

硕 士 研 究 生 读 书 报 告



题目 基于Jenkins利用IBM UrbanCode Deploy实现WebSphere Application Server部署自动化

作者姓名 李潇颖

作者学号 21651135

指导教师 李启雷

学科专业 移动互联网与游戏开发

所在学院 软件学院

提交日期 二○一七 年 1月

Using UrbanCode Deploy to implement WebSphere Application Server Automated Deployment based on Jenkins

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Qilei Li

By

XiaoYing Li

Zhejiang University, P.R. China

2017

摘要

随着应用程序越来越复杂，部署包含多个组件的应用程序极具挑战性。手动部署即慢又容易出错，自动化构建和部署的实施迫在眉睫。基于Jenkins利用UrbanCode Deploy实现WebSphere Application Server自动化部署解决了这样的问题。

Jenkins与uDeploy连接的插件已经被开发为集成提供了技术基础；uDeploy提供图形化流程编辑器能够可视化的创建部署流程，并且可以将相同的应用程序部署到多个环境，提高了效率同时降低了部署错误；WebSphere为 SOA（面向服务架构）环境提供软件，以实现动态的、互联的业务流程，为所有业务情形提供高度有效的应用程序基础架构。

本文主要介绍了这三个平台的特性，并且以一个案例详细说明了如何通过Jenkins启动uDeploy中配置好的部署流程，通过插件识别WebSphere应用的配置概要，从而将项目发布到WebSphere的过程。

**关键字：Jenkins ， WebSphere ，**uDeploy **，自动化构建**

Abstract

As applications become more complex, deploying applications that contain multiple components can be more challenging. Deploying manually has cause many problems, So it is time to deploy applications automatically. Using UrbanCode Deploy to implement WebSphere Application Server Automated Deployment based on Jenkins which solves those problems.

Jenkins ‘plug-in with uDeploy have been developed which provide the technical foundation for integration. uDeploy provides a graphical process editor that visualizes the creation of a deployment process and can deploy the same application to multiple environments, improving efficiency while reducing deployment errors; WebSphere provides software for an SOA (Service Oriented Architecture) environment to enable dynamic, interconnected business processes to provide a highly efficient application infrastructure for all business scenarios.

This article describes the characteristics of these three platforms, and a detailed description of the case how use Jenkins to start the deployed deployment process in uDeploy, how through the plug-in to identify WebSphere application configuration profiles. Eventually publish the project to WebSphere automatically.

**Keyword: Jenkins, WebSphere, uDeploy, Build Automatically**

1引言

持续集成是软件开发的一个重要实践[1]，持续集成的引入降低了软件开发的风险，提高了软件开发效率，提高代码质量，使得项目管理人员能更好的了解项目的开发进度。每次集成过程包括编译、构建、部署、测试、发布整个过程是自动化实现的。

基于Jenkins利用UrbanCode Deploy实现WebSphere部署自动化能够最大程度的实现自动化部署的优点：首先建立单一的部署来源，对于拥有不同团队的项目制定一个单一的、可信任的、实时更新的部署来源能过很好的实现可视和跟踪；其次能过让令人痛苦的的手工步骤自动化起来，传统的手动部署带来了许多风险，例如如果步骤没写特别详细，对新手来说存在着很大的不确定性；最重要能够实现在不同环境中保持部署的一致性，对部署环境可以进行严格的控制等。

1. 平台简介

2.1 Jenkins

Jenkins[2][3] 原名Hudson，是一款用java语言开发的开源持续集成软件，它是一款服务器平台类型软件，支持master／slave分布式的build。Jenkins在作为持续集成的工具选择上有如下特点：

1.支持多语言，插件资源丰富。由于Jenkins可以支持多技术领域的技术和编程语言，所以Jenkins在Githhub上有个庞大的项目成员和公共仓库的积累。

2.易用性高、过程可视化。Jenkins之所以被广泛接受的原因在于：Jenkins是简单易用的，清晰明了的用户界面，功能强大学习成本低。

3.版本更新快，功能强大。Jenkins能够满足需要持续集成解决实际问题的需求。它在互联网上有成百上千个开源的插件，病情不断更新，插件几乎涵盖了所有的版本控制系统，编译工具，代码检查工具等，安装快捷简单。并且具有良好的扩展性，可以自己编写相关的插件，个性化定制项目的构建参数和构建过程。

Jenkins主要由两个部分构成部分构成：一个部分是可以独立运行的java程序,用来负责构建；另外一部分是web程序，可以负责实现多任务并行化处理的宏调用，通过循环的方式执行各个构建周期。如图2.1所示：

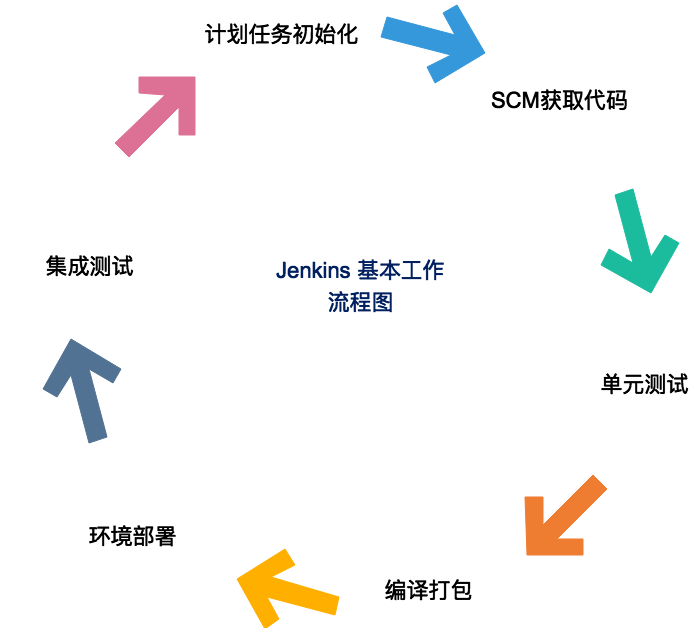


图2.1 Jenkins 基本工作流程图

* 1. WebSphere Application Server

WebSphere Application Server（WAS）[4][5]为 SOA（面向服务架构）环境提供软件，以实现动态的、互联的业务流程，为所有业务情形提供高度有效的应用程序基础架构如图[6]2.2。

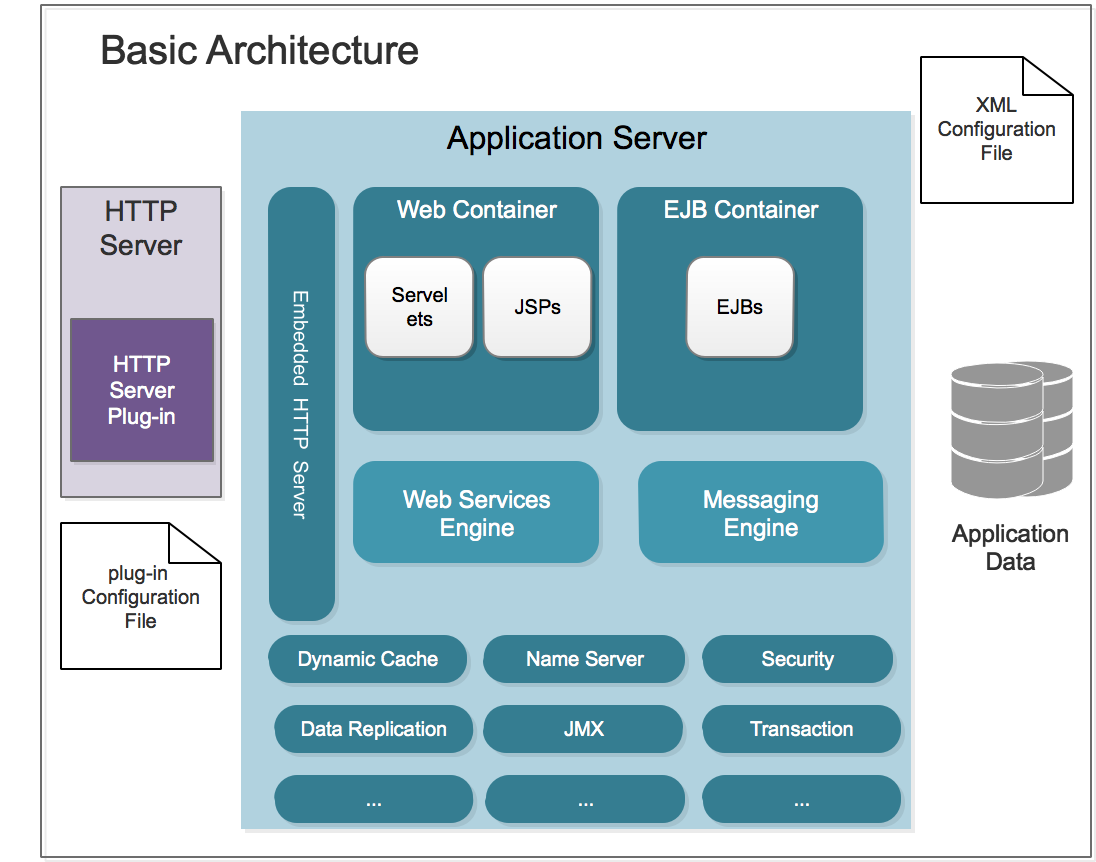


图2.2 WebSphere Application Server 架构图

WAS是 IBM 的应用程序和集成软件平台，包含所有必要的中间件基础架构（包括服务器、服务和工具），这些基础架构是创建、部署、运行和持续监视企业级 Web 应用程序和跨平台、跨产品的解决方案所必需的。WAS具有以下特性：完全基于标准，以服务器为中心，可伸缩，能利用已有的核心系统，快速部署和易用。

WAS遵守 Java EE 6 规范，使用这些规范作为它支持的编程模型之一。从业务关键型应用程序和重要企业级应用程序到最小的部门级应用程序，WebSphere Application Server 都提供了最高的可靠性、可用性、安全性和可伸缩性水平。

* 1. IBM UrbanCode Deploy

IBM UrbanCode Deploy[7]（简称 uDeploy）是 IBM 的应用自动化部署工具。该工具基于一个完善的应用部署自动化管理信息模型，提供可视化的部署逻辑设计手段，并通过远程代理技术，实现对复杂应用在不同环境下的自动化部署。该工具功能组件[8]参考如图2.3所示：



图2.3 uDeploy组件图

uDeploy作为自动化构建的工具具有以下特点：流程自动化，可以使用uDeploy图形化流程编辑器可以采用拖拽的方式创建组件流程，避免了编程的繁琐，并且流程具有良好的复用行；环境和配置管理，每个应用程序都包含可以部署它的环境的模型。环境包含1个或者多个部署目标。uDeploy 可以针对每个环节有不同的配置设置。uDeploy中强大的配置管理功能支持环境配置变更和应用部署；构件存储库，uDeploy内置的构件存储库CodeStation为以部署或要求部署的文件提供安全的存储；安全、审批和通知，uDeploy提供了细粒度的可定制用户权限，保障了部署文件的安全可靠传输。

IBM UrbanCode Deploy 作为自动化部署工具的一个抽象过程如下图2.4所示：

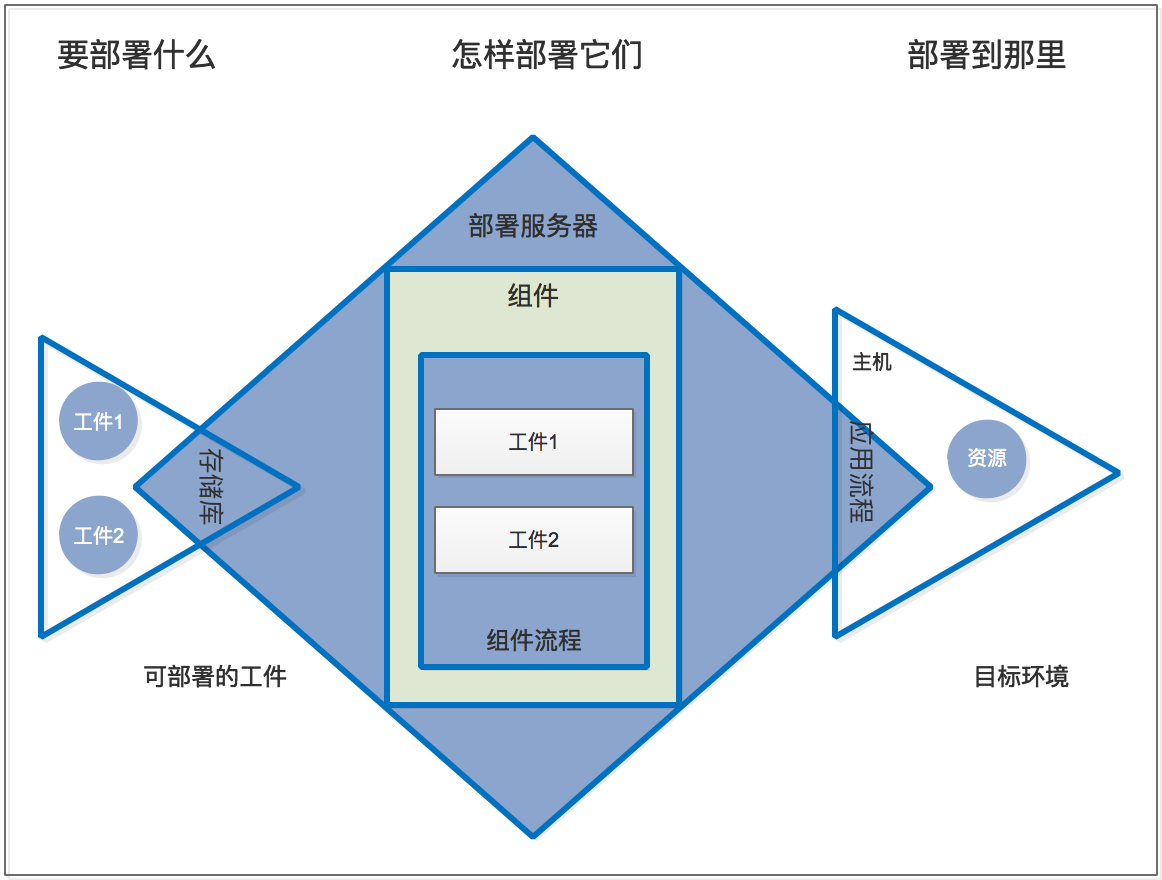


图2.4 uDeploy 部署抽象过程图

3 软件的配置

3.1 Jenkins的配置

3.1.1 Jenkins的安装

Jenkins的安装比较便捷，到官网上下载Jenkins.war包直接运行即可，但是前提是需要安装和配置java环境。

3.1.2 Jenkins插件安装

IBM UrbanCode Deploy针对Jenkins开发了一款插件，这款插件能够使得Jenkins可以触发uDeploy实现项目的自动构件过程（插件可以在<https://developer.ibm.com/urbancode/plugin/jenkins/>该链接下载）。

打开Jenkins面板“系统管理”->“插件管理”->“高级”->“上传插件”将下载好的 “IBM UrbanCode Deploy Plugin” [9]插件上传，并且安装到Jenkins平台上。

安装“Git Plugin”和“GitHub Plugin”，git插件的安装后Jenkins可以直接从git上获取sourcecode，同时可以指定build该项目的脚本文件所在位置。打开Jenkins面板：“系统管理”->“插件管理”->“可选插件”中选择“Git Plugin”和“GitHub Plugin”这个两个插件进行安装；插件需要安装成功后在“已安装”模块可以看到如图3.1所示插件：



图3.1 Jenkins 所需插件图

* 1. WebSphere Application Server的配置

3.2.1 WebSphere Application Server的安装

WAS可以在IBM产品的官网可以下载到试用版，它到安装依赖于IBM Installation Manager。所以先下载IBM Installation Manager，基于此道安装步骤较为简单，基本上点击“下一步”即可。

3.2.2 WebSphere Application Server的概要

WAS的安装过程只是简单地放置一组运行时进程必需的核心产品文件。在安装之后，需要创建一个或多个用来定义运行时的概要，以获得一个可工作的系统。这些概要所定义的运行时组件将共享核心文件。每个应用程序服务器都是在一个单独的单元和节点中定义的。定义一个自定义概要，用来创建一个将与单元联合的空白节点。在联合之后，可以在部署管理器的管理控制台中创建应用程序服务器和集群，以进一步配置该节点。如图3.2所示：

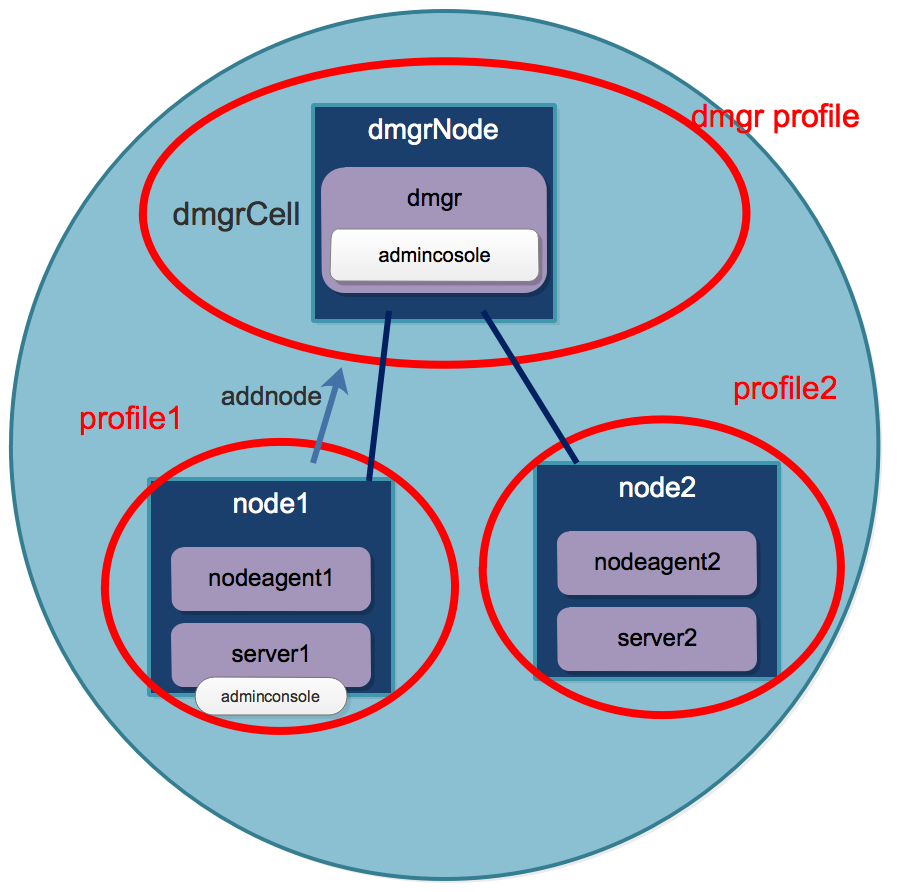


图3.2 WAS部署图

基本结构:单元(cell) ——> 多个节点(node) ——> 多个服务(server) ——> 多个应用(app)[10]

单元Cell:整个分布式网络中一个或多个节点的逻辑分组，是一个管理上的概念, 是管理员将节点间逻辑关联起来的实现方法, 可以将单元看作是WAS最大的作用域

节点Node:是受管服务器(Server)的逻辑分组, 节点通常与具有唯一IP主机地址的逻辑或物理计算机系统对应, 节点不能跨多台计算机，它也是管理使用上的概念

服务Server: Application Server Instance，即实际部署应用的地方，一个节点通常只能使用一个Server

概要文件Profile:定义一个独立应用程序服务器(Server)的运行时环境, 包括服务器在运行时环境中处理的所有文件。与节点是同一事物，它是物理上存在的

节点代理:Node Agent，它将管理请求路由至服务器的管理代理程序。Node Agent是是服务器, 是一个管理代理程序, 并不涉及应用程序服务功能, Node Agent 进程在每个受管节点上运行, 并专门执行特定于节点的管理功能, 如服务器进程监视、配置同步、文件传输和请求路由。

3.2.3 WAS的拓扑结构

将以应用部署到名为Cluster01的WAS集群上为例，它在名为cell01的WebSphere单元中，如图3.2所示。Cell01涵盖主机PC-2，并且它有两个节点：Node2和Node3，Node2上有一台服务器Server1，Node3上有也有一台服务器Sever2。所有的服务器都属于Cluster01：

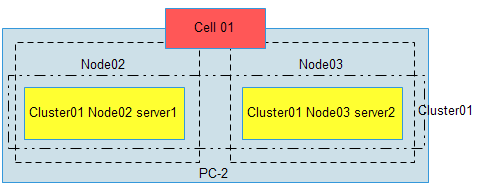


图3.2 WAS 应用程序拓扑图

集群拓扑如图3.3所示的WebSphere Integrated Solutions Console中以层次化的方式显示。



图 3.3 WAS 层次图

* 1. IBM UrbanCode Deploy的配置

3.3.1 UrbanCode Deploy的server和 agent的安装

uDeploy的server和 agent的安装过程目前支持命令行的形式来进行安装。进入uDeploy的server和 agent的目录下启动“install-server.bat”、“install-agent.bat”文件，然后根据命令行的提示，一步步往下安装即可。

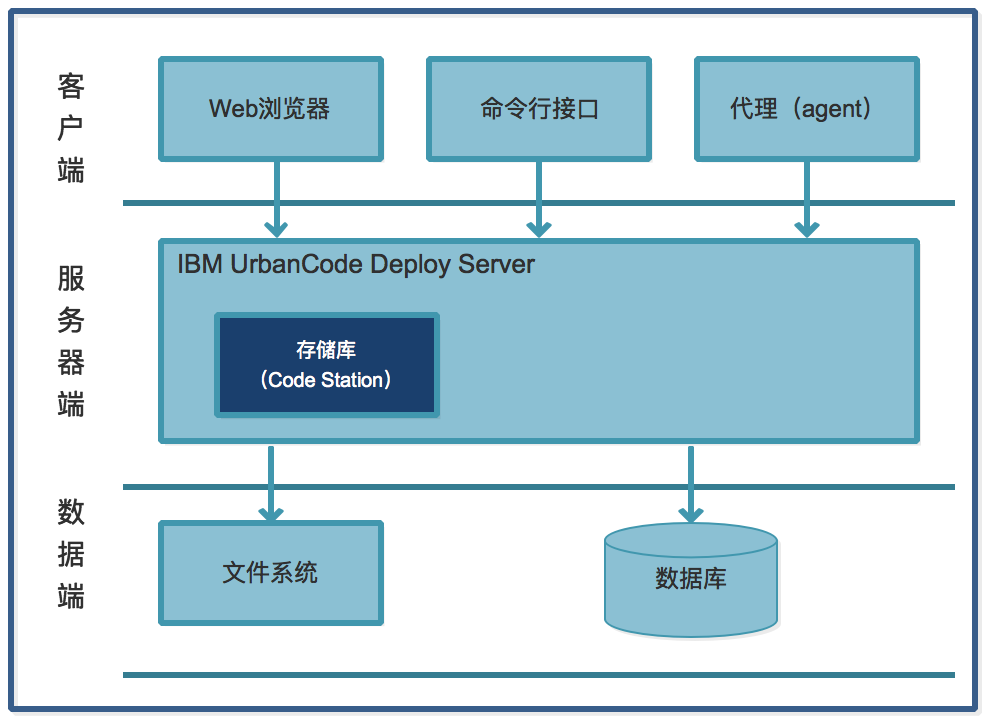
IBM UrbanCode Deploy Server 提供了产品的核心功能，包括用户使用界面、组件和应用的配置、流程设计引擎、权限管理等。其中的存储库（Code Station）是管理和存储可部署工件的服务。Agent（代理）是 IBM UrbanCode Deploy 产品架构中的一个重要组成部分，由于它可以和服务器进行通信，因此也算作一种客户端。uDeploy的产品架构图如下：

图 3.4 uDeploy 架构图

* + 1. UrbanCode Deploy插件的安装

为了实现jenkins、UrbanCode Deploy和WebSphere三者的联合工作，需要为uDeploy安装针对第三方工具的接口，称之为专用领域插件[9][11]（Domain-Specific Plug-ins），使用was插件就提供了多种对was服务器、应用对操作手段，如停止或启动服务器，安装货卸载应用等。

这些插件在uDeploy的插件官网上都可以下载到，先下载到本地，然后在uDeploy的server端控制面板中进入关于“系统管理”模块选择本地存储的相应的插件文件进行安装。

4 以具体案例描述联合部署的实现

在本次实践中我们通过Jenkins触发uDeploy实施项目的自动化构建，然后将项目部署到WAS上[12]。

4.1 Jenkins配置

step1：插件上传安装成功后，可以到“系统设置”模块配置该插件的一些信息，如图4.1所示：

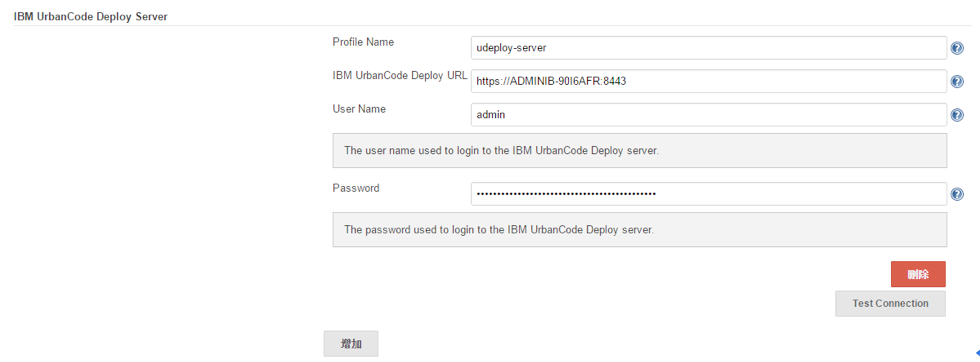
:

图4.1 Jenkins 配置图

Profile Name：通常取默认值“udeploy server”

Username和password指的是登入uDeploy的账号和密码。

Step2：在Jenkins新建一个项目并且在“源代码管理”模块，填入git的相关参数（Reprository url地址）

Step3：“构建”模块选择“Invoke Ant”参数配置如下，填入build.xml文件所在路径，整个Jenkins配置过程就完成了

4.2利用IBM UrbanCode uDeploy 实现WebSphere自动化部署

**4.2.1.准备要部署的应用程序**

我们将部署的应用程序是JPetStore应用程序，在演示中常常用到的一个开源应用程序。我们将使用到的是JPetStore的一个war组件，它将被部署到一个集群上。

war文件在Jenkins调用build.xml时，写好的build文件将sourcecode打包成了war包，在uDeploy中指定好war包所在位置，然后在进行自动部署。

**4.2.2. 准备使用到软、硬**

PC-1安装了Windows8操作系统、uDeploy6.2

PC-2安装了Linux操作系统、uDeploy-agent

如图4.2所示：

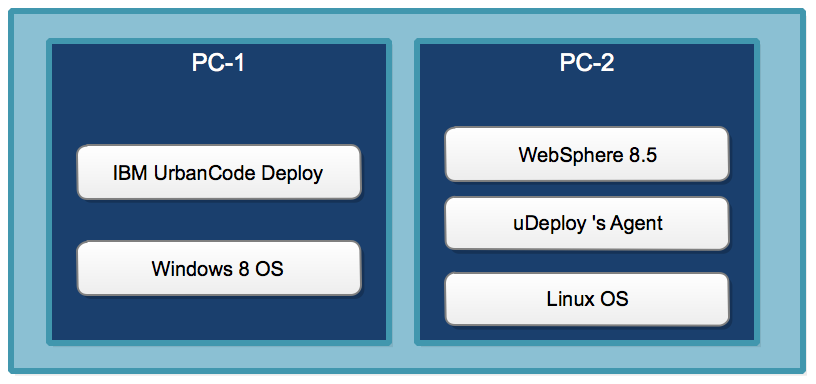


图4.2 设备图

**4.2.3 WebSphere的结构[13]**

我们将应用部署到名为server1的一个WAS的server上，它在名为cell01的WebSphere单元中。如图4.3所示：

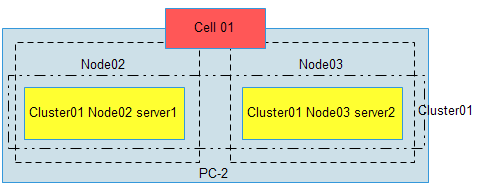


图4.3

集群的构建步骤如下：

①建立一个Deployment Manager节点，取名为Dmgr01

执行以下语句

进入： /opt/IBM/WebSphere/AppServer/bin

键入：./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere /AppServer/

profileTemplates/dmgr -profileName Dmgr01进行创建

②建立第一个应用单元，取名为AppSrv02

执行以下语句

进入：/opt/IBM/WebSphere/AppServer/bin目录

键入：./manageprofiles.sh -create -templatePath /opt/IBM/WebSphere /AppServer/

profileTemplates/default -profileName AppSrv02 进行创建

③建立第二个应用单元，取名为AppSrv03

④联合AppSrv02节点入Dmgr01

进入：

/opt/IBM/WebSphere/AppServer/profiles/AppSrv02/bin目录

执行：

addNode.sh localhost -username admin -password admin 进行节点联合

⑤联合AppSrv03节点入Dmgr01

⑥启动：用startManager启动dmgr01

用startNode启动node

⑥打开浏览器输入：

<https://localhost:9043/ibm/console/logon.jsp>

进入Dmgr01的控制台界面（9043为Administrative secure port）

⑦创建WebSphere的集群，过程如图4.4至图 4.7所示：

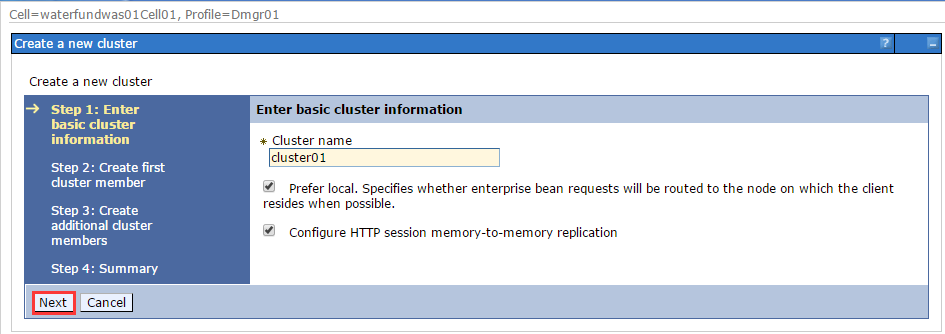


图4.4

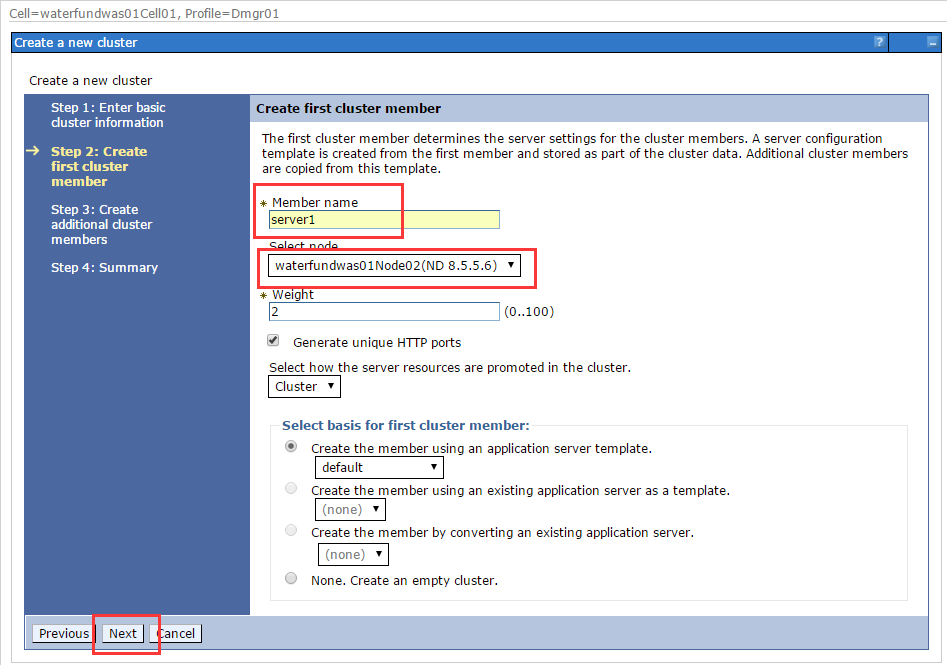


图4.5

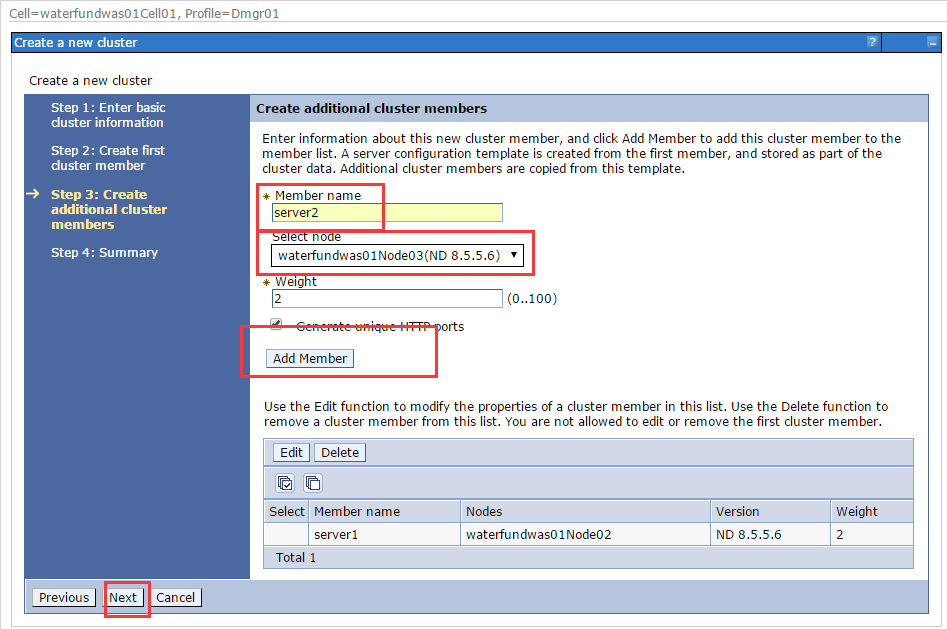


图4.6

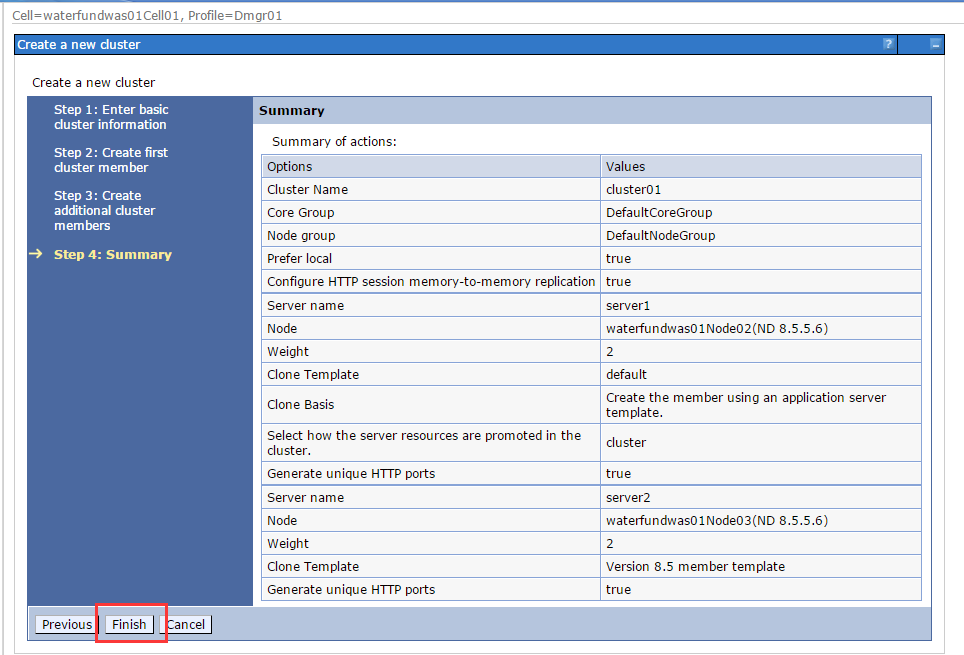


图4.7

**4.2.4 uDeploy准备工作[14]**

在PC-1上安装uDeploy的Sever端用来将JPetStore应用程序部署到server1上；

导入相关的插件:WAS-Configure、WAS-Deploy等用来构建组件流程和自动发现WAS的资源；

创建Agent，该Agent位于PC-2上。进入Resourses面板，点击“Install New Agent”选择以“SSH”的方式创建Agent，如图4.8所示填入相应的参数。点击“Agent Installation Properties”为Agent-1配置相应参数，如图4.9所示。最后点击保存，Agent-1创建完成，图4.10所示。

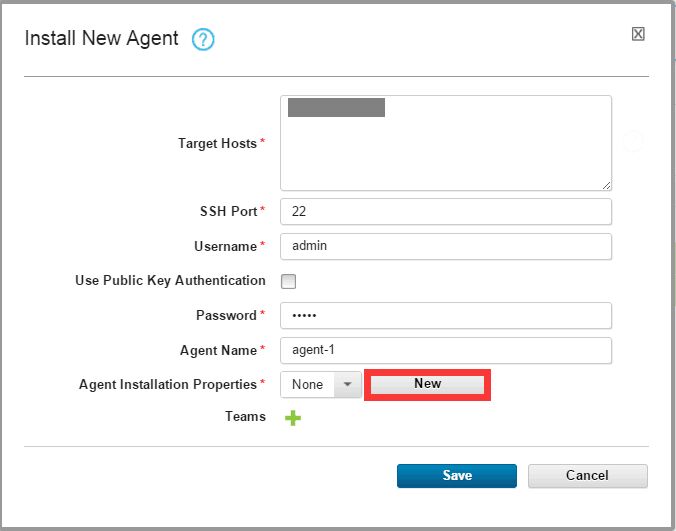


图4.8

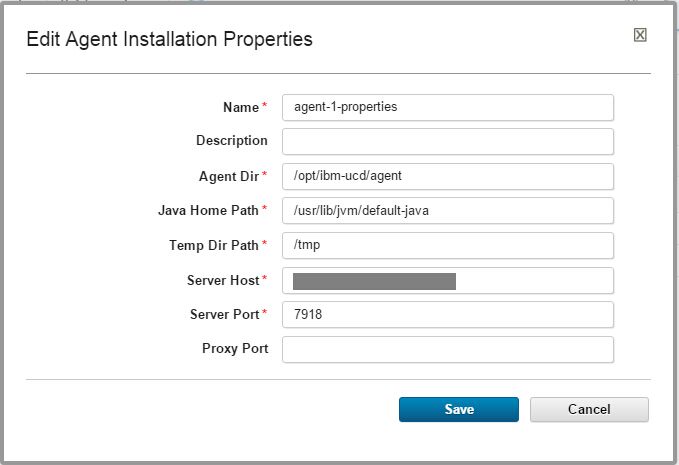


图4.9

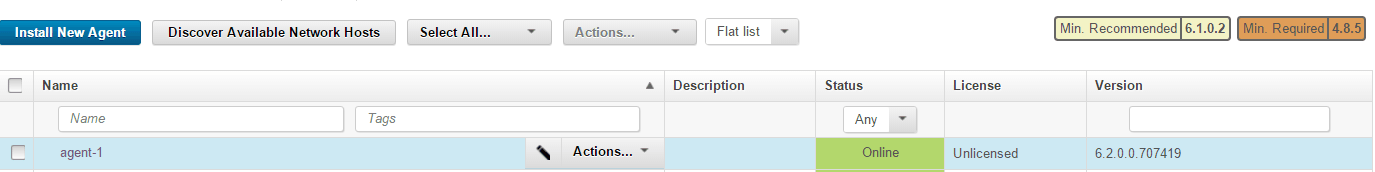


图4.10

**4.2.5 uDeploy 和 WebSphere的联合部署**

**WebSphere的自动发现**

1. 自动发现是uDeploy用于通过Agent在主机上发现特定技术的一个机制。当发现WAS时，可使用自动匹配来创建部署目标的一个模型。WebSphere的自动发现和自动配置是WAS插件的一部分。利用它们可以获知在自动部署到WAS中的任何资源。
2. 我们将演示WebSphere的资源是如何被自动发现和启动的：
3. ①配置Agent-1，在Agent的属性中添加“wsadmin.path”参数，如图4.11所示：
4. ②创建“顶级组”was-test，在资源属性中添加“websphere.profilePath”参数，如图4.12所示：

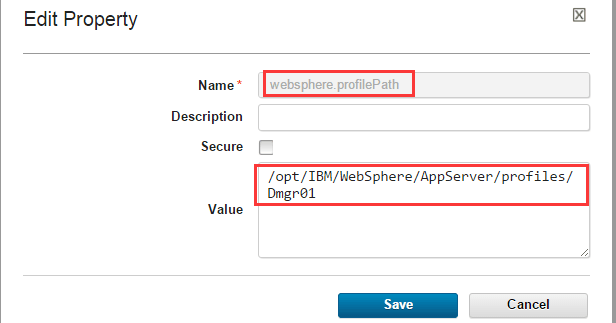
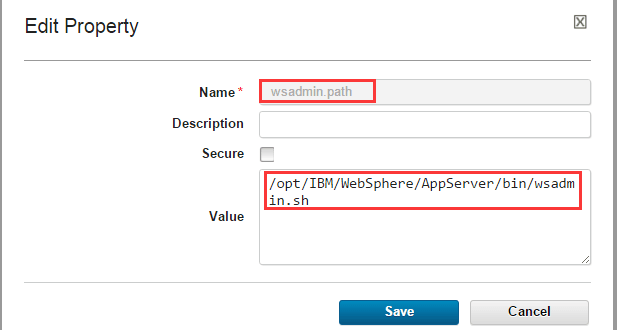


图4.11 图4.12

③WebSPhere的自动发现

返回Resource-Resource Tree面板，等待30s后更新界面，如图4.13所示：

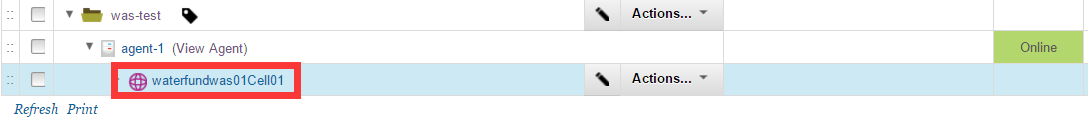


图4.13

点击waterfundwas01Node01cell栏对应的“编辑”按钮来编辑资源。如图4.14所示：

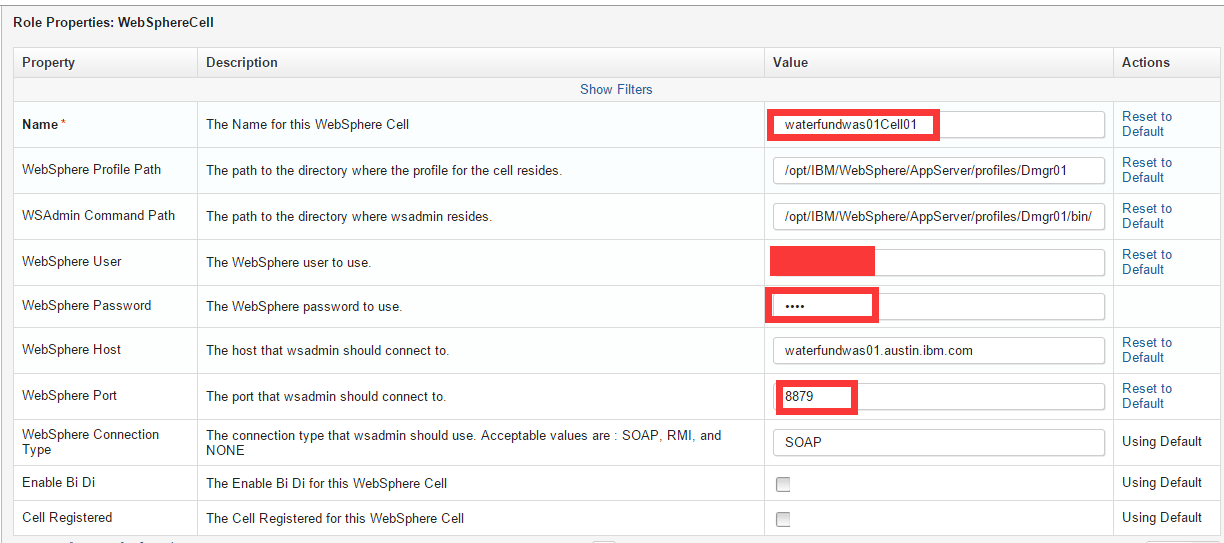


图4.14

填入红色框标记的参数，保存。

点击waterfundwas01cell01栏对应的“Action”按钮，然后选择“Configure using WebSphere Topology Discovery”等待其自动自动配置。更新页面，如图4.15所示

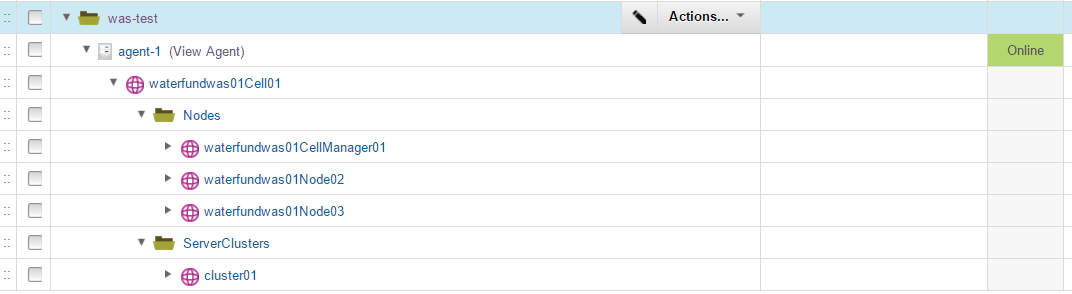


图4.15

在自动配置的过程中，Agent遍历单元，映射单元的拓扑结构并捕获数据。该信息被转发到uDeploy，用于生产代表单元的节点。

**2、创建组件及组件流程**

创建组件was-componet，采用Jenkins的方式来获取JPetStore.war组件。

创建组件流程，在uDeploy中组件流程是一系列用户定义的步骤，执行包含组件的操作。可以使用uDeploy可视的拖放式的流程编辑器来创建组件的流成。为了获取到JPetStore.war文件并且部署它，需要执行下列的步骤：

①从组件库中下载最新的组件。

②检查应用程序是否正在执行？如果是，停止它。

③卸载应用程序，无论成功与否均往下执行（如果一开始为未安装则该步骤执行失败）

④安装应用程序

⑤等待应用程序

⑥启动应用程序

具体的步骤如图4.16所示：

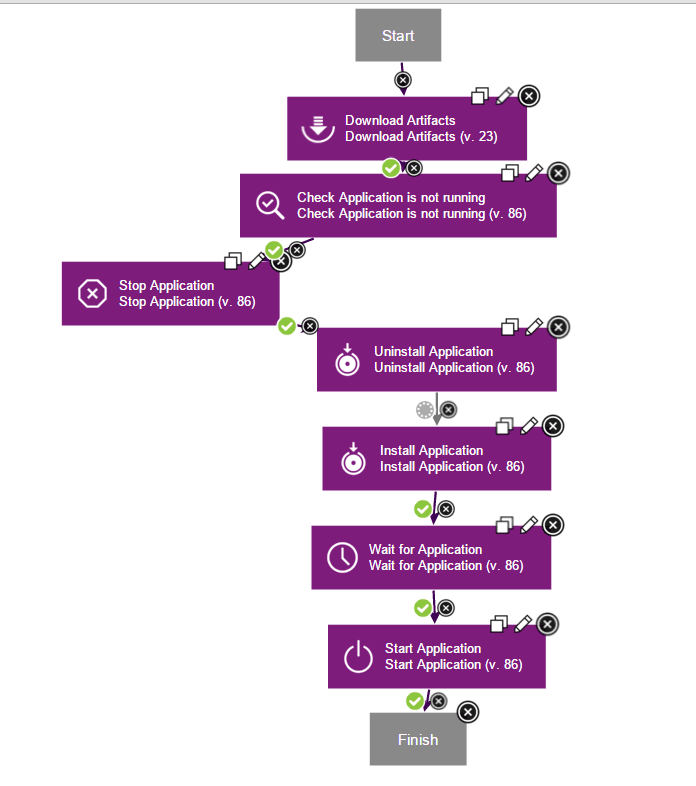


图4.16

大部分步骤有WebSphere插件提供。插件步骤不要求安装任何环境特定信息。因为自动匹配已手机了WAS单元信息，将它关联到可以在其上面运行改库存的资源。大部分步骤的参数保持默认值即可，Install Application的参数如图4.17：

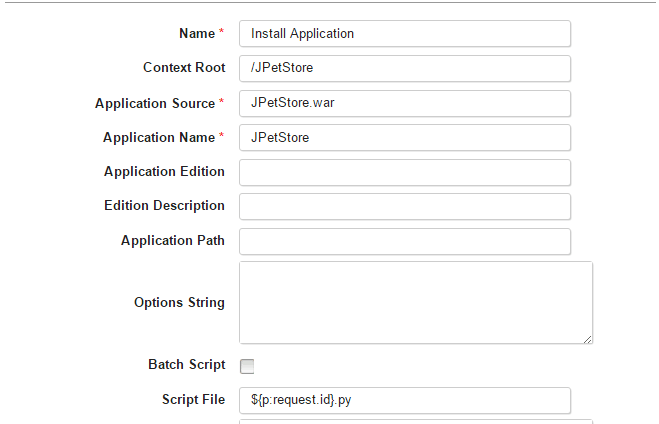


图4.17

1. **创建应用程序及其流程**

uDeploy应用程序流让我们能够将多个组件的流程结合起来，创建协调的部署。应用程序流程是可以结构化的，这样组件流程可以以串\并行的方式运行。使用uDeploy可视化的拖放流程编辑器可以创建应用程序的流程（与创建组件流程相似）

创建一个名为was app的应用程序，将was-component添加为该应用程序的组件。应用程序流程的工作就是启动该组件，其流程步骤和参数如图4.18所示：

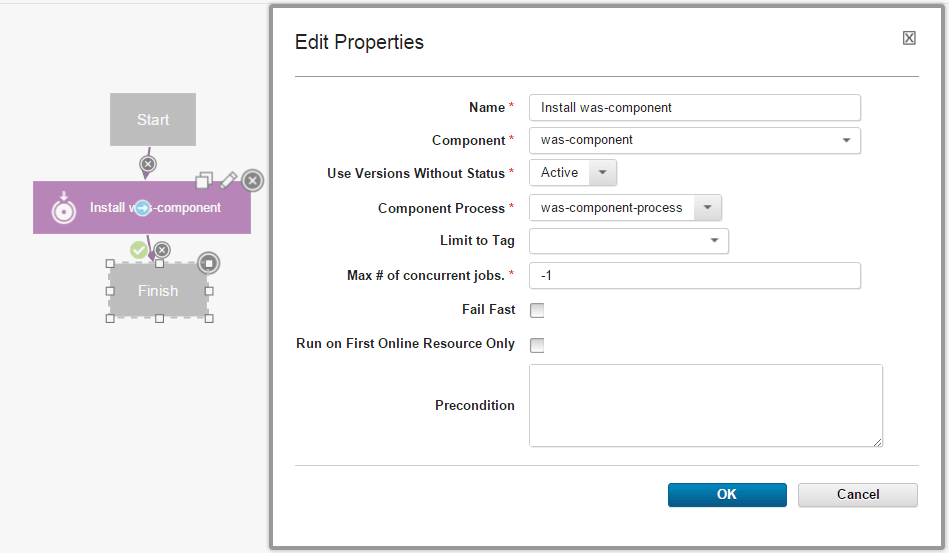


图4.18

1. **创建应用程序的环境**
2. 创建名为was app environment的环境
3. 为该环境添加基本资源，如图4.19所示

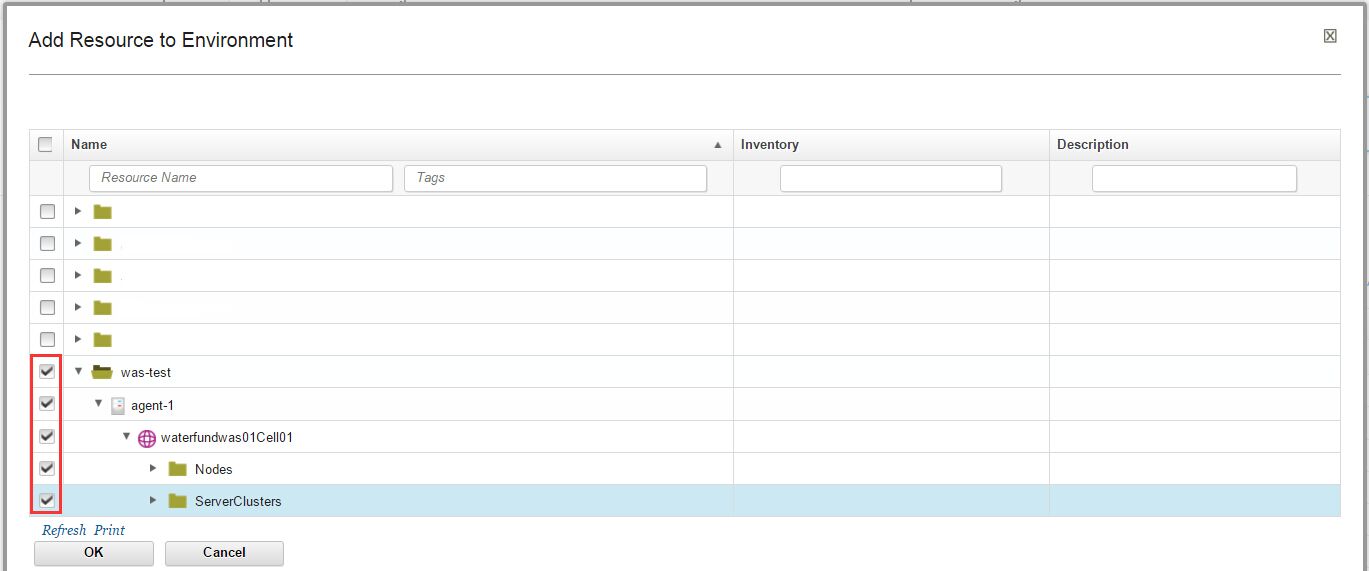


图4.19

1. 为应用程序添加组件“was-component”。
2. 环境由托管应用程序的资源组成，我们将部署到代表Cluster01的资源。在示例中唯一的资源是代表Cluster01的资源，因此在代表Cluster01的资源上执行was-component组件的流程。如图4.20所示，点击“Add Component”将组件was-component添加到该资源下

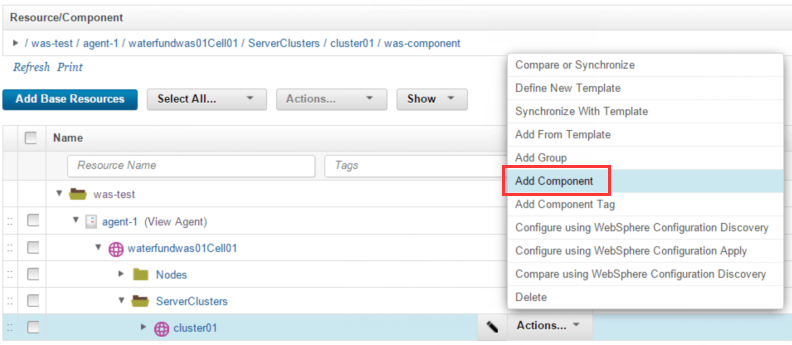


图4.20

1. **部署应用程序**

创建好了组件、应用程序以及应用程序的环境。已经具备了部署应用程序的条件。

进入Jenkins启动uDeploy服务器

在部署运行的时候，可以看到它随着应用程序流程的执行而推进，如图4.21：

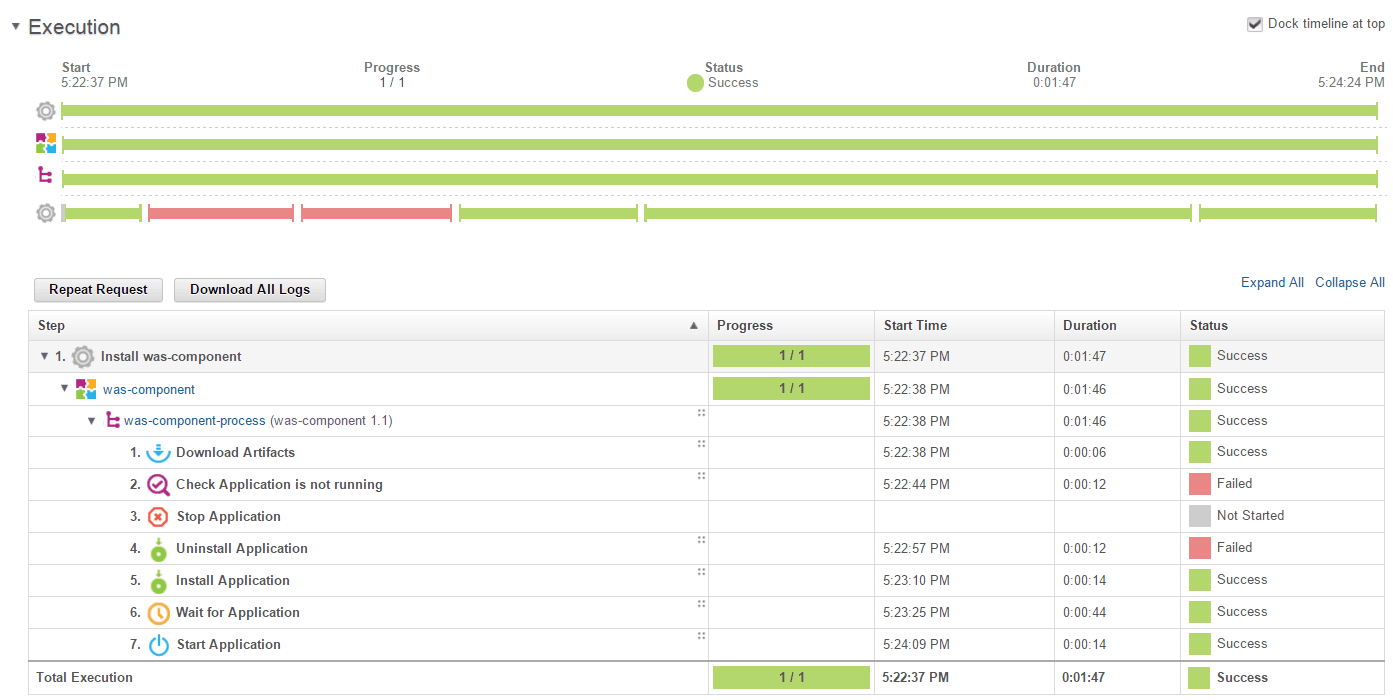


图4.21

步骤二“Check Application is not running”应用程序未执行，所以执行失败，跳过步骤三“Stop Application”。步骤四“Uninstall Application”因为应用程序未安装所以卸载失败，进入安装应用程序步骤。

可以点击来查看执行的具体输出。

**4.2.6 检验最终部署成果**

打开浏览器，输入：[https://hostname:9044/ibm/console/logon.jsp](https://localhost:9043/ibm/console/logon.jsp)

可以看到如图4.22至4.24所示

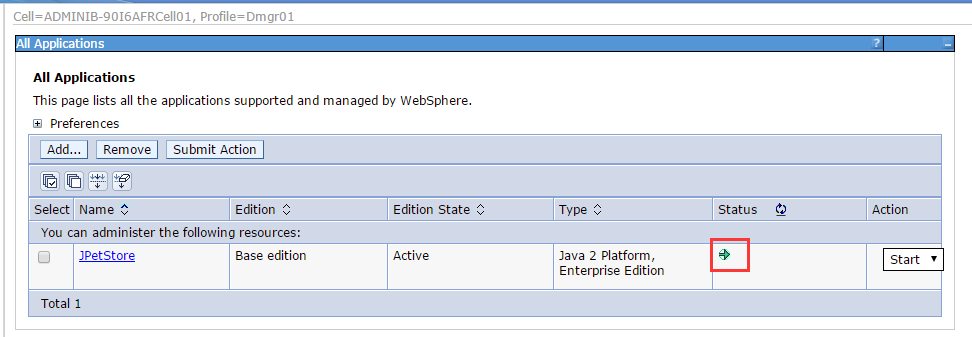


图4.23

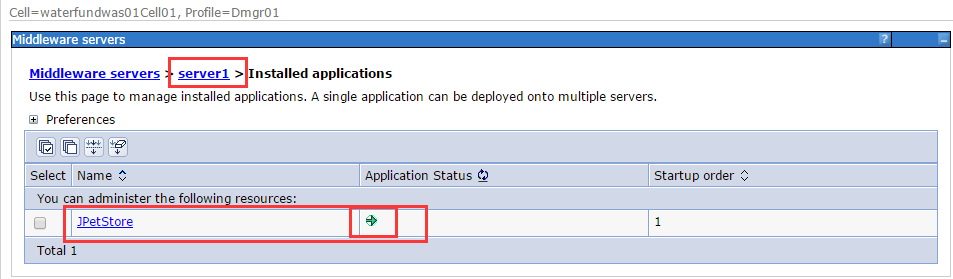


图4.24

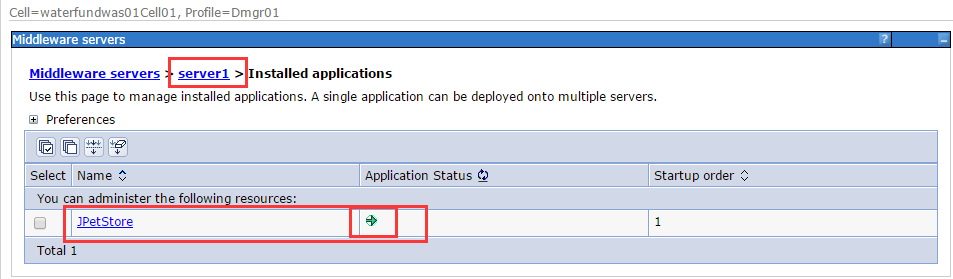


图4.25

已经成功的将应用程序部署到了WAS上了，如图4.26所示可以访问到所部署到程序



图4.26

参考文献

[1] 周莹, 欧中红, 李俊. 基于Jenkins的持续集成自动部署研究[J]. 计算机与数字工程, 2016(2):267-270.

[2] 卞孟春. 基于Jenkins的持续集成方案设计与实现[D]. 中国科学院大学(工程管理与信息技术学院), 2014.

[3] https://jenkins.io/index.html

[4] Keys Botzum. WebSphere Application Server V6.1: System Management and Configuration [J]. 2006.

[5]http://www.ibm.com/developerworks/cn/websphere/library/redbooks/was\_profileintro.html

[6] http://blog.chinaunix.net/uid-20274021-id-1969723.html

[7]IBM Developers. IBM UrbanCode Deploy 从开发到生产的应用部署自动化 [J].

[8] http://blog.csdn.net/hintcnuie/article/details/39375233

[9] https://developer.ibm.com/urbancode/plugins/udeploy/

[10] http://www.webspherechina.net/Article/58484

[11]http://www.ibm.com/developerworks/cn/rational/1403\_zhaojing\_urbancode2/

[12] IBM Developers. 利用uDeploy实现WebSphere部署自动化[J].

[13]http://www.ibm.com/support/knowledgecenter/zh/SSEQTP\_8.5.5/as\_ditamaps/was855\_welcome\_base\_dist\_iseries.html

[14]http://www.ibm.com/support/knowledgecenter/zh/SS4GSP\_6.1.0/com.ibm.udeploy.doc/topics/agent\_installremote.html