user.cattle.io/workloadselector: deployment-abi-abi531
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      workload.user.cattle.io/workloadselector: deployment-abi-abi531
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
    type: RollingUpdate
  template:
    metadata:
      labels:
        workload.user.cattle.io/workloadselector: deployment-abi-abi531
    spec:
      containers:
        - image: registry.thunisoft.com:5000/test/abi531:v1
          imagePullPolicy: IfNotPresent
          name: abi531
          ports:
```

图 3-326 abi531-deployment.yaml2

```
- containerPort: 8080
  hostPort: 36060
  name: abi531port
  protocol: TCP
resources:
  limits:
    memory: 3Gi
  requests:
    memory: 512Mi
securityContext:
  allowPrivilegeEscalation: false
  capabilities: {}
  privileged: false
  readOnlyRootFilesystem: false
  runAsNonRoot: false
  stdin: true
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
  tty: true
  volumeMounts:
    - mountPath: /opt/workdir
      name: vol1
dnsPolicy: ClusterFirst
restartPolicy: Always
schedulerName: default-scheduler
securityContext: {}
terminationGracePeriodSeconds: 30
volumes:
- name: vol1
```

图 3-327 abi531-deployment.yaml3

```
persistentVolumeClaim:
  claimName: abi531-pvc
```

图 3-328 abi531-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: abi531
  namespace: abi
spec:
  ports:
  - name: abi531port-abi531
    port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    workload.user.cattle.io/workloadselector: deployment-abi-abi531
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

3.3.12 初始化 ABI

通过以上章节了解报表服务器如何部署在各web应用服务器上后，您可以通过如下步骤完成ABI5.3.1初始化工作。

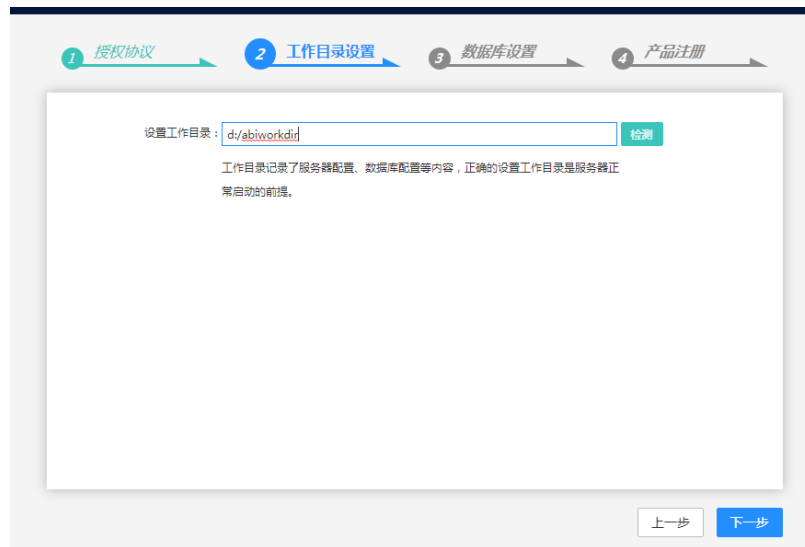
1. 输入登录网址，如：<http://192.168.1.1:8080/ABI>，进入ABI5.3.1服务器配置界面

图 3-329 服务器配置界面



2. 单击“下一步”

图 3-330 下一步

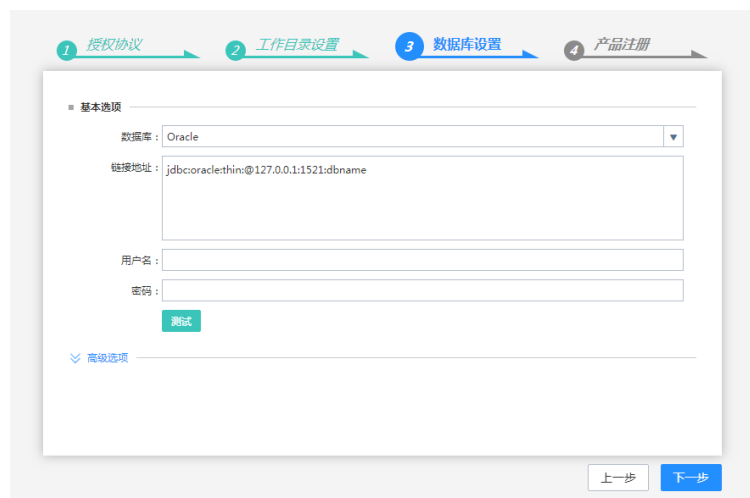


说明

如果在docker中部署时，需要先在本地创建工作目录，以挂载工作目录的方式启动docker

3. 单击“下一步”，设置数据库

图 3-331 设置数据库



此处可以单击测试，测试数据库链接。

4. 单击“下一步”，注册服务器

图 3-332 注册服务器 1

此处，需要填写用户名称，项目名称，联系电话和联系邮箱等，然后生成注册申请，将生成的注册申请文件导出，进行注册码申请，由具有审批权限的人员将生成的许可文件发给部署人员，然后导入许可文件。系统会重启服务器。

图 3-333 注册服务器 2



管理员admin，初始密码为admin。

3.3.13 产品集群部署（单 redis 部署）

产品集群需要redis支持，redis部署分为两种。一种是单redis部署，适应于可靠性要求不高的场景；一种是redis哨兵部署，适应于可靠性要求非常高的场景；本章讲解单redis部署的情况

集群场景说明

集群4个节点部署在两台服务器上，对应的服务器IP、端口、工作目录如下。Redis单独一台服务器部署。以双机双节点的配置ABI集群为例，说明在ABI中如何配置集群。

表 3-48 集群场景说明 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine1	Server1 (主控)	172.21.160.1	9001	/work_dir/ ABlrootdir1/
Machine1	Server2	172.21.160.1	9002	/work_dir/ ABlrootdir2/
Machine2	Server3	172.21.160.2	9003	/work_dir/ ABlrootdir3/
Machine2	Server4	172.21.160.2	9004	/work_dir/ ABlrootdir4/

表 3-49 集群场景说明 2

物理机器	安装软件	IP地址	端口号
Machine3	Redis	192.168.1.103	6379

工作目录指定

涉及到集群的工作目录指定，每个节点必须指定一个单独的工作目录，而且多节点不能同时使用同一个工作目录。

指定工作目录需要在中间件启动文件设置，在Server1的启动文件中设置：

```
-Desen.abi.workdir =/work_dir/ABlrootdir1
```

在Server2的启动文件中设置：

```
-Desen.abi.workdir =/work_dir/ABlrootdir2
```

在Server3的启动文件中设置：

```
-Desen.abi.workdir =/work_dir/ABlrootdir3
```

在Server4的启动文件中设置：

```
-Desen.abi.workdir =/work_dir/ABlrootdir4
```

JVM 参数设置

表 3-50 JVM 参数设置

节点名称	参数设置	备注
Server1	-Desen.abi.workdir=/work_dir/ ABlrootdir1 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	-Xmx和-Xms根据服务器内存大小设置，64位OS建议分配8G-16G 注：jdk1.8开始取消了PermGen，取而代之的是Metaspace，需要将PermSize改为MetaspaceSize，如下：-XX:MetaspaceSize=256m -XX: MetaspaceSize =512m
Server2	-Desen.abi.workdir=/work_dir/ ABlrootdir2 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上
Server3	-Desen.abi.workdir=/work_dir/ ABlrootdir3 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

节点名称	参数设置	备注
Server4	-Desen.abi.workdir=/work_dir/ ABlrootdir4 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

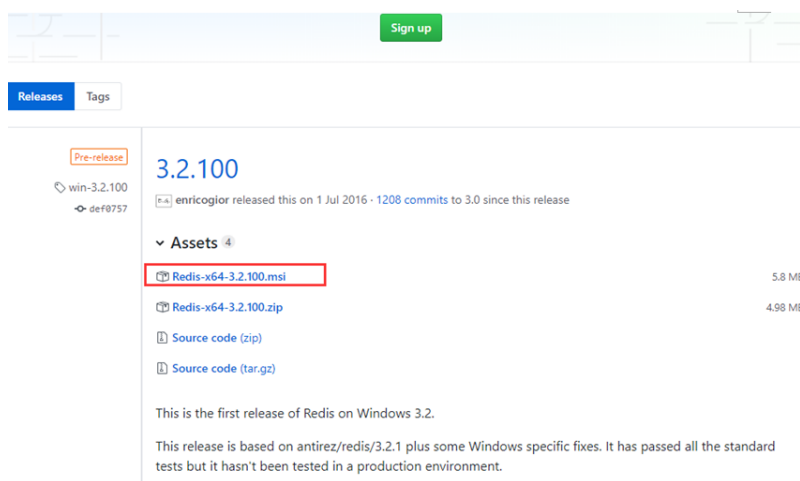
Redis 安装

Redis支持32位和64位的Windows和Linux操作系统。

1. Windows下安装redis

下载地址：<https://github.com/MSOpenTech/redis/releases>

图 3-334 Redis 安装



根据需要选择合适版本的Redis，下载zip包，下载完成后解压缩，会得到一个redis-server.exe。下载完成后，双击安装，安装完成后，启动服务即可

使用客户端测试连接：redis-desktop-manager-0.9.3.817.exe

其实Redis在Windows上不需要安装，在命令行下执行这个redis-server.exe，就会启动redis服务，启动后不要关闭这个命令行窗口，关闭它服务就停止了。

2. Linux下安装redis(支持国产CPU，如：鲲鹏、龙芯等)

redis在linux下的安装使用包管理器来安装即可，对于没有外网的环境也可以复制压缩包，解压后手动安装。

安装

```
# wget http://download.redis.io/releases/redis-5.0.5.tar.gz  
# tar -zxf redis-5.0.5.tar.gz
```

```
# cd redis-5.0.5  
# make  
# make install
```

启动；编译完成以后，启动 redis-server

```
# cd src  
# ./redis-server ../redis.conf &
```

访问redis

用自带的redis-cli客户端工具就可以访问Redis服务器了。

Redis的默认服务端口是6379

```
# redis-cli  
127.0.0.1:6379>
```

Redis配置文件说明

Redis的配置文件Redis.conf在Redis的安装目录下，里面有很多关于Redis的设置。用户可以通过直接修改文件更改配置，也可以通过命令像设置key-value数据库一样修改配置

```
config get xxx //获取某一个配置的值  
config set xxx xxx //设置某一项配置  
config get * 获取所有配置
```

建议直接修改配置文件，因为通过命令修改，重启Redis后就会失效。

表 3-51 主要配置参数列表

参数	值	描述
port	6379	redis的端口
bind	127.0.0.1	绑定的ip地址，默认是127.0.0.1 。 如果本机有多个网卡，且限制只能通过其中一个网卡对应的IP才能访问redis时，可绑定该IP，例：bind IP
protected-mode	yes	保护模式，默认为yes，如果要其他网络能访问，改为no即可
requirepass	-	登录密码，默认是空
maxclients	0	同一时间能够连接到redis的客户端数量，默认是无限制，0表示不限制
maxmemory	-	redis能够使用的最大内存 单位byte, 默认无限制

建议修改如下，如果考虑安全性，设置密码即可：

图 3-335 设置密码

```
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~
#bind 127.0.0.1
# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited.
#
# When protected mode is on and if:
#
# 1) The server is not binding explicitly to a set of addresses using the
#    "bind" directive.
# 2) No password is configured.
#
# The server only accepts connections from clients connecting from the
# IPv4 and IPv6 loopback addresses 127.0.0.1 and ::1, and from Unix domain
# sockets.
#
# By default protected mode is enabled. You should disable it only if
# you are sure you want clients from other hosts to connect to Redis
# even if no authentication is configured, nor a specific set of interfaces
# are explicitly listed using the "bind" directive.
protected-mode no
# Accept connections on the specified port, default is 6379 (IANA #815344).
# If port 0 is specified Redis will not listen on a TCP socket.
port 6391
```

部署包和产品注册和配置

当在所有节点部署ABI包后，需要分别访问下列地址完成产品的注册。

<http://172.21.160.1:9001/ABI>

<http://172.21.160.1:9002/ABI>

<http://172.21.160.2:9003/ABI>

<http://172.21.160.2:9004/ABI>

在所有4个节点均完成注册后，分别登录所有节点，确认所有节点可以单独访问。

集群文件配置

停掉所有服务器，分别在各个工作目录的conf文件夹中放置cluster.properties，其中Server1为主控节点

表 3-52 集群文件配置 1

节点名称	配置内容
Server1	<pre> #charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=true cluster.type=redis cluster.group=group1 cluster.nodeid=node1 cluster.name=测试集群节点1 #单redis连接配置方式 cluster.redis.ip=192.168.1.103 cluster.redis.port=6379 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jovertime=8 #是否使用连接池 #cluster.redis.usepool=false </pre>

节点名称	配置内容
Server2	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node2 cluster.name=测试集群节点2 #单redis连接配置方式 cluster.redis.ip=192.168.1.103 cluster.redis.port=6379 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server3	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node3 cluster.name=测试集群节点3 #单redis连接配置方式 cluster.redis.ip=192.168.1.103 cluster.redis.port=6379 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server4	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node4 cluster.name=测试集群节点4 #单redis连接配置方式 cluster.redis.ip=192.168.1.103 cluster.redis.port=6379 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

配置文件中参数说明:

表 3-53 集群文件配置 2

参数名	描述	例子
cluster.enabled	是否启用集群	默认false不启用
cluster.nodeid	集群节点id	node1
cluster.name	集群节点名称	测试集群节点1
cluster.type	集群类型 (redis: redis集群, netty: netty集群)	redis, 默认使用redis

参数名	描述	例子
cluster.message.retry	消息发送失败后重发次数	3
cluster.health.interval	心跳时间间隔（单位：毫秒）	1500，默认1500
cluster.health.retry	集群节点离线时重试次数，即连续几次没有收到心跳则判定节点离线	3
cluster.heath.rejoin	集群节点离线后重新加入重试次数，即集群节点连续几次都受到心跳消息则判定节点加入	3
cluster.redis.ip	redis ip	172.21.1.65
cluster.redis.port	redis端口	6379
cluster.redis.password	redis密码，没有密码可不传	默认注释掉

启动服务器

集群配置文件完成后，先启动server1，待server1启动完成后，可同时启动其它三个服务器

集群检查

以admin身份登录ABI，依次进入“系统设置 > 性能与维护 > 集群信息”，检查如下信息：

- 检查所有配置节点是否存在
- 检查所有节点在线状态
- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

集群扩容说明

新增N个节点，与原集群各节点配置保持一致，连接同一个Redis，即可达到集群扩容目的。

例如新增物理机器Machine4，节点名称Server5,IP地址172.21.160.3，端口号9005

表 3-54 集群扩容说明 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine4	Server5	172.21.160.3	9005	/work_dir/ ABlrootdir5/

指定工作目录、设置JVM参数后，正常注册ABI（参考本章第3、4、6节内容），然后停掉新增节点服务器，在工作目录的conf文件夹中放置cluster.properties，配置内容为：

表 3-55 集群扩容说明 2

节点名称	配置内容
Server5	<pre>#必须配置的属性： cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node5 cluster.name=测试集群节点5 #需要二选一配置的属性： #集群连接配置方式1 cluster.redis.ip=192.168.1.103 cluster.redis.port=6379 #视情况配置的属性： #如果redis设置了密码，则配置 #cluster.redis.password=123 #可以不配置的属性： #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

📖 说明

该配置文件中cluster.redis.ip与原集群各节点保持一致。

然后启动新增节点服务器，启动成功后以admin身份登录ABI，依次进入“系统设置 > 性能与维护 > 集群信息”，检查如下信息：

- 检查所有配置节点是否存在
- 检查所有节点在线状态

- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

注意事项

1. 集群内各服务器版本需保持一致
只能同一版本的服务器做集群
2. 各服务器的数据库配置需一致
各服务器数据库配置需要保持一致，其中url中ip地址也要一致
3. 各服务器redis配置需一致
各服务器配置的redis相关参数需要是一致的，如redis ip、端口、密码

3.3.14 产品集群部署（redis 哨兵部署）

产品集群需要redis支持，redis部署分为两种。一种是单redis部署，适应于可靠性不高的场景；一种是redis哨兵部署，适应于可靠性要求非常高的场景；本章讲解redis哨兵部署的情况

集群场景说明

集群4个节点部署在两台服务器上，对应的服务器IP、端口、工作目录如下。Redis哨兵模式部署和应用服务器部署在一起。以双机双节点的配置ABI集群为例，说明在ABI中如何配置集群。

表 3-56 集群场景说明 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine1	Server1（主控）	172.21.160.1	9001	/work_dir/ ABlrootdir1/
Machine1	Server2	172.21.160.1	9002	/work_dir/ ABlrootdir2/
Machine2	Server3	172.21.160.2	9003	/work_dir/ ABlrootdir3/
Machine2	Server4	172.21.160.2	9004	/work_dir/ ABlrootdir4/

表 3-57 集群场景说明 2

物理机器	安装软件	IP地址	端口号
Machine1	Redis（主）	172.21.160.1	6391
Machine2	Redis（从）	172.21.160.2	6392
Machine1	Sentinel	172.21.160.1	26379
Machine2	Sentine2	172.21.160.2	26379

物理机器	安装软件	IP地址	端口号
Machine2	Sentine3	172.21.160.2	26380

工作目录指定

涉及到集群的工作目录指定，每个节点必须指定一个单独的工作目录，而且多节点不能同时使用同一个工作目录。

指定工作目录需要在中间件启动文件设置，在Server1的启动文件中设置：

-Desen.abi.workdir =/work_dir/ABlrootdir1

在Server2的启动文件中设置：

-Desen.abi.workdir =/work_dir/ABlrootdir2

在Server3的启动文件中设置：

-Desen.abi.workdir =/work_dir/ABlrootdir3

在Server4的启动文件中设置：

-Desen.abi.workdir =/work_dir/ABlrootdir4

JVM 参数设置

表 3-58 JVM 参数设置

节点名称	参数设置	备注
Server1	-Desen.abi.workdir=/work_dir/ ABlrootdir1 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	-Xmx和-Xms根据服务器内存 大小设置，64位OS建议分配 8G-16G 注：jdk1.8开始取消了 PermGen，取而代之的是 Metaspace，需要将 PermSize改为 MetaspaceSize，如下： - XX:MetaspaceSize=256m -XX: MetaspaceSize =512m

节点名称	参数设置	备注
Server2	-Desen.abi.workdir=/work_dir/ ABlrootdir2 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上
Server3	-Desen.abi.workdir=/work_dir/ ABlrootdir3 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上
Server4	-Desen.abi.workdir=/work_dir/ ABlrootdir4 -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK -Xms4096m -Xmx8192m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true	同上

Redis 安装

参考redis安装

redis 哨兵模式部署

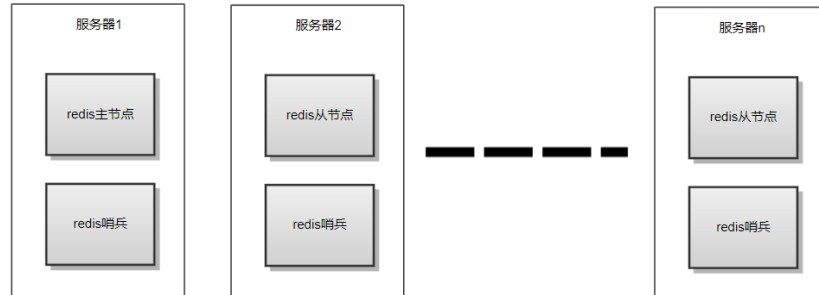
1. 简述

Redis哨兵模式的作用是提高Redis高可用性，当Redis主节点异常退出时，会有从节点替换为主节点，保证Redis可用。

2. 部署架构

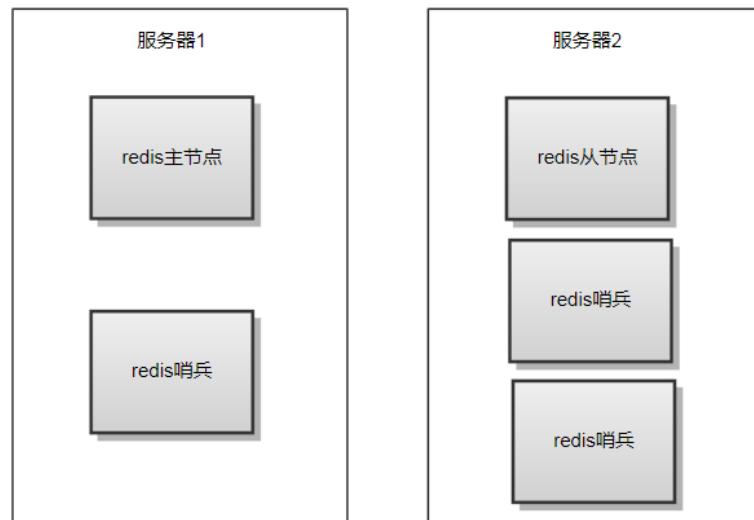
Redis哨兵模式要求最少有1主1从3哨兵，由此推荐三台及以上服务器部署架构图如下：

图 3-336 部署架构 1



两台服务器部署架构图如下：

图 3-337 部署架构 2



3. Linux 下部署

部署过程以两台服务器1主1从3哨兵为例，进行redis哨兵模式安装
启动Redis服务器

先正常启动两个redis，参考第七章第4节redis安装

配置从节点

进入到主节点redis安装目录下，连接从节点、执行命令：

```
# cd /{redis-path}/src 注：{redis-path}为redis安装目录，按实际填写  
# ./redis-cli -h 172.21.160.2 -p 6392
```

-h：从节点IP；-p：从节点端口；172.21.160.2：6392> slaveof 172.21.1601 6391。

说明：声明主节点

修改哨兵配置文件

sentinel.conf文件在redis安装目录下。

注意其中两个关键配置

- a. port 指定哨兵端口
- b. sentinel monitor mymaster 172.21.160.1 6391 2

其中mymaster表示哨兵名称，172.21.160.1 6391是主节点的ip和端口，表示监控主节点，最后的数字2是哨兵用来判断某个 redis 服务器是否下线的参数，表示投票需要的"最少法定人数"，比如有10个sentinal哨兵都在监控某一个主节点，如果需要至少6个哨兵发现主节点异常退出后，才认为主节点真正down掉，那么这里就配置为6，最小配置1台主节点，1台从节点。

如果一台服务器有多个哨兵，需要准备多个sentinel.conf文件一一对应，例如172.21.160.2上有端口26380,26381的两个哨兵，就需要两个sentinel.conf文件，命名为sentinel1.conf、sentinel2.conf，其中的哨兵名称要配置成一样的，并且都监控主节点，只要修改端口即可。

📖 说明

sentinel.conf的哨兵名称要与后面集群文件里的哨兵名称保持一致

图 3-338 参考配置

```
# sentinel down-after-milliseconds <master-name> <milliseconds>
#
# Number of milliseconds the master (or any attached replica or sentinel) should
# be unreachable (as in, not acceptable reply to PING, continuously, for the
# specified period) in order to consider it in S_DOWN state (Subjectively
# Down).
#
# Default is 30 seconds.
sentinel deny-scripts-reconfig yes

# sentinel parallel-syncs <master-name> <numreplicas>
#
# How many replicas we can reconfigure to point to the new replica simultaneously
# during the failover. Use a low number if you use the replicas to serve query
# to avoid that all the replicas will be unreachable at about the same
# time while performing the synchronization with the master.
sentinel monitor mymaster 172.21.160.1 6391 2

# sentinel failover-timeout <master-name> <milliseconds>
#
# Specifies the failover timeout in milliseconds. It is used in many ways:
#
# - The time needed to re-start a failover after a previous failover was
#   already tried against the same master by a given Sentinel, is two
#   times the failover timeout.
#
```

启动哨兵服务进程

```
# cd /{redis-path}/src
# ./redis-sentinel /{path}/sentinel.conf &
注: {redis-path}、{path}为redis安装路径、哨兵的sentinel.conf文件所在路径，启动多个哨兵时指定各自配置文件。
```

示例：在172.21.160.2上启动两个哨兵服务

图 3-339 示例 1

```
[root@localhost redis-5.0.5]# pwd
/opt/redis-5.0.5
[root@localhost redis-5.0.5]# ls
00-RELEASENOTES  CONTRIBUTING  deps  Makefile  README.md  runtest  runtest-moduleapi  sentinel.conf  src  utils
BUGS             COPYING      INSTALL  MANIFESTO  redis.conf  runtest-cluster  runtest-sentinel  sentinel.conf2  tests
[root@localhost redis-5.0.5]# cd src/
[root@localhost src]# pwd
/opt/redis-5.0.5/src
[root@localhost src]# ./redis-sentinel ../sentinel.conf &
```

图 3-340 示例 2

```
[root@localhost redis-5.0.5]# pwd
/opt/redis-5.0.5
[root@localhost redis-5.0.5]# ls
00-RELEASENOTES  CONTRIBUTING  deps  Makefile  README.md  runtest  runtest-moduleapi  sentinel2.conf  src  utils
BUGS             COPYING      INSTALL  MANIFESTO  redis.conf  runtest-cluster  runtest-sentinel  sentinel.conf  tests
[root@localhost redis-5.0.5]# cd src/
[root@localhost src]# pwd
/opt/redis-5.0.5/src
[root@localhost src]# ./redis-sentinel ../sentinel2.conf &
```

查看哨兵模式是否启动

Redis主从节点检查:

连接主节点: # ./redis-cli -h 172.21.160.1 -p 6391

执行命令: info replication

图 3-341 检查

```
[root@localhost src]# ./redis-cli -h 172.21.160.1 -p 6391
172.21.160.1:6391> info replication
# Replication
role:master
connected_slaves:1
slave0:ip=172.21.160.2,port=6392,state=online,offset=18744,lag=0
master_replid:dd9884e125f2473bc81a57d4a8ec19d8495a7c27
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:18744
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:18744
```

哨兵进程检查:

执行命令: ps -ef|grep redis

Redis配置密码说明

如果redis主从节点需要设置密码, 则需要在配置文件加入如下配置参数:

a. **redis.conf配置示例如下:**

主redis:

requirepass 123 #连接该redis需要密码123

masterauth 123 #主redis的密码123 (为从redis节点时生效)

从redis:

requirepass 123 #连接该redis需要密码123

masterauth 123 #主redis的密码123 (为从redis节点时生效)

图 3-342 配置示例

```
# refuse the replica request.
#
# masterauth <master-password>
masterauth 123
requirepass 123
# When a replica loses its connection with the master and
# is still in progress, the replica can act in two ways:
#
```

说明

主从redis节点大于2时, redis密码必须保持一致, 在主从切换时, 保证主从正常通讯进行数据同步

b. **配置主从节点时连接redis-cli后需要验证密码, 示例如下:**

图 3-345 集群文件配置

```
cluster.enabled=true
cluster.iscontrol=true
cluster.type=redis
cluster.group=group1
cluster.nodeid=node1
cluster.name=server1
cluster.redis.sentinel=mymaster
cluster.redis.nodes=172.21.160.1:26379,172.21.160.2:26379,172.21.160.2:26380
~
~
```

以下为详细配置内容，只做参考：

表 3-59 详细配置

节点名称	配置内容
Server1	<pre>#charset UTF-8 #必须配置的属性： cluster.enabled=true cluster.iscontrol=true cluster.type=redis cluster.group=group cluster.nodeid=node1 cluster.name=测试集群节点1 #mySentinel为哨兵名称 cluster.redis.sentinel=mymaster cluster.redis.nodes=172.21.160.1:26379,172.21.160.2:26379,172.21.160.2:26380 #视情况配置的属性： #如果redis设置了密码，则配置 #cluster.redis.password=123 #可以不配置的属性： #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server2	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group cluster.nodeid=node2 cluster.name=测试集群节点2 #mySentinel为哨兵名称 cluster.redis.sentinel=mymaster cluster.redis.nodes=172.21.160.1:26379,172.21.160.2:26379,172.21.160.2:26380 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server3	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group cluster.nodeid=node3 cluster.name=测试集群节点3 #mySentinel为哨兵名称 cluster.redis.sentinel=mymaster cluster.redis.nodes=172.21.160.1:26379,172.21.160.2:26379,172.21.160.2:26380 #视情况配置的属性: #如果redis设置了密码,则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

节点名称	配置内容
Server4	<pre>#charset UTF-8 #必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group cluster.nodeid=node4 cluster.name=测试集群节点4 #mySentinel为哨兵名称 cluster.redis.sentinel=mymaster cluster.redis.nodes=172.21.160.1:26379,172.21.160.2:26379,172.21.160.2:26380 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

启动服务器

集群配置文件完成后, 先启动server1, 待server1启动完成后, 可同时启动其它三个服务器

集群检查

以admin身份登录ABI, 依次进入“系统设置 > 性能与维护 > 集群信息”, 检查如下信息:

- 检查所有配置节点是否存在
- 检查所有节点在线状态
- 检查所有节点同步状态

- 检查是否存在控制节点，且只有一个

集群扩容说明

新增N个节点，与原集群各节点配置保持一致，连接同一个Redis，即可达到集群扩容目的。

例如新增物理机器Machine4，节点名称Server5,IP地址172.21.160.3，端口号9005

表 3-60 集群扩容说明 1

物理机器	节点名称	IP地址	端口号	工作目录
Machine4	Server5	172.21.160.3	9005	/work_dir/ ABlrootdir5/

指定工作目录、设置JVM参数后，正常注册ABI，然后停掉新增节点服务器，在工作目录的conf文件夹中放置cluster.properties，配置内容为：

表 3-61 集群扩容说明 2

节点名称	配置内容
Server5	<pre>#必须配置的属性: cluster.enabled=true cluster.iscontrol=false cluster.type=redis cluster.group=group1 cluster.nodeid=node5 cluster.name=测试集群节点5 #mySentinel为哨兵名称 cluster.redis.sentinel=mymaster cluster.redis.nodes=172.21.160.1:26379,172.21.160.2:26379,172.21.160.2:26380 #视情况配置的属性: #如果redis设置了密码, 则配置 #cluster.redis.password=123 #可以不配置的属性: #集群服务器与redis最大时间差 默认120s #cluster.health.maxtimegap=120 #两次心跳间隔时间默认3s #cluster.health.interval=3000 #离开时间间隔算离开 默认30s #cluster.health.leavetime=30 #加入时间算加入 默认8s #cluster.heath.jointime=8 #是否使用连接池 #cluster.redis.usepool=false</pre>

📖 说明

该配置文件中cluster.redis.ip与原集群各节点保持一致。

然后启动新增节点服务器，启动成功后以admin身份登录ABI，依次进入“系统设置 > 性能与维护 > 集群信息”，检查如下信息：

- 检查所有配置节点是否存在
- 检查所有节点在线状态
- 检查所有节点同步状态
- 检查是否存在控制节点，且只有一个

注意事项

1. 集群内各服务器版本需保持一致
只能同一版本的服务器做集群
2. 各服务器的数据库配置需一致
各服务器数据库配置需要保持一致，其中url中ip地址也要一致
3. 各服务器redis配置需一致
各服务器配置的redis相关参数需要是一致的，如redis ip、端口、密码
4. Redis主从切换说明

图 3-346 场景

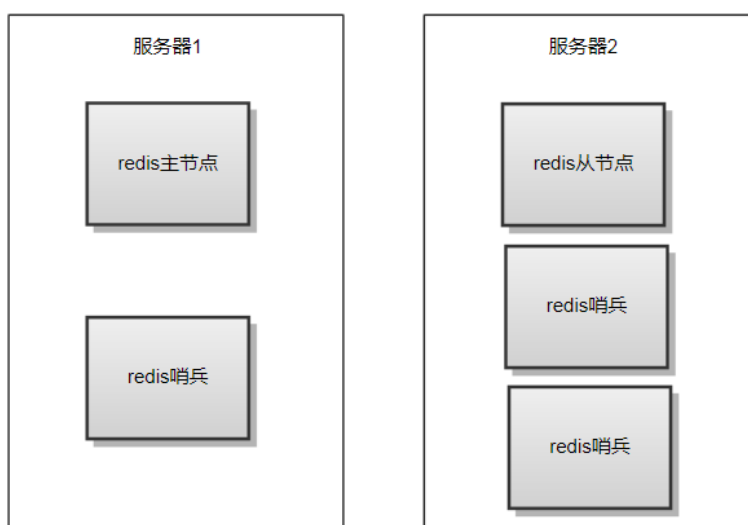


表 3-62 Redis 主从切换说明

物理机器	安装软件	IP地址	端口号
Machine1	Redis (主)	172.21.160.1	6391
Machine2	Redis (从)	172.21.160.2	6392
Machine1	Sentinel	172.21.160.1	26379
Machine2	Sentine2	172.21.160.2	26379
Machine2	Sentine3	172.21.160.2	26380

当redis部署架构如上图时，主从切换以后，主节点为服务器2，有两个redis哨兵；从节点为服务器1，有一个redis哨兵。这时如果主节点再次异常退出，两个redis哨兵也就异常退出了而无法参与主从redis切换投票，主从切换就会失败，所以当redis主从切换一次后，需要进行人工干预再主从切换回来。

操作：

- a. 连接原从节点，执行slaveof 172.21.160.1 6391

```
./redis-cli -h 172.21.160.2 -p 6392
172.21.160.2:6392> slaveof 172.21.160.1 6391
```

b. 连接原主节点，执行slaveof no one

```
./redis-cli -h 172.21.160.1 -p 6391
172.21.160.1:6391> slaveof no one
```

3.3.15 系统诊断

很多问题其实都是部署的时候没有正确设置某些参数，或者没有按要求修改配置文件，往往在遇到异常时很难定位到这些错误，ABI5.3.1能自动诊断这些文件，并加以提示。

在管理员登录界面后单击会出现如下诊断界面

图 3-347 系统诊断 1

下面列出的是当前配置和状态的检测结果。如果存在警告（黄色文字）或错误（红色文字）时请尽快修复。

诊断内容	描述
最大内存设置	系统最大可用内存设置较少，目前是1.8G，32位系统最好大于1G，64位最好大于2G，可以通过jvm的命令行参数 "-Xmx2048m -Xms512m" 来设置（您的系统...
可用字体	通过，可用中文字体：仿宋, 华文宋体, 华文仿宋, 华文宋体, 华文彩云, 华文新魏, 华文楷体, 华文细圆, 华文细黑, 华文隶书, 宋体, 幼圆, 微软雅黑, 新宋体, 方正三...
注册信息	通过，系统剩余使用天数：364天
java.awt.headless设置	通过
时区设置	通过，是Asia/Shanghai
操作系统字符集	通过
file.encoding设置	通过，是GBK
sun.jnu.encoding设置	通过，是GBK
可用内存	通过，FREE=483M TOTAL=844M MAX=1.8G

系统进行了下列检测

表 3-63 检测

诊断项	检测内容（中文）	The detection content (English)
时区设置	必须是Asia/Shanghai，否则提示错误	必须是Asia/Shanghai，否则提示错误
file.encoding设置	必须是"UTF-8","UTF8","GB18030","GBK",否则提示错误	必须是"UTF-8",否则提示错误
sun.jnu.encoding设置	同上	同上
操作系统字符集	检查文件名或文件内容是否支持GBK编码，不支持提示错误	检查文件名或文件内容是否支持"UTF-8"编码，不支持提示错误
java.awt.headless设置	windows不检查，linux检查或未设置提示错误	windows不检查，linux检查或未设置提示错误
最大内存设置	32位要求大于1024M，64位要求大于2048M	32位要求大于1024M，64位要求大于2048M

诊断项	检测内容 (中文)	The detection content (English)
可用内存	<%5 系统已几乎无可用内存 <30% 系统可用内存偏低 >=30% 通过	<%5 系统已几乎无可用内存 <30% 系统可用内存偏低 >=30% 通过
可用字体	必须有宋体字体, 否则不能通过	-
检查超级管理员密码	通过注释不能和用户名相同, 设置的"123456"等;	提示密码需要字符和数字组合
注册码检查	>0 & <30, 提示 线程池<10, 提示 同时在线人数<50, 提示	-

3.3.16 疑难问题处理方案

3.3.16.1 中间件相关

1. 双网卡、Vmware

说明: 软件注册会绑定网卡的MAC地址, 所以在有双网卡的或者安装了VMware软件的时候, 请确认根据那个网卡注册的。并且双网卡有可能会引起集群失效的问题, 建议先关闭一个网卡, 注册和配置完毕后再开启。

2. Tomcat

如果使用tomcat部署, 需要修改配置文件server.xml中的WEB应用端口对应<Connector段标签属性值配置maxPostSize, 值设置为20971520

形如: <Connector maxPostSize="20971520"

以支持超大表 (看板内容可能超过2M, 凡是超过2M的表, 想要保存必须设置此属性), tomcat不配置此值时此值默认为2097152, 即为2MB。

3.3.16.2 数据库相关

1. sybase

- a. Sybase环境下需要修改字符集, 不修改可能会出现乱码以及因为大字段导致的数据库异常退出

解决步骤:

在服务器上打开MS-DOS窗口, 进入sybase安装目录中的charsets\cp936目录, 如:

```
cd sybase\charsets\cp936
```

输入以下命令加载cp936字符集:

```
charset -Usa -P -SWHDS binary.srt cp936
```

图 3-348 加载

```
D:\sybase\charsets\cp936>charset -Usa -P -SWHDS binary.srt cp936
Loading file 'binary.srt'.

Found a [sortorder] section.

This is Class-1 sort order.

Finished loading the Character Set Definition.

Finished loading file 'binary.srt'.

1 sort order loaded successfully
```

进入isql，查询cp936字符集对应的id

图 3-349 查询

```
D:\sybase\charsets\cp936>isql -Usa -P -SWHDS
Msg 2401, Level 11, State 2:
Server 'WHDS':
Character set conversion is not available between client character set 'cp936'
and server character set 'cp850'.
No conversions will be done.
1> select name,id from syscharsets where name='cp936'
2> go
   name                id
-----
cp936                  171
(1 row affected)
```

设置default character set id参数值为171（即cp936对应的id）

图 3-350 设置

```
1> sp_configure 'default character set id',171
2> go
In changing the default sort order, you have also reconfigured SQL Server's
default character set.
Parameter Name          Default      Memory Used Config Value
Run Value  Unit
-----
default character set id 2            0          171
2 id          static

(1 row affected)
Configuration option changed. Since the option is static, Adaptive Server must
be rebooted in order for the change to take effect.
Changing the value of 'default character set id' to '171' increases the amount
of memory ASE uses by 4 K.
(return status = 0)
```

完成后，重启实例两次（第一次启动会失败）

- b. Sybase环境下，恢复备份包以及某些操作会出现提示锁不够的问题

解决步骤：

修改锁使用数量

```
sp_configure "number of locks",100000
```

如果不够，继续增加到30W

- c. Sybase环境下，提示错误：Space available in the log segment has fallen critically low in database 'tempdb'. All future modifications to this database will be suspended until the log is successfully dumped and space becomes available.

解决步骤：

tempdb默认参数3M太小，修改至200M

```
1> sp_cacheconfig 'tempdb_cache','200m','mixed' 2> go 1> sp_bindcache  
'tempdb_cache',tempdb 2> go
```

需要重启

2. oracle

Oracle环境下，提示ORA-12519: TNS:no appropriate service handler found可能是数据库上当前的连接数目已经超过了它能够处理的最大值

解决步骤：

```
select count(*) from v$process --当前的连接数  
select value from v$parameter where name = 'processes' --数据库允许的最大连接数
```

修改最大连接数：

```
alter system set processes = 500 scope = spfile;
```

重启数据库：

```
shutdown immediate;  
startup;
```

3. db2

db2环境下，展开左树上计划任务节点，报错：： [jcc][10120][11936][3.52.95]
操作无效：已关闭 Lob。 ERRORCODE=-4470, SQLSTATE=null

解决步骤：

按下面格式配置默认连接池的url。

```
jdbc:db2://192.168.1.222:50000/  
testdb:driverType=4;fullyMaterializeLobData=true;fullyMaterializeInputStreams=true;progressiveStream  
ing=2;progressiveLocators=2;
```

testdb为数据库名。注意，最后有个分号。

4. mysql

MySQL环境下，初始化插件失败

提示如下错误：

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name  
'platform_pluginmgr' defined in file [/var/lib/tomcat/webapps/qbi/WEB-INF/classes/com/esen/  
platform/plugin/ActivexPluginsManager.class]: Invocation of init method failed; nested exception is  
com.esen.exception.RuntimeException4I18N: 初始化插件管理器失败!
```

解决步骤：

[mysqld]段下面添加max_allowed_packet =20M，添加完毕重启MySQL服务器。
Windows修改my.ini， linux修改/etc/my.cnf

```
[mysqld]  
max_allowed_packet =20M
```

另外还需要加上表名不区分大小写参数

```
Lower_case_table_names=1
```

5. Hive

a. ABI中打入Hive驱动之前，需要将提供的Hive驱动中，按路径查找org\json
\包，删掉org\json整个文件夹。

b. 如果没有按照（1）中删除org\json整个文件夹，直接在ABI中打入Hive驱
动，如果出现打印设置页面报错：Cannot read property 'length' of
undefinde 或者提示报错：gc overhead limit exceeded（内存不够）

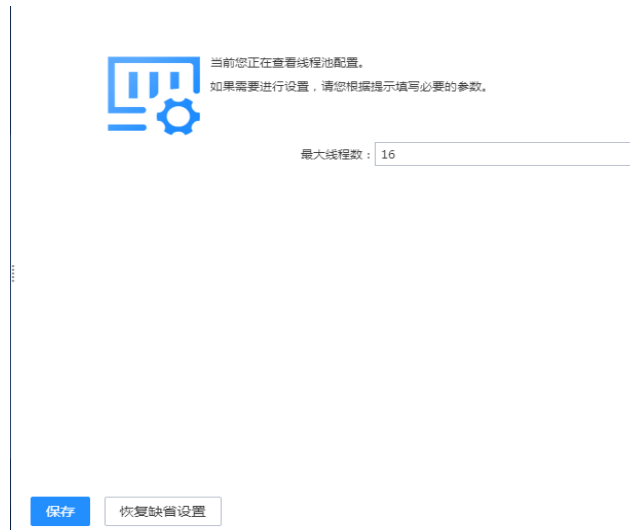
解决步骤：

在提供的Hive驱动中，按路径查找org\json\包，删掉org\json整个文件夹
后，重启服务器即可。

3.3.16.3 应用相关

1. 后台提示：数据库连接池已满或者连接超时
解决方法：修改JDBC中的maxactive值，设置为100
2. 计算报表的时候提示正在计算，有多少用户在排队
解决方法：修改线程数，设置为CPU个数*4

图 3-351 图示



3.3.16.4 Redis 部署相关

问题现象：由于某些项目服务器的操作系统是最小安装的，安装redis时会提示缺少gcc等rpm包

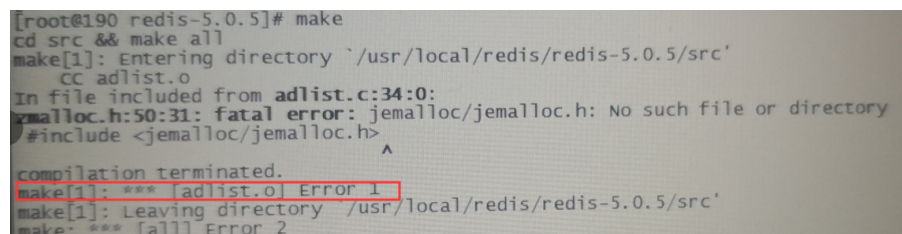
解决办法：

1. 如果处于外网环境（能ping通baidu.com），可直接执行命令进行安装

```
# yum install -y gcc-c++
```


如果缺少其他依赖包，同样可以用yum进行安装
重新Make可能会遇到如下图报错

图 3-352 报错



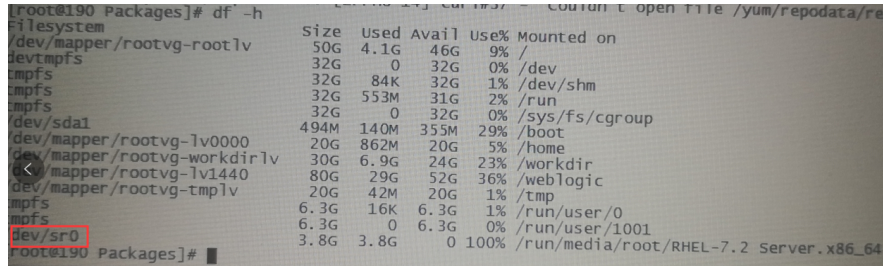
先清理上一次make产生的文件再次make

```
# make distclean  
# make
```

2. 如果处于内网环境，则需要如下几步配置yum源实现yum安装：

- a. 给服务器所在虚拟机或物理机插入相同版本的操作系统光盘
- b. 在操作系统层面将光盘挂载在某一个目录下，如下图

图 3-353 挂载



```
# mount /dev/sr0 /mnt/cdrom
```

#/dev/sr0为光盘路径，/mnt/cdrom为挂载路径。

- c. 在/etc/yum.repos.d目录下备份原有文件，新建local.repo文件，后缀必须为.repo，增加如下内容：

```
[local] #必须，唯一的repo名称  
name=local #自定义名称  
baseurl=file:///mnt/cdrom #光盘mount后的路径  
enable=1 #启用该repo，0为禁用  
gpgcheck=0 #不检查rpm包来源
```

3.3.16.5 Docker 相关、IPv6 部署

Docker 相关

要求1：工作目录的指定必须在中间件上配置或者在war包中直接指定

要求2：docker 启动时，需要将工作目录指定到宿主机上，例子如下

```
docker run -d -p 80:8080 -v /opt/workdir:/opt/workdir tomcat
```

IPv6 部署

ABI531支持ip v6部署，此处以tomcat为例介绍如何配置

- 1. 修改tomcat/conf/server.xml文件，找到下图位置，添加address=" ip v6地址”，

图 3-354 ip v6 地址

```
<Connector port="8080" protocol="HTTP/1.1" address="fd15:4ba5:5a2b:1008:d8b3:df29:fd2:7776"  
connectionTimeout="20000"  
redirectPort="8443" />
```

- 2. 其余配置和ip v4一样，可参考前面章节
- 3. 初始化ABI服务器，在数据库设置这里，填写链接地址时，ipv6地址要用[]括起来，即标红部分。如使用oracle11g，则参照下面写法
jdbc:oracle:thin:@//[fd15:4ba5:5a2b:1008:d8b3:df29:fd2:7776]:1523/orcl
- 4. 注册成功后，ipv6访问地址也要用[]括起来，如
http://[fd15:4ba5:5a2b:1008:d8b3:df29:fd2:7776]:8080/abi

3.3.17 附录

ABI5.3.1 测试通过环境组合

中间件

📖 说明

仅支持jdk1.8及以上版本

表 3-64 中间件

中间件	jdk	操作系统	备注
中创InforSuiteAS	Open jdk1.8	银河麒麟4.02	-
TongWeb6.1.5.8	Open jdk1.8	银河麒麟4.02	-
TAS2.8.5	Open jdk1.8	银河麒麟4.02	-
Tomcat7.0	Sun jdk1.8	CentOS 6.5 Win2008	建议使用tomcat7.0.79以后的小版本
Websphere8.5	自带jdk1.8	CentOS 6.5 Win7	-
Weblogic12c	Sun jdk1.8	CentOS 7.2	-
金蝶9.0	openjdk1.8	uos操作系统 (龙芯cpu)	-

数据库

表 3-65 数据库

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
星环6.0	分析库	无	不能	只兼容bi, 豌豆bi部分, 指标管理, 数据整合 驱动类名: org.apache.hive.jdbc.HiveDriver URL地址: jdbc:inceptor2://172.17.3.41:10000/<database_name>
PostgreS QL9.6	系统库	无	能	-

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
Gauss DB200	分析库	无	不能	只兼容bi部分。 驱动类名： org.postgresql.Driver 注，Gauss DB200驱动和ABI内置 GreenPlum驱动冲突，需要在ABI 将GreenPlum驱动删除，再打入 Gauss DB200驱动文件 链接地址： <a href="jdbc:postgresql://127.0.0.1:5432/<database name>">jdbc:postgresql://127.0.0.1:5432/ <database name>
Informix 11.5	系统库 &分析 库	无	能	只兼容bi、豌豆bi部分。 驱动类名： com.informix.jdbc.IfxDriver 说明 informix 11.5驱动比ABI内置 informix12驱动版本低，需要在ABI 将informix12驱动删除，再打入 informix 11.5驱动文件 链接地址： sqli:// 127.0.0.1:9088/dbname: INFORMIXSERVER=ifxserver; NEWCODESET=utf8,8859-1,819; CLIENT_LOCALE=en_US.utf8; DB_LOCALE=en_US.8859-1 说明 informix链接有两种写法，在作为系 统库和分析库时不可混用，也不可交 替使用，推荐使用上方的链接写法。
达梦 DM7.6	系统库 &分析 库	有	能	驱动类型： dm.jdbc.driver.DmDriver 链接地址： url=jdbc:dm://127.0.0.1:5236
易鲸捷 EsgynDB 2.4.0	系统库 &分析 库	有	能	只兼容bi和填报部分。 驱动类型： org.trafodion.jdbc.t4.T4Driver 链接地址： jdbc:t4jdbc:// 172.17.3.191:23400/:schema=TA CS

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
Sqlserve r2016	系统库 &分析 库	有	能	-
Oracle1 1g	系统库 &分析 库	有	能	驱动类型： oracle.jdbc.driver.OracleDriver 链接地址： jdbc:oracle:thin:@192.168.1.102: 1521:sid
Db2 10.5	系统库 &分析 库	有	能	不支持指标管理、数据整合 驱动类型： com.ibm.db2.jcc.DB2Driver 链接地址： jdbc:db2://192.168.1.224:50000/ testdb
Mysql5. 6 Mysql5. 7	系统库 &分析 库	有	能	驱动类型： com.mysql.jdbc.Driver 链接地址： jdbc:mysql://192.168.1.223/ testdb? useUnicode=true&characterEnc oding=utf8 作为系统库，需要5.7版本
Petabas e2.1	分析库	无	不能	分析只支持BI分析 驱动类型： com.esen.jdbc.PetaBaseDriver 链接地址： jdbc:petabase://127.0.0.1:10000/ testdb
人大金仓 7 人大金仓 8	系统库 &分析 库	7有8无	能	分析库（不支持填报和工作流、 指标管理），支持表单。 驱动类型： com.kingbase.Driver（7） com.kingbase8.Driver（8） 链接地址： jdbc:kingbase://192.168.1.247/ testdb（7） jdbc:kingbase8://192.168.1.247/ testdb（8）

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
GreenPlum5.20.1	支持BI分析和敏捷分析、数据整合	有	不能	驱动类型： com.pivotal.jdbc.GreenplumDriver 链接地址： jdbc:pivotal:greenplum://172.21.150.69:5432; DatabaseName=gsgs
河图6.5.1	分析库	无	不能	支持BI分析和敏捷分析 驱动类型： io.prestosql.jdbc.PrestoDriver 链接地址： jdbc:presto://10.146.11.204:24002,10.146.11.205:24002,10.146.11.206:24002/hive/rz? serviceDiscoveryMode=zooKeeper&zooKeeperNamespace=hsbroker 扩展参数示例： jdbc.KerberosKeytabPath=D:\test\huawei\user.keytab jdbc.SSL=true jdbc.KerberosConfigPath=D:\test\huawei\krb5.conf jdbc.KerberosPrincipal=lkjt_hycj jdbc.KerberosRemoteServiceName=HTTP jdbc.tenant=default jdbc.deploymentMode=on_yarn jdbc.KerberosServicePrincipalPattern=\${SERVICE}@\${HOST} jdbc.SSLTrustStorePath=d:\test\huawei\hetuserver.jks
Gbase8a8.6.2.43	系统库	有	能	支持bi分析、填报、敏捷分析（不支持 workflow、指标、数据整合、表单）

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
Clickhouse20.6.4	分析库	无	不能	<p>驱动地址： ru.yandex.clickhouse.ClickHouse Driver</p> <p>链接地址： jdbc:clickhouse:// 172.21.160.69:8123/abi525</p> <p>注意事项：</p> <ol style="list-style-type: none"> 1. 不支持以批量复制主题表的方式新建主题表。 2. 不支持多个窗口函数一起使用，如使用两个或两个以上的_rk()函数。 3. 不支持使用TOP%函数。 4. 不支持使用求标准方差函数stdev()。 5. Illegal types Decimal(38, 2) and Float64 of arguments of function multiply, 类似这样的报错，是因为两个数据类型不一致引起的，解决办法有两种，一种是像asnum(fszhzb.ojse) +10.0这样，用asnum()函数转下，另一种是将参数类型改为Float64。 6. 出现“Division by zero”错误提示，有两种解决方法，第一种在主题集属性-分析对象属性这里，分母为零时的处理选择“过滤指标中分母为0或者Null的数据”，第二种是在分析区表格里的 高级 “0和NULL的处理”，将“过滤指标中的分母为0或者null的数据”勾选解决，如果每个分析表都设置很麻烦，建议主题集设置。

客户端

表 3-66 客户端

操作系统	浏览器
Win7	IE11, Chrome
Win10	IE11, Chrome
银河麒麟4.02	360安全版

360安全版和Chrome不支持Word分析报告编辑插件

ABI5.3.1测试通过环境组合

服务器及硬件

表 3-67 服务器

服务器	CPU
联想、长城	飞腾、龙芯、兆芯、鲲鹏

数据库

表 3-68 数据库

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
达梦 DM7.6	系统库 &分析 库	有	能	驱动类型： dm.jdbc.driver.DmDriver 链接地址： url=jdbc:dm://127.0.0.1:5236
人大金仓 7 人大金仓 8	系统库 &分析 库	7有8无	能	分析库（不支持填报和工作流、 指标管理），支持表单。 驱动类型： com.kingbase.Driver（7） com.kingbase8.Driver（8） 链接地址： jdbc:kingbase://192.168.1.247/ testdb（7） jdbc:kingbase8://192.168.1.247/ testdb（8）

数据库	分析库 or系统 库	包中是 否含驱 动文件	是否能修改数 据库表结构或 数据	备注
Gbase 8a 8.6.2.43	系统库	有	能	支持bi分析、填报、敏捷分析（不 支持工作流、指标、数据整合、 表单）

中间件

📖 说明

仅支持jdk1.8及以上版本。

表 3-69 中间件

中间件	jdk	操作系统
中创InforSuiteAS	Open jdk1.8	银河麒麟4.02
TongWeb6.1.5.8	Open jdk1.8	银河麒麟4.02
TAS3.0	Open jdk1.8	银河麒麟4.02
金蝶9.0	openjdk1.8	uos操作系统（龙芯 cpu）

ABI参数配置总结

常用中间件设置JVM参数

1. Tomcat:

修改内存大小一般在启动文件startup.bat或startup.sh中设置，设置内存不能低于如下数值：

Windows下：注：jdk1.8开始取消了PermGen，取而代之的是Metaspace

Set JAVA_OPTS=-Xmx2048m -Xms2048m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=256m

Linux下：注：jdk1.8开始取消了PermGen，取而代之的是Metaspace

export JAVA_OPTS=-Xmx2048m -Xms2048m -XX:MetaspaceSize=128m -
XX:MaxMetaspaceSize=256m

2. WebLogic 11g（其它版本大致不变）：

单机环境：

在startWebLogic.sh（cmd）中，增加

Windows环境：Set JAVA_OPTIONS=-Xmx2024m -Xms2048m -

XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/
Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK

Linux环境：JAVA_OPTIONS=" -Xmx2048m -Xms256m -

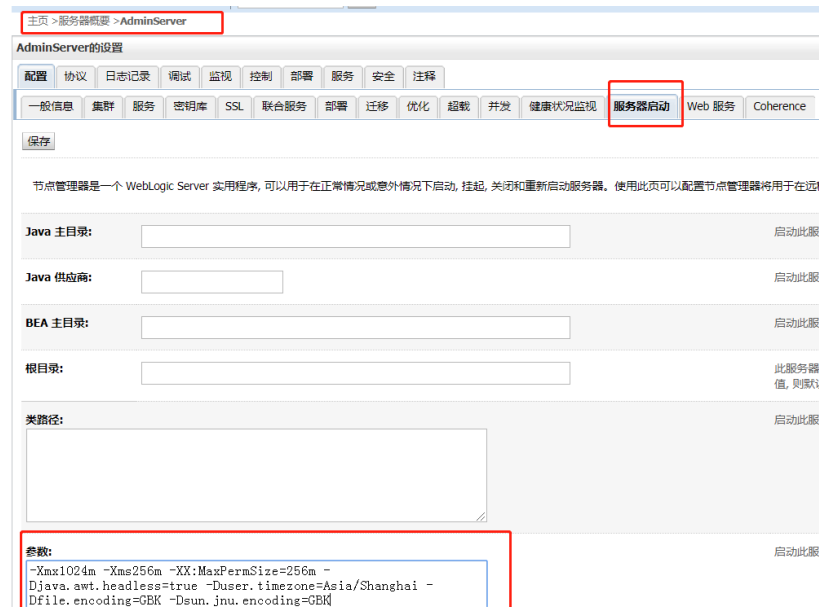
XX:MaxMetaspaceSize=256m -Djava.awt.headless=true -Duser.timezone=Asia/
Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK

其中，设置内存不能低于上述值。

集群环境：

如下图，进入weblogic控制台，对每一个节点（如server1、server2）进行内存设置，设置内存不能低于图示值。

图 3-355 内存设置



3. Websphere 8.5（其它版本大致不变）：

Websphere默认内存最大内存为256M，在大用户量情况下容易出现内存不够的情况，所以需要进行调整。在如下图所示堆内存地方进行调整，直接输入数字即可。

在JVM参数配置内存参数-Xmx1024m -Xms256m -XX:MaxMetaspaceSize=512m -Djava.awt.headless=true -Duser.timezone=Asia/Shanghai -Dfile.encoding=GBK -Dsun.jnu.encoding=GBK（数值不能低于此值）。

内存调整需根据服务器内存大小、操作系统版本、WEB服务器版本、JDK版本和实际使用情况进行调整，在64bit环境下建议分配8G-16G内存。

注意

某些JDK版本或操作系统限制最大堆内存，不支持分配超过较大内存，如果有充足的内存，可以多建立几个节点集群成员，逐一对每个成员做相应的修改。

图 3-356 应用程序服务器 1

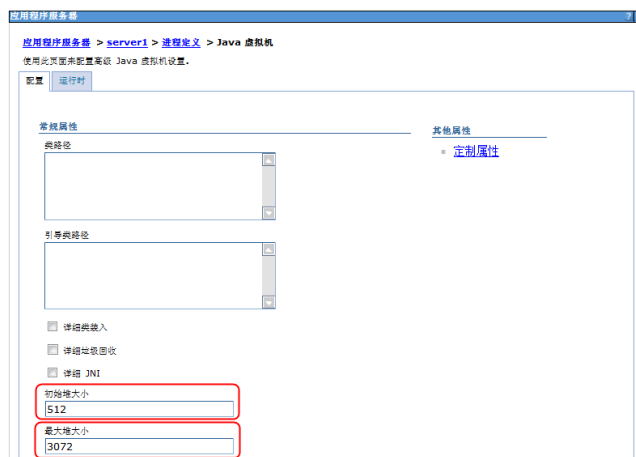
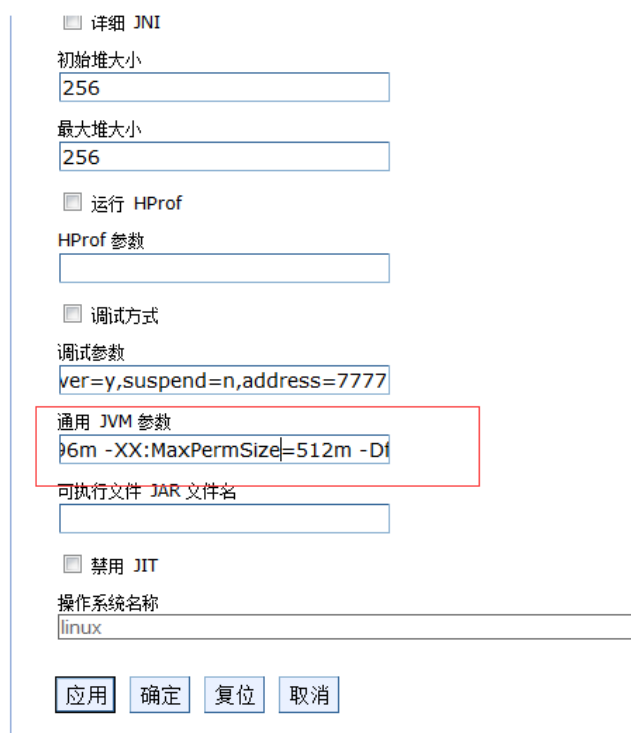


图 3-357 应用程序服务器 2



工作目录设置方法总结

1. 在war包中指定

ABI一系列产品在war包的WEB-INF\classes目录下的application.properties中指定，如下图位置

图 3-358 指定

```

1 #spring\u76f8\u5173\u914d\u7f6e
2 spring.main.banner-mode=off
3 spring.aop.proxy-target-class=true
4 spring.mvc.favicon.enabled=true
5
6 #windows\u5de5\u4f5c\u76ee\u5f55\u914d\u7f6e\u652f\u6301\\u548c/
7 #esen.abi.workdir=D:\\esendev\\workdir\\abi
8 #linux\u5de5\u4f5c\u76ee\u5f55\u914d\u7f6e
9 #esen.abi.workdir=/opt/workdir/abi
10 #esen.abi.workdir=/opt/workdir/abi
11
12 server.error.path=/eweb/error.do
13 server.error.includeStackTrace=ALWAYS
14 #server.systableprefix=ES31
15 spring.groovy.template.enabled=false
16
17 #java\u73af\u5883\u53d8\u91cf\u4e2d\u5de5\u4f5c\u76ee\u5f55\u7684\u914d\u7f6e\u992e\u540d
18 workdir.javaopt=esen.abi.workdir
19 #windows\u6ce8\u518c\u86f0\u5de5\u4f5c\u76ee\u5f55\u914d\u7f6e\u992e\u540d
20 workdir.winreg=com.esensoft/workdir/abi/workdir
21 #\u914d\u7f6e\u4e3a\u4e2a\u4ea7\u54c1id\u52a0\u4e2d\u7248\u672c\u53f7\u7f00\u5982\u540c\u4e00\u4ea7\u54c1\u90e8\u7f
22 spring.jmx.default-domain=ABI52
23
24 #ecas\u5ba2\u6237\u7aef\u914d\u7f6e
25 #esen.ecas.casserverurl=http://localhost:9090/esenface
26 #esen.ecas.casserverlogouturl=eacl/login.do?action=logout
27
28 #server.context-path=/abi
29 #spring-mvc\u73af\u5883\u53d8\u91cf
30 spring.thymeleaf.mode=LEGACYHTML5
31 spring.thymeleaf.cache=false
    
```

2. 在JVM参数中设置
工作目录参数为-Desen.abi.workdir，可以在JVM参数中设置，形式如下：
-Desen.abi.workdir=/opt/workdir
3. 在产品注册界面设置

图 3-359 设置



三种方法优先级由高到低：war包指定 > JVM参数指定 > 注册界面指定。

表 3-70 数据库权限分配

数据库	分配权限
Oracle	connect、resource、create any view(备注：oracle12c及以上需要 alter user xxx quota unlimited on users;)

4 附录

常见问题

1. 亿信官网提供帮助文档，汇聚快速入门和复杂场景应用分解说明，使产品更易用
ABI一站式数据分析平台：<https://help.esensoft.com/doc-view-488.html>
睿治数据治理平台：<https://help.esensoft.com/doc-view-692.html>
2. 公开社区活动，数千专业工程师入驻，7*24小时全天候服务，9分钟内社区快速响应，
累积解决5万+问题 <https://bbs.esensoft.com>。

5 修订记录

发布日期	修订记录
2024-04-18	规范词、敏感词专项处理，章节优化
2023-10-16	第一次正式发布。