

jenkins+weblogic的分享

前言

为了方便我们更好的熟悉jenkins和weblogic建议，自己在外网搭建一个虚拟机，去学习

软件准备有：jdk,maven,git,jenkins,weblogic等

软件准备：VM虚拟机和一个centos7的环境，开始安装,我自己的电脑的软件目录都放到了/app/software

，每个安装的应用对应各自的文件夹，上传的时候安装xftp连接虚拟机上传安装包就可以

jenkins的相关配置

jenkins的工作流程介绍

jenkins主要的方便之处在：可以实时拉去最新仓库的代码，实时部署，实现代码更新。

jenkins的思考？有助于我们深刻的理解jenkins

既然是jenkins可以这么做，那么到底是如何做到的？jenkins主要是依据大量的插件，整合第三方的插件去完成的，所以，我们就会想，到底需要什么插件才能够支撑他去完成这些功能，以及需要配置什么？

答：jenkins既然可以拉代码，那么就需要指定拉代码的插件，

- 1.本次演示我的代码在gitee上放的，所以我需要的插件有gitee，拉代码，需要整合配置git
- 2.需要编译代码，所以jenkins需要去整合jdk,那么在jenkins的配置中需要配置jdk
- 3.在拉代码的过程中，肯定需要一些依赖，所以需要整合maven，所以需要配置maven
- 4.因为我们的实际的项目中需要发布到weblogic环境，所以我們也需要使用到发布的插件。

- Maven Integration：Maven 构建工具
- Publish Over SSH：整个工具，将来把Jenkins 打包好的 jar 上传到应用服务器上。
- Gitee：协助使用 Gitee 仓库。
- Deploy WebLogic Plugin：发布到weblogic的插件。

jenkins中项目的具体配置如下

1.插件的下载，这里举例一个就可以，别的插件安装步骤一样



安装/更新 插件中

准备

- Checking internet connectivity
- Checking update center connectivity
- java.net.HttpRetryException: Invalid response code (404) from URL:
https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/updatecenter.json?uctest
at hudson.model.UpdateCenter\$UpdateCenterConfiguration.testConnection(UpdateCenter.java:140)
at hudson.model.UpdateCenter\$UpdateCenterConfiguration.checkUpdateCenter(UpdateCenter.java:1636)
at hudson.model.UpdateCenter\$ConnectionCheckJob.run(UpdateCenter.java:1636)
at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at hudson.remoting.AtMostOneThreadExecutor\$Worker.run(AtMostOneThreadExecutor.java:121)
at java.lang.Thread.run(Thread.java:750)

Publish Over FTP

✓ 完成

Loading plugin extensions

✓ Success

出现这个就代表下载安装完成

➡ [返回首页](#)

(返回首页使用已经安装好的插件)

➡ ☐ 安装完成后重启Jenkins(空闲时)

🔍 192.168.1.188:8888/restart

1.在浏览器后面跟输入restart命令

你确定要重启 Jenkins 吗?

是

2.点击是，就可以完成重启

🔍 publish over ftp

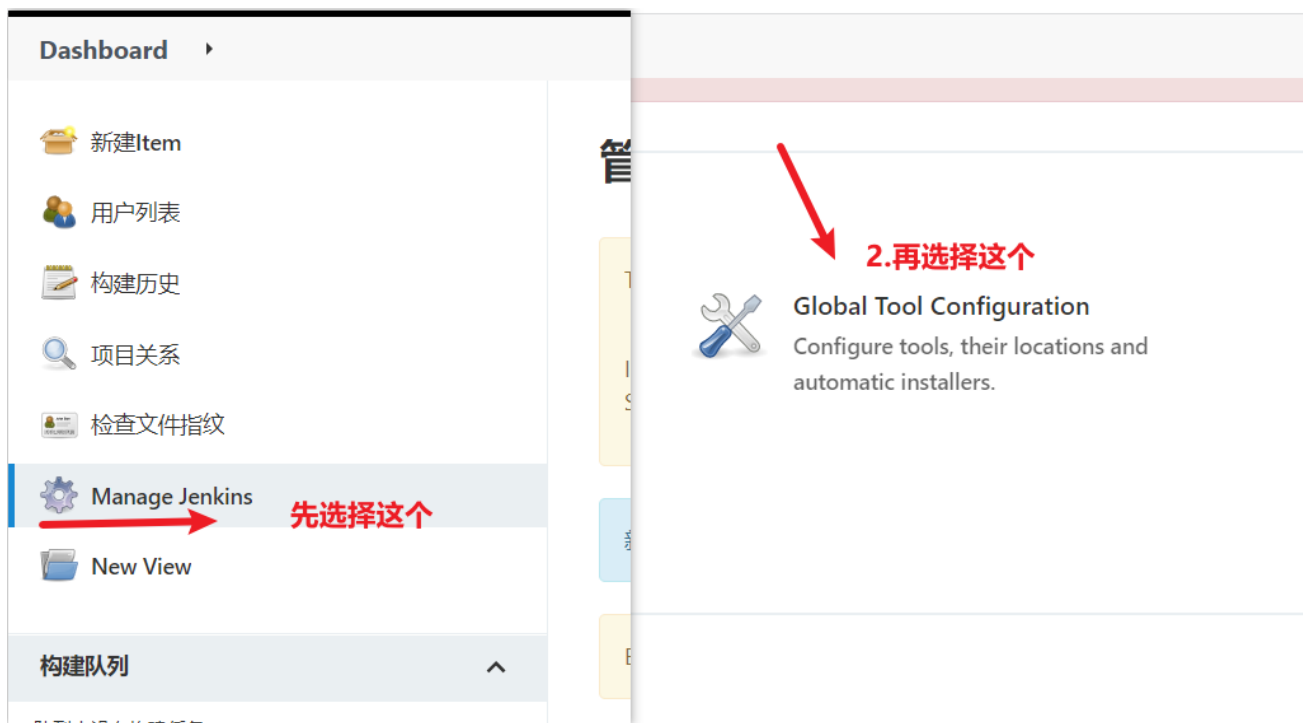
选择已安装，在已安装里面搜索刚才的软件，就可以看到

可更新	可选插件	已安装	高级	
启用	名称 ↓	版本	上一个安装	卸载
<input checked="" type="checkbox"/>	Infrastructure plugin for Publish Over X Send build artifacts some where.	0.22		<input type="button" value="卸载"/>
<input checked="" type="checkbox"/>	Publish Over FTP Send build artifacts over FTP	1.17		<input type="button" value="卸载"/>
<input checked="" type="checkbox"/>	Structs Plugin Library plugin for DSL plugins that need names for Jenkins objects.	324.va_f5d6774f3a_d		<input type="button" value="卸载"/>

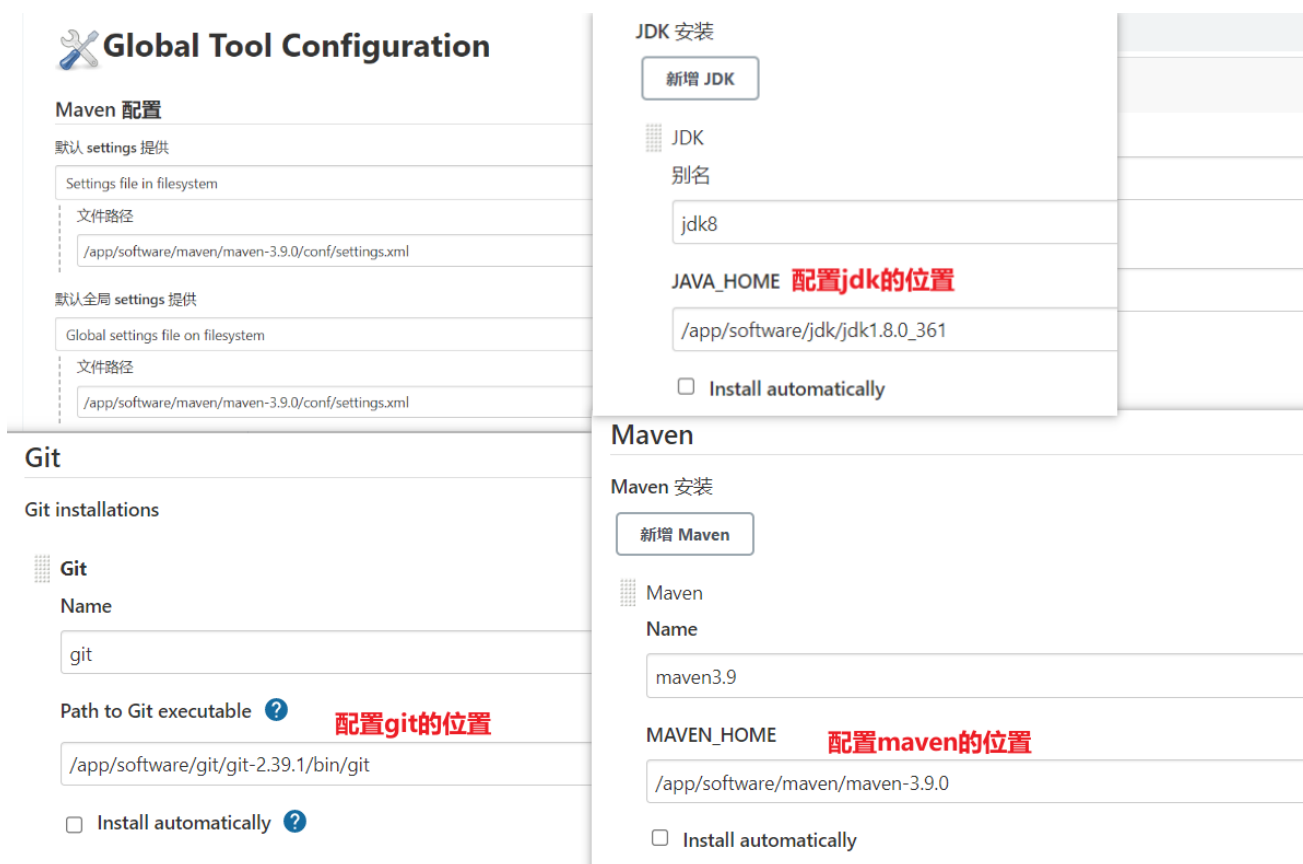
证明安装完成

jenkins的全局配置

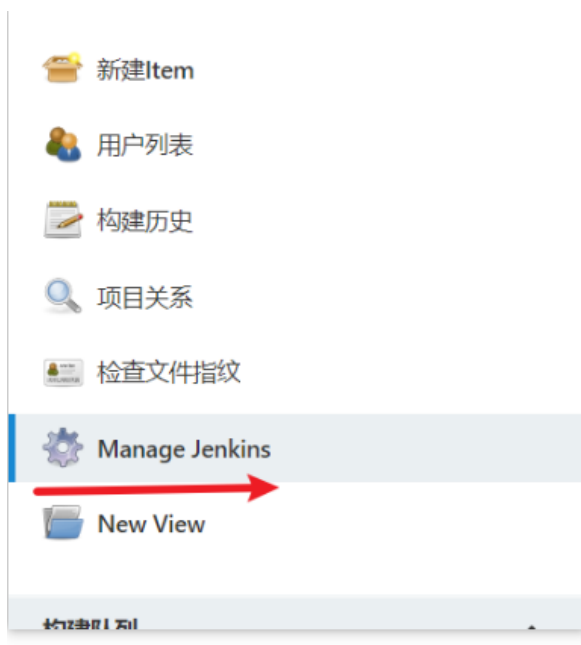
1.进入全局管理界面



2.全局的配置主要是配置maven, jdk,git的位置




3.系统的配置，这个配置一般都是安装完指定的插件后，就可以出现对应的配置



[Sandbox bypass vulnerability](#)
[Whole-script approval vulnerable to SHA-1 collision](#)
[Sandbox bypass vulnerability](#)
[JDK Parameter Plugin 1.0: Stored XSS vulnerability](#)

System Configuration
系统配置

 **Configure System**
Configure global settings and paths.

4.配置全局的jdk和maven,如果不配置的话,构建项目的时候会报错,提示配置

错误提示 JAVA_HOME environment

全局属性

☒ Environment variables

键值对列表

键

JAVA_HOME

我们这里添加一个全局的jdk的配置和maven的配置就可以了

值

/app/software/jdk/jdk1.8.0_361

键

M2_HOME

值

/app/software/maven/maven-3.9.0

键

PATH+EXTRA

值

\$M2_HOME/bin

gitee的配置

使用统计

- ☒ 通过发送匿名的使用信息以及程序崩溃报告来帮助Jenkins做的更好。

Gitee 配置

Gitee 链接

链接名

https://gitee.com

我的代码在gitee上，所以需要配置一个gitee

Gitee 域名 URL

https://gitee.com

Gitee 域名完整URL地址 (例如 https://gitee.com)

证书令牌

Gitee API 令牌 ▾

添加 ▾

Gitee API V5 的私人令牌 (获取地址 https://gitee.com/profile/personal_access_tokens)

weblogic插件的配置

这个是weblogic插件安装完后就会出现这个配置

WebLogic Deployment Plugin

Additional classpath

/app/wlfullclient.jar

指定wlfullclient.jar所在的位置

Configuration File

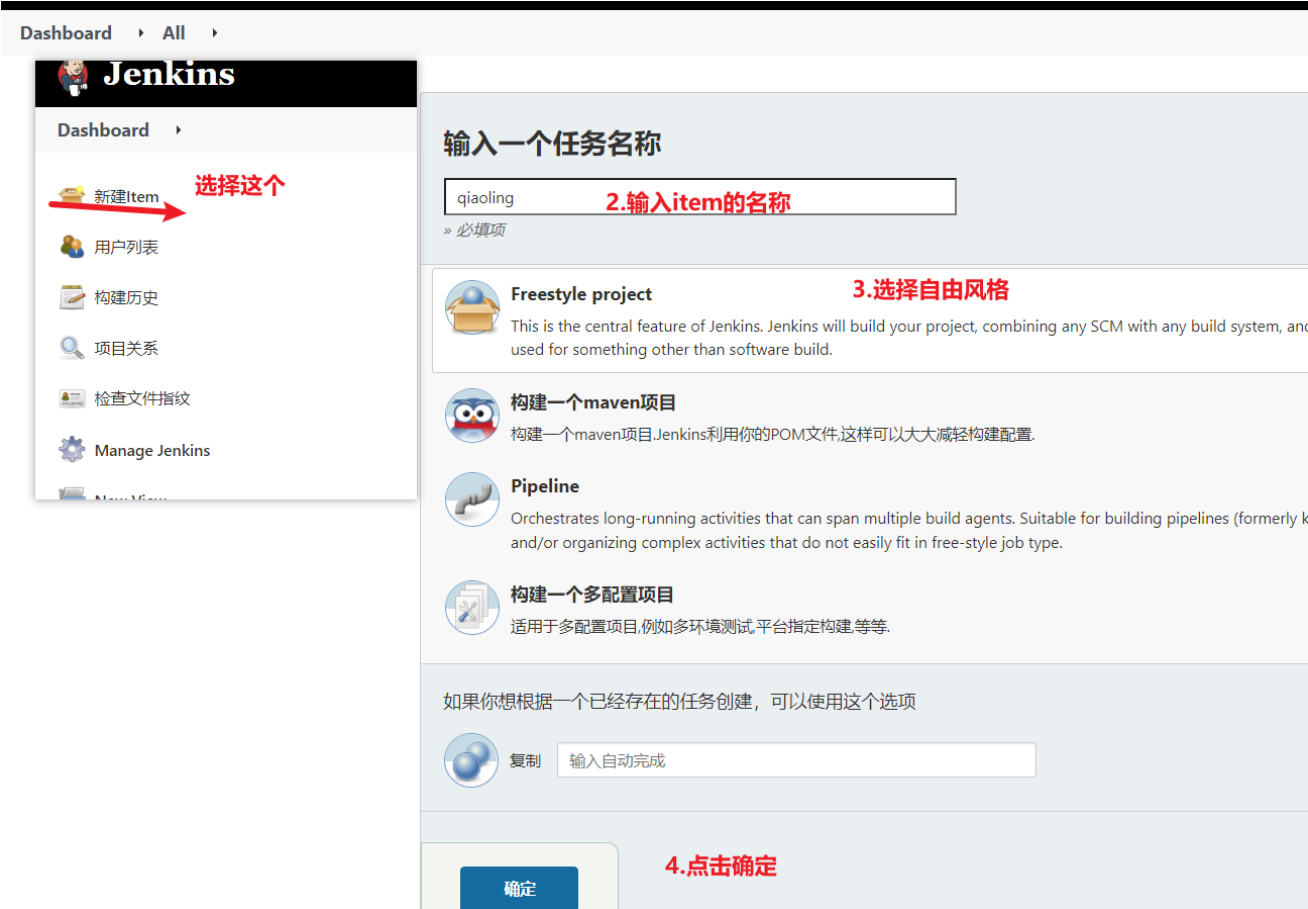
❌ The file does not exists.

wlfullclient.jar生成

```
连接虚拟机 进入/app/software/weblogic/oracle/wlserver/server/lib      #找到自己weblogic所在的目录即可
java -jar wljarbuilder.jar      #就可以生成该jar包，然后copy到app目录下即可。
```

jenkins创建一个item

1.创建一个自由风格的item



2.首先配置git的仓库地址和拉代码权限的凭据



2.1 git仓库的地址来源，我的代码在gitee上，所以gitee的地址来源如下图



登录

没有帐号? [点此注册](#)

18636594158

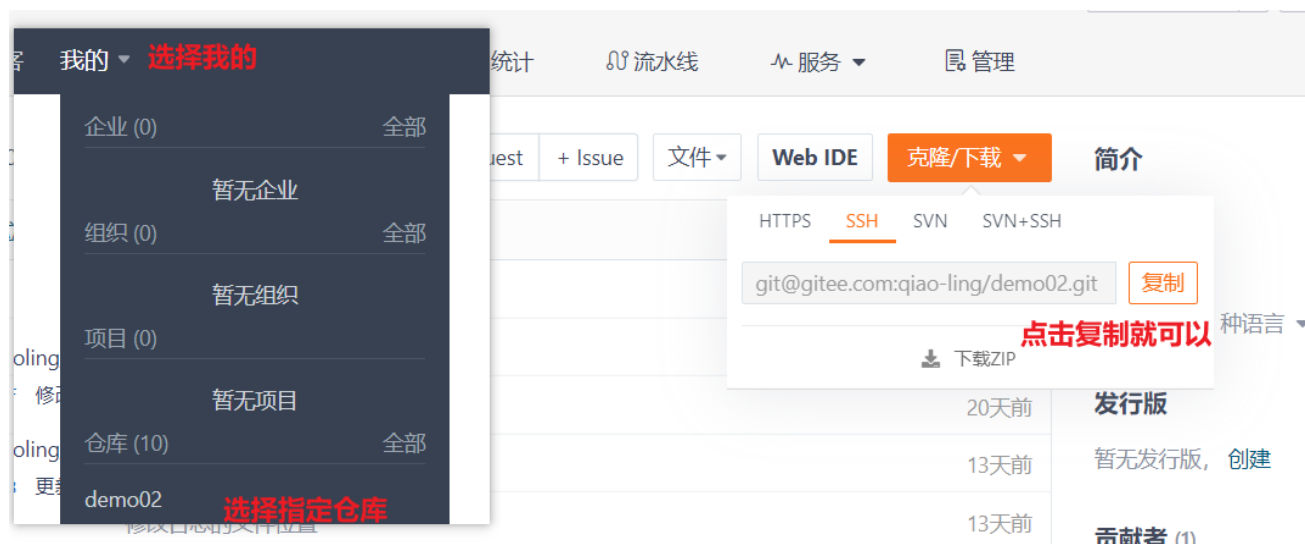
登录gitee, 没有就注册一个

☐ 记住我

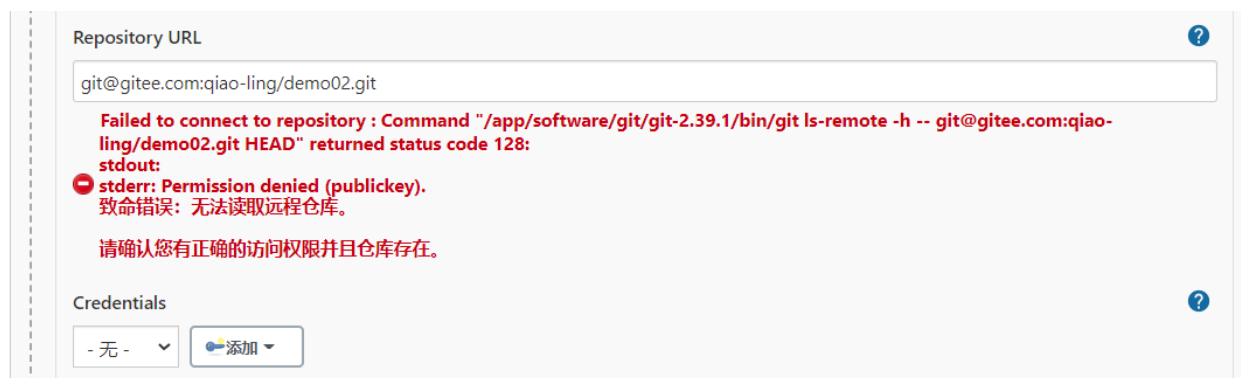
[短信验证登录](#)

登录

2.2找到仓库的地址:



2.3添加凭据的配置



这就代表没有凭据, 需要配置一个。步骤如下:

需要配置一个公私钥

在虚拟机中右键打开git, 添加一个用户

```
git config --global user.name "qiaoling"
```

```
git config --global user.email "773961901@qq.com"
```

生成git的公私钥


```
ssh-keygen -t rsa -C "773961901@qq.com"
```

```
[root@localhost ~]# ssh-keygen -t rsa -C "773961901@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

使用命令查看具体的公钥和私钥

```
vim /root/.ssh/id_rsa.pub 复制出公钥配置到gitee中的SSH公钥。即可，这个公钥配置到仓库中
```


GITEE的令牌，配置私钥，下面的私钥是要配置到jenkins中的凭据

```
577b4ea17dc953388e492d5d6c21d693
```

问题原因：在使用SSH Username with primary key创建凭据时，需要输入私钥，在选择直接输入私钥时图中红框部分的没有输入。

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAIc4qw0YtioZQUEz4E+iKpgpKKJmFmTSR8TFFB0c36K1D2Qh
w0AvysQgLqaE3ZkI5ZWVhbzD1V0nEw6DzIBTsIvzOHK2FrTOhz7WfDa9NLor9wA9
AyFN7fTAFnUmUXWT3R9uZ5SqwQZ7tZDitqiU16PWz2uUaTrHfZHfMIfAu1f5Y/p6
yUM9syE0BeYGTBAOYHJaXYbcb4soA/GSFeKojVw8GS1zkHj240BsLXiNZHXSTICF
WujgwySIRBtkVC/t7KCG1MiVBXsOUvrKPbQ2WCvrnsAhWyrZ7YzR1mqJZnjZsGE
OBynNY9V+rRoJ/LkKb34p1t52N1cysoy9lh40wIDAQABAoIBAQCumG0F2jGVujfZ
3/OQR14+8gdQarwqPk6rMgRzGBEUe12EksMPLiP89XpxzV8XV6Trp9a105AZp1B0
KZ2mtUZgzvZguBWUoxjbZas4/vk2wHLgrXbFsGkC3o1k6cbbIy3Tk1YyibQj+f7L
vdoTZBgdfHXA6jeJfwUWSyNuYqUoRT3r+ut8wqsJ6kT0B/DX2lUrS9Rfovoit1qt
cFd27ouDZc4BanVly9RuGKJaPLGGugWkw5LWTI7s8F4f3PPHkpcdY1J4hUjqG1qA
9D/Pu+UyF/SjqJLiW+AE0iNfvkGSwp++9Z25xNmewjeyrXl19ZOm/HoYwPn/q4SO
KO4+2UaBAoGBAPLPLTQbi0tpbgDws5a3UP5U15Xpsqyd1bSZX3L1UMwN4W14Ehq
+3z8b9tkZemsS/0bbwSANRx28hPi2iR+CUqsXfsmHUuFYvqck0Kna0T0lUXu0IkT
k2mPijgwtqFkyNV3o5K5Lgzq7wo/XeZ02LqPD5Vav6KhGMG4cJ9bCECTAoGBAOy+
16SppJANrXq5JHCPTIWpyDAzue1308VvsZWt3+XGp6imRFpHW4LX4KGRlzCpMqag
cmysal46R41Rap47TADqY0yKv1717oFu4xEQ60DnhUE2wsJLnbHkAYtfiWMURVvC
PzDJIq1hwHMBWwUhnjuFgJ1z5VwtVUe5wtTdbq5AoGAEdBDvxzgSBvw4hzciznf
TKKZ6bRS1+LaoPNVYTK41nJOULB04bt/8r7vU1D+rXPoK418yv7BZvJPdygZS743
3lPhnZFBaw5BohQ1Nink1C2CWpxYjwZxtbhdSzXX40XtWsSpqXWNjIMcwUq/ECxO
+PmQciwzjbJ2h5iKIdxh9qMCgYBPz9kzq/VMky/e5bfBMq4riZdxmJ6fzciawRLv
URTgU96F3rfAb7oMHFcgyN+12fS3c/PfayjY53DiFE4jVf5DeMCAPi5SBzsG66k1
QbwgxiyIVbvG4NMi+6kBoY2h113il1PSUJyc2dWVb+kDdU8E7rrfWgkC0E2lpxpG
Qte0EQKBgHpOuKjZ3p/d6CToSv00a3NxIgKbU5sX4V85ikVMK342VGcqvBLns/ba
rBM/+I83vZ971ZyqG96d0xaiE1QH1bG4N9RLjb5/b1C0TxM+DvZiiGY8/3wAGSsc
XhNviCCt+2uZ+aMp0C6T6dLbiBNzjCHPn1I4tyzc7klzKM0UB8yv
-----END RSA PRIVATE KEY-----
```

问题解决：修改凭据，将红框部分输入，然后便可以拉取代码了。

 Jenkins 凭据提供者: Jenkins

添加凭据

Domain

全局凭据 (unrestricted)

类型

SSH Username with private key 1.选择这个

范围

全局 (Jenkins, nodes, items, all child items, etc)

Private Key

☒ Enter directly 2.选择这个

Key

Enter New Secret Below

3.复制私钥的内容到这里就可以

Passphrase

☐ 无

☒ Git


Repositories

Repository URL

git@gitee.com:qiao-ling/demo02.git

Credentials

jenkins

 添加

配置正确后，错误就消失了

3.配置运行的pom，配置如下

构建

1.选择这个

增加构建步骤 ▲

Execute Windows batch command

Execute shell

Invoke top-level Maven targets

2.选择这个

Send files or execute commands over SSH

Send files over FTP

Set build status to "pending" on GitHub commit

构建



Invoke top-level Maven targets

X

Maven 版本

maven3.9

1.这个选择指定运行的maven

目标

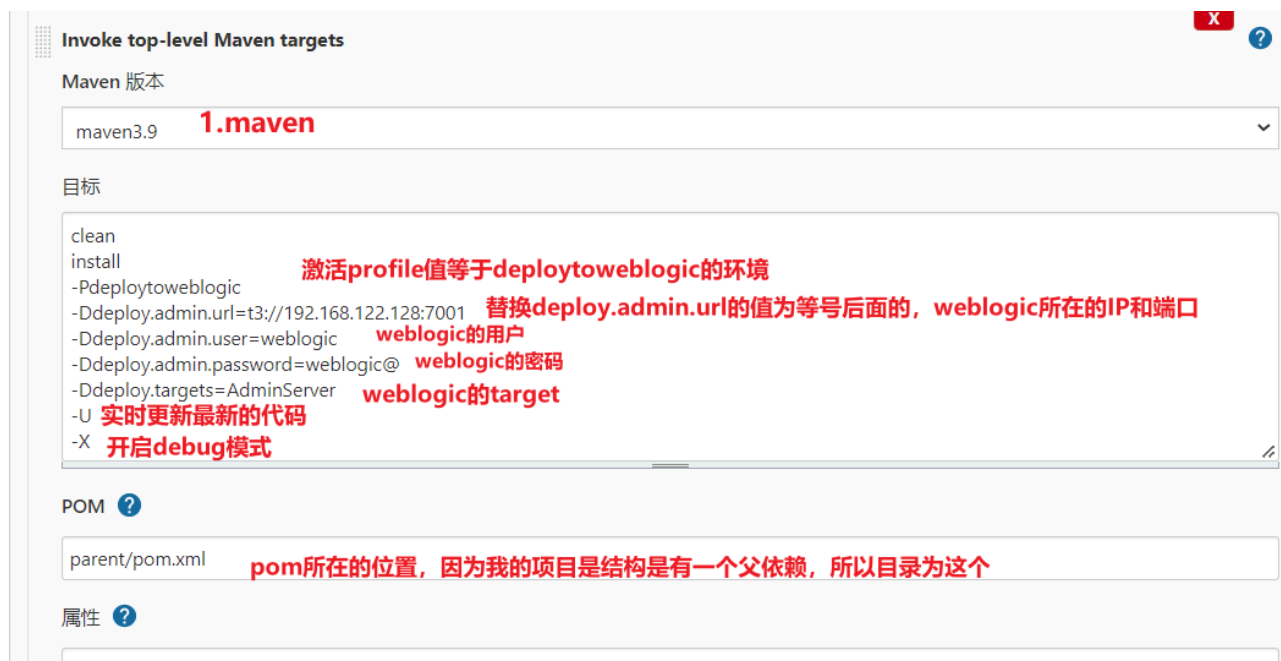
2.这个一般都是执行的maven指令，

POM ?

3.这个需要选择pom所在的位置，不写的话，不能执行，系统无法识别pom所在的位置

属性 ?

列举一个配置如下，具体的说明如下：



```
<build>
  <finalName>${project.artifactId}</finalName> 打包的项目的模块名称
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId> 需要用到的插件的依赖
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
      <plugin>
        <groupId>com.oracle.weblogic</groupId>
        <artifactId>weblogic-maven-plugin</artifactId>
        <version>10.3.4</version>
        <configuration>
          <adminUrl>${deploy.admin.url}</adminUrl> weblogic的url
          <user>${deploy.admin.user}</user> weblogic的用户名
          <password>${deploy.admin.password}</password> weblogic的密码
          <upload>true</upload> 是否上传
          <action>redeploy</action> 是否重新部署
          <remote>false</remote>
          <targets>${deploy.targets}</targets> weblogic的target的名称
          <verbose>true</verbose>
          <source>${project.build.directory}/${project.build.finalName}.${project.packaging}</source> 打包的文件来源
          <name>${project.build.finalName}</name>
        </configuration>
        <executions>
          <execution>
            <id>DeployToRemoteContainer</id>
            <phase>install</phase> 安装到weblogic上
            <goals>
              <goal>redeploy</goal> 发布失败是否需要重新发布
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
```



```
<profiles>
  <profile>
    <id>deploytoweblogic</id> 激活这个配置，实现发布到weblogic
    <build>
      <plugins>
        <plugin>
          <groupId>com.oracle.weblogic</groupId>
          <artifactId>weblogic-maven-plugin</artifactId>
        </plugin>
      </plugins>
    </build>
  </profile>
</profiles>
```

配置完成后，点击保存就可以。

Dashboard

乔令2

deploy2

返回面板

状态

修改记录

工作空间

Build Now

配置

删除 Project

重命名

Build History

构建历史

Filter builds...

#46

2023-3-20 下午1:46

Proje

相关键

Last b

Last s

Last s

Last f

Last u

Last c

出现这个进度条就证明开始构建了

如果构建过程中出现报错的话，就是下面，如果有报错，直接点进去，选择控制台输出，就可以查看相应的报错信息，去针对性处理问题

Filter builds...

#46

2023-3-20 下午1:46

红的代表构建失败

#45

2023-3-1 下午1:15

绿的代表构建成功

#44


2023-3-1 下午1:11


#43


2023-3-1 下午1:09


#42


2023-3-1 上午11:28


 返回到工程

 状态集


 变更记录

 控制台输出 **选择这个**

 编辑编译信息

 删除构建 '#46'

 Git Build Data

 上一次构建

adapting to one

[ERROR]

```
java.sql.SQLException: IO 错误: The Network Adapter could not establish the connection
    at oracle.jdbc.driver.T4CConnection.logon (T4CConnection.java:858)
    at oracle.jdbc.driver.PhysicalConnection.connect (PhysicalConnection.java:793)
    at oracle.jdbc.driver.T4CDriverExtension.getConnection (T4CDriverExtension.java:57)
    at oracle.jdbc.driver.OracleDriver.connect (OracleDriver.java:747)
    at oracle.jdbc.driver.OracleDriver.connect (OracleDriver.java:562)
    at java.sql.DriverManager.getConnection (DriverManager.java:664)
    at java.sql.DriverManager.getConnection (DriverManager.java:247)
```

这个错误，代表的是没有和weblogic连接起来，可以查看weblogic是否启动或者ip和端口是否正确就可以
我的原因是因为：weblogic没有启动起来

weblogic的相关配置

启动weblogic的命令,我的启动命令放在app/startWeblogic.sh

```
[root@localhost app]# cat startweblogic.sh
nohup /app/software/weblogic/oracle/user_projects/domains/base_domain/startWebLogic.sh >> /app/schedule/weblogic.7001.log 2>&1 &
[root@localhost app]#
```

 nohup后台不间断执行

 weblogic的位置

 把日志输出到这个位置

连接虚拟机

cat /app/startWeblogic.sh #查看命令

直接右键复制下来就可以了。

使用tail -f /app/schedule/weblogic.7001.log #动态的查看

```
pp]# tail -f /app/schedule/weblogic.7001.log
```

动态查看日志

常见的weblogic的配置有如下：需要保证下面的配置开启

更改中心

查看更改和重新启动

启用配置编辑。将来在修改, 添加或删除此域中的项目时, 将自动激活这些更改。

域结构

base_domain

- 域分区
- 环境
 - 服务器 **1.**
 - 集群
 - Coherence 集群
 - 资源组
 - 资源组模板
 - 计算机
 - 虚拟主机
 - 虚拟目标
 - 工作管理器
 - 并发模板
 - 资源管理

帮助主题

- 创建托管服务器
- 克隆服务器
- 删除托管服务器
- 删除管理服务器

主页 注销 首选项 记录 帮助

主页 > qiaolingdatasource > base_domain > 域分区概要 > base_domain > 部署概要 > 服务器

服务器概要

配置 控制

服务器是 WebLogic Server 的实例, 它运行在自己的 Java 虚拟机 (JVM) 上, 并具有自己的值。此页概括了已在当前 WebLogic Server 域中配置的每一个服务器。

定制此表

服务器 (已筛选 - 更多列存在)

新建 克隆 删除

<input type="checkbox"/>	名称 ^	类型
<input type="checkbox"/>	AdminServer (管理) 2.	已配置

新建 克隆 删除

AdminServer的设置

配置 协议 日志记录 调试 监视 控制 部署 服务 安全 注释

一般信息 集群 服务 密钥库 SSL 联合服务 部署 迁移 优化 超载 并发 健康状况监视 服务器启动 We

保存

使用此页可以配置该服务器的一般功能, 例如默认网络通信。

查看 JNDI 树

名称: AdminServer

模板: (未指定值) 更改

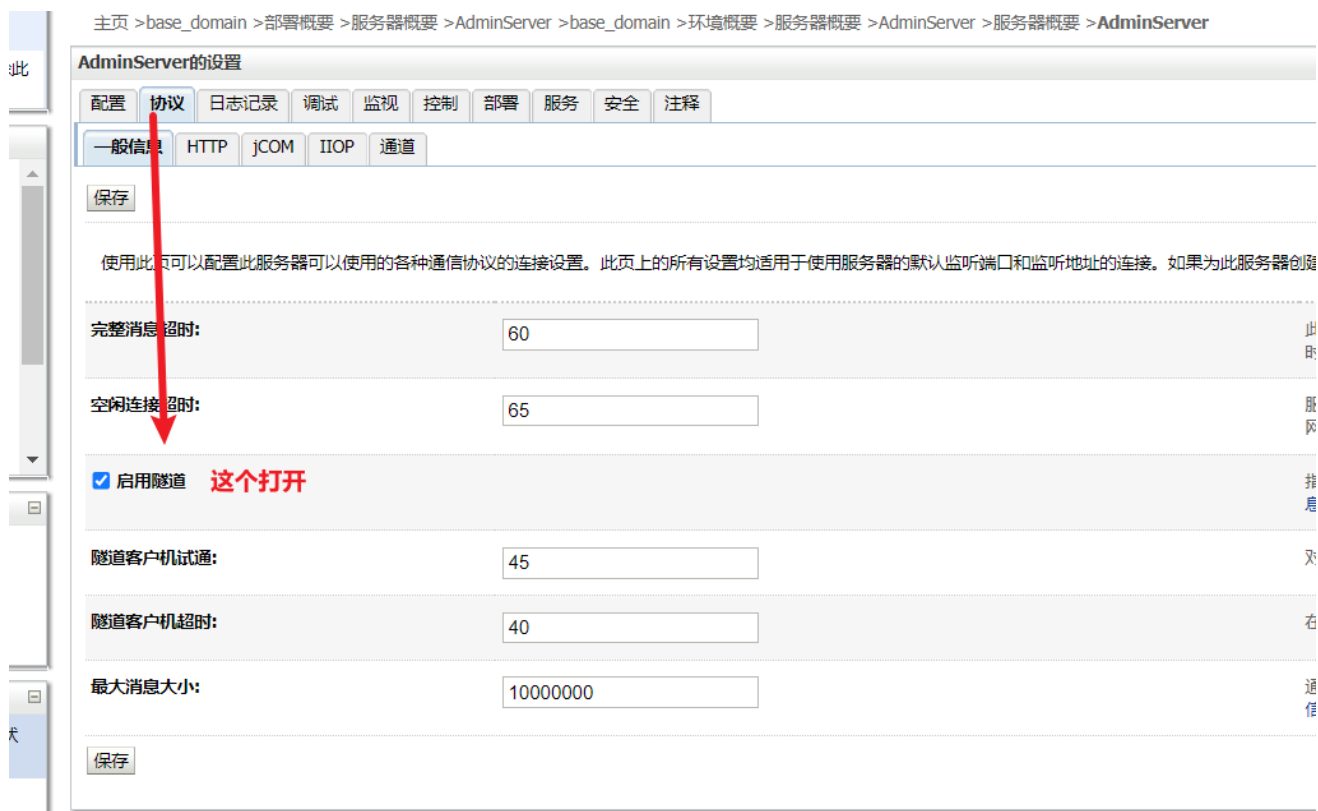
计算机: (无)

集群: (独立)

监听地址: 192.168.1.188

☒ 启用监听端口 **这个端口是否打开**

监听端口: 7001



weblogic中创建数据源，主要是项目中使用

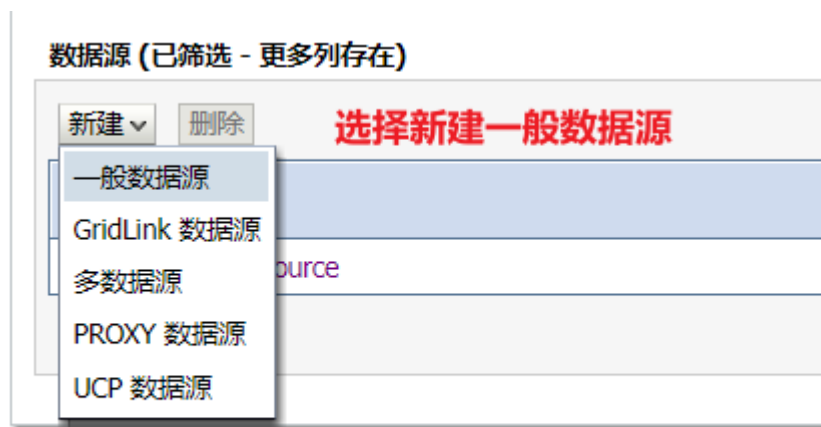
配置数据源的目的，
可以简化代码中的配置，代码中只需要配置一个JNDI的数据源的名称就可以解析

JNDI执行过程解析

代码发布到weblogic，weblogic通过JNDI的名称，就可以正确操作数据库，因为weblogic操作数据库的属性，已经提前配置到了weblogic，具体的配置如下

weblogic连接oracle数据源的创建





您希望如何命名您的新 JDBC 数据源?

* 名称:

您希望在什么范围内创建数据源?

范围:

您希望为新的 JDBC 数据源分配什么 JNDI 名称?

JNDI 名称:

您要选择哪种数据库类型?

数据库类型:

2个名称保持一致，这个是JNDI数据源的名称

我们选择oracle

新建 JDBC 数据源

JDBC 数据源属性

以下属性将用于标识新的 JDBC 数据源。

数据库类型:

您希望使用什么数据库驱动程序来创建数据库连接? 注: * 指示 Oracle WebLogic Server 明确支持该驱动程序。

数据库驱动程序:

默认下一步就可以了

上一步 下一步 完成 取消

事务处理选项

您已选择了非 XA JDBC 驱动程序, 以在新数据源中创建数据库连接。

此数据源是否支持全局事务处理? 如果支持, 请为此数据源选择事务处理协议。

☒ 支持全局事务处理

如果要使来自数据源的非 XA JDBC 连接可以通过使用记录最后一个资源(LLR) 事务处理优化来参与全局事务处理, 请选择此选项。建议您用此

☐ 记录最后一个资源

如果要使来自数据源的非 XA JDBC 连接可以通过使用 JTA 来仿真参与全局事务处理, 请选择此选项。仅当应用程序可允许出现试探性情况时才

☐ 仿真两阶段提交

如果要使来自数据源的非 XA JDBC 连接可以通过使用一阶段提交事务处理来参与全局事务处理, 请选择此选项。选择此选项后, 其他资源都不

☒ 一阶段提交

上一步 下一步 完成 取消

默认下一步

上一步 下一步 完成 取消

连接属性

定义连接属性。

您希望连接到的数据库的名称是什么?

数据库名称:

ORCL

数据库的名称

数据库服务器的名称或 IP 地址是什么?

主机名:

192.168.1.188

主机的ip

数据库服务器上用于连接到数据库的端口是哪个?

端口:

1521

数据库的端口

您希望用什么数据库帐户用户名创建数据库连接?

数据库用户名:

qiaoling

数据库的用户名

用于创建数据库连接的数据库帐户口令是什么?

口令:

.....

数据库的密码

确认口令:

.....

其他连接属性:

oracle.jdbc.DRCPConnectionClass:

上一步 下一步 完成 取消

然后选择下一步

主页 > base_domain > 环境概要 > 服务器概要 > AdminServer > 服务器概要 > AdminServer > 服务概要 > JDBC 数据源概要 > qiaolingdat

消息

✓ 连接测试成功。

2. 出现连接测试成功，就代表配置成功了

新建 JDBC 数据源

3. 一定要点击下一步，因为还有一个操作

测试配置

上一步

下一步

完成

取消

测试数据库连接

测试数据库可用性和您提供的连接属性。

用于在连接池中创建数据库连接的 JDBC 驱动程序类的完整包名是什么？

(请注意，此驱动程序类一定要位于要将其部署到的任何服务器的类路径中。)

驱动程序类名称：

oracle.jdbc.OracleDriver

1. 点击测试配置

新建 JDBC 数据源

上一步

下一步

完成

取消

选择目标

您可以选择一个或多个目标以部署新建的 JDBC 数据源。如果没有选择目标，则将创建但不部署数据源。您需要稍后部署此数据源。

服务器

☒ AdminServer

这个打勾，才算完成

上一步

下一步

完成

取消

2. 点击完成就可以了

这个就算是配置完成了。