基于jenkins + gitlab实现mave项目自动发布

一、环境描述

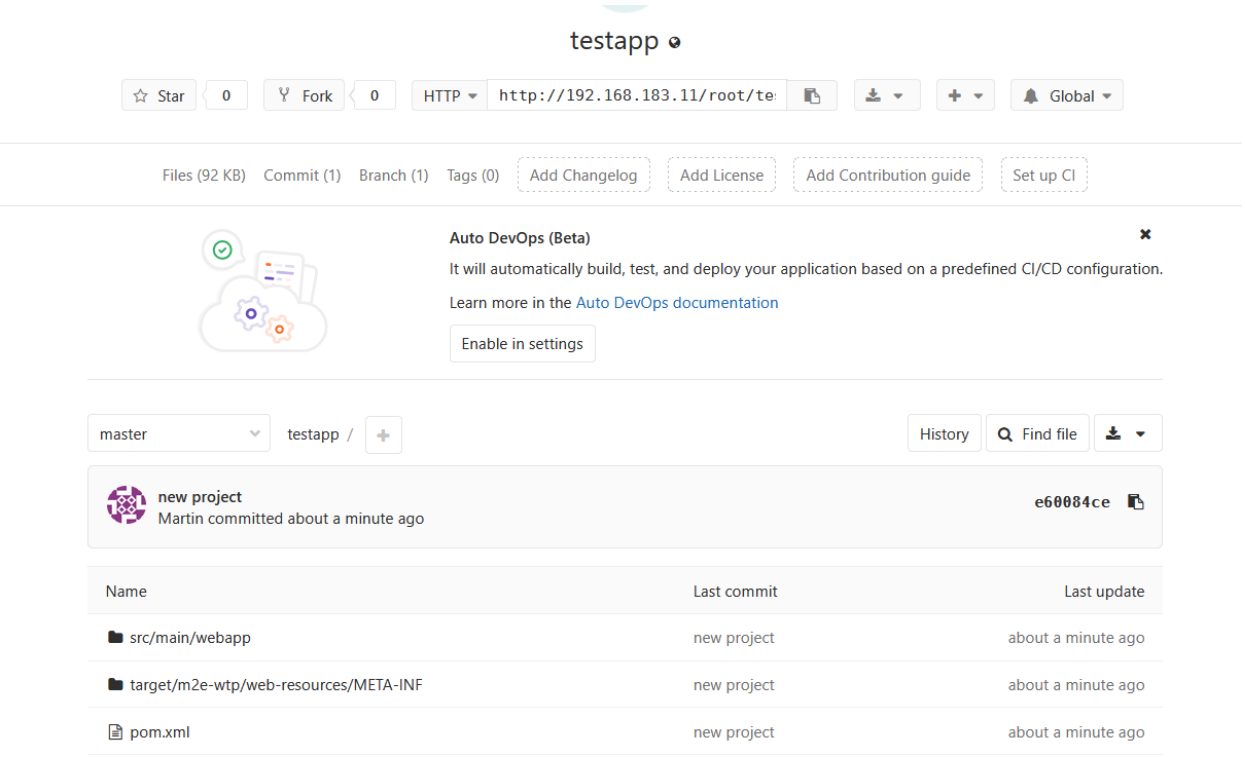
```
192.168.183.10 jenkins.linux.com
192.168.183.11 gitlab.linux.com
192.168.183.12 tomcat server
```

二、安装gitlab，上传项目测试代码

1、安装部署gitlab

```
[root@gitlab ~]# yum install -y gitlab-ce-10.1.5-ce.0.el7.x86_64.rpm

[root@gitlab ~]# grep -i "external_url" /etc/gitlab/gitlab.rb
external_url 'http://192.168.183.11'

[root@gitlab ~]# gitlab-ctl reconfigure
```

2、上传测试代码



三、安装配置jenkins

1、安装jdk1.8.0

```
[root@jenkins ~]# tar xf jdk-8u91-linux-x64.tar.gz -C /usr/local/
[root@jenkins ~]# vim /etc/profile
export JAVA_HOME=/usr/local/jdk1.8.0_91
export PATH=$PATH:$JAVA_HOME/bin

[root@jenkins ~]# source /etc/profile

[root@jenkins ~]# java -version
java version "1.8.0_91"
 Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
 Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
```

2、安装maven编译工具

```
[root@jenkins ~]# tar xf apache-maven-3.6.3-bin.tar.gz -C /usr/local/

[root@jenkins ~]# vim /etc/profile
export PATH=$PATH:$JAVA_HOME/bin:/usr/local/apache-maven-3.6.3/bin/
[root@jenkins ~]# source /etc/profile

[root@jenkins ~]# mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /usr/local/apache-maven-3.6.3
 Java version: 1.8.0_91, vendor: Oracle Corporation, runtime: /usr/local/jdk1.8.0_91/jre
 Default locale: en_US, platform encoding: UTF-8
 OS name: "linux", version: "3.10.0-693.el7.x86_64", arch: "amd64", family: "unix"
```

3、安装git客户端工具

```
[root@jenkins ~]# yum install -y git
```

4、安装jenkins

```
[root@jenkins ~]# rpm -ivh jenkins-2.277-1.1.noarch.rpm
warning: jenkins-2.277-1.1.noarch.rpm: Header V4 RSA/SHA512 Signature, key ID 45f2c3d5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:jenkins-2.277-1.1                ################################# [100%]
```

5、修改jenkins启动脚本，添加java命令路径

```
[root@jenkins ~]# vim /etc/init.d/jenkins
candidates="
/etc/alternatives/java
/usr/lib/jvm/java-1.8.0/bin/java
/usr/lib/jvm/jre-1.8.0/bin/java
/usr/lib/jvm/java-11.0/bin/java
/usr/lib/jvm/jre-11.0/bin/java
/usr/lib/jvm/java-11-openjdk-amd64
/usr/bin/java
 /usr/local/jdk1.8.0_91/bin/java
 "
 [root@jenkins ~]# systemctl daemon-reload
 [root@jenkins ~]# systemctl start jenkins
 [root@jenkins ~]# systemctl enable jenkins

 [root@jenkins ~]# chkconfig jenkins on
 [root@jenkins ~]# netstat -antp | grep 8080
 tcp6 0 0 :::8080 :::* LISTEN 1451/java
```

6、浏览器访问jenkins管理界面，进行初始化设置

```
http://192.168.183.10:8080
```

入门

解锁 Jenkins

为了确保管理员安全地安装 Jenkins，密码已写入到日志中（**不知道在哪里？**）该文件在服务器上：

`/var/lib/jenkins/secrets/initialAdminPassword`

请从本地复制密码并粘贴到下面。

管理员密码

••••••••••••••••••••••••••••••••

离线

离线

该Jenkins实例似乎已离线。

参考 **离线Jenkins安装文档**了解未接入互联网时安装Jenkins的更多信息。

可以通过配置一个代理或跳过插件安装来选择继续。

配置代理　跳过插件安装

1 此处出现离线，是因为jenkins默认联网的地址是国外服务器，国内无法正常访问，可选择跳过插件安装，后续修改国内插件下载地址

新手入门

创建第一个管理员用户

Username: martin

Password: ••••••••••

Confirm password: ••••••••••

Full name: martin

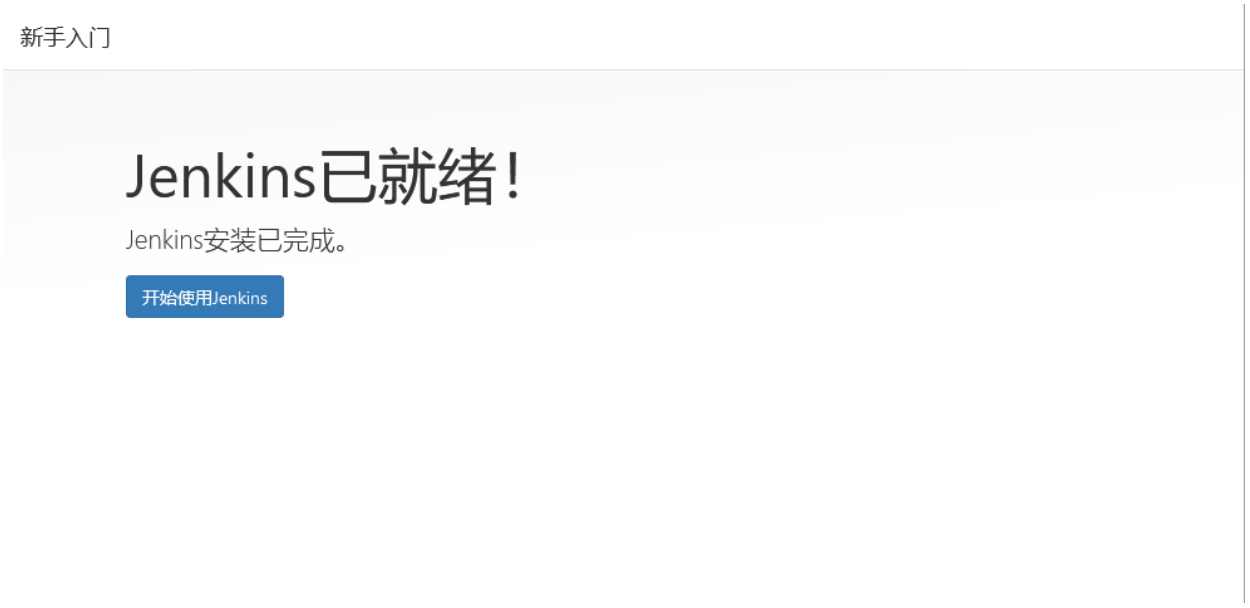
新手入门

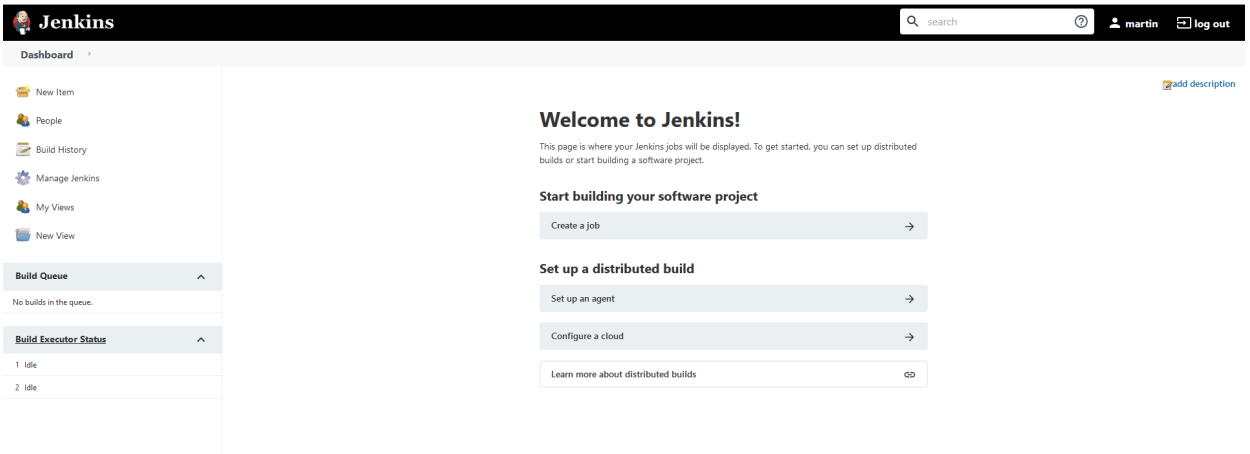
实例配置

Jenkins URL: http://192.168.183.10:8080/

Jenkins URL 用于给各种Jenkins资源提供绝对路径链接的根地址。 这意味着对于很多Jenkins特色是需要正确设置的，例如：邮件通知、PR状态更新以及提供给构建步骤的`BUILD_URL`环境变量。

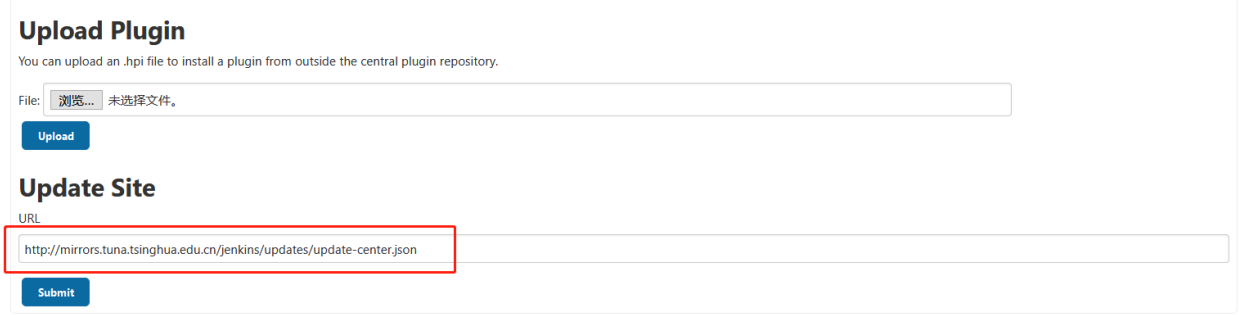
推荐的默认值显示在**尚未保存**，如果可能的话这是根据当前请求生成的。 最佳实践是要设置这个值，用户可能会需要用到。这将会避免在分享或者查看链接时的困惑。






```
至此，jenkins安装完成!!!
```

四、修改jenkins插件下载地址为国内地址




```
[root@jenkins ~]# systemctl restart jenkins
```

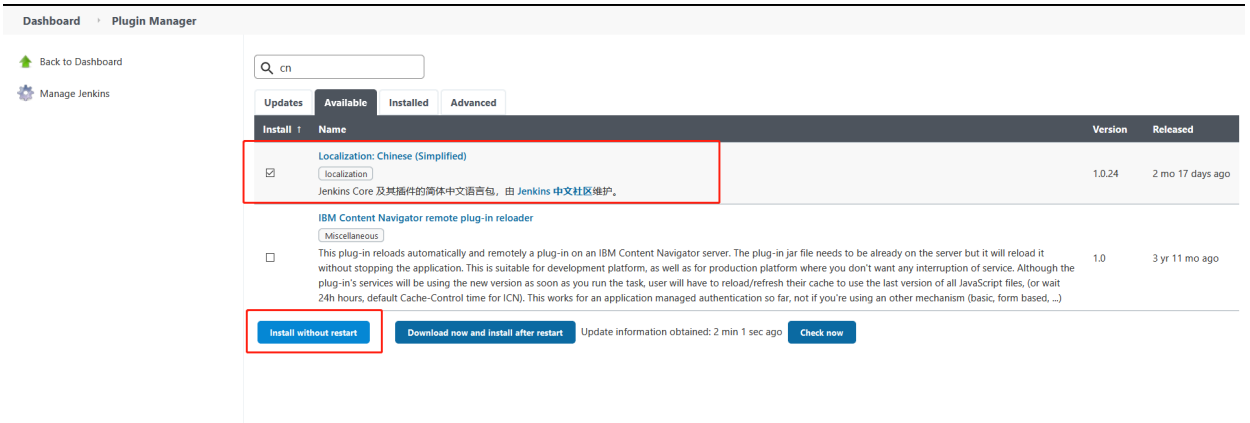
```

[root@jenkins ~]# sed -i 's/https:\/\/updates.jenkins.io\/download/http:\/\/mirrors.tuna.tsinghua.edu.cn\/jenkins/g' /var/lib/jenkins/updates/default.json && sed -i 's/http:\/\/www.google.com/https:\/\/www.baidu.com/g' /var/lib/jenkins/updates/default.json

[root@jenkins ~]# systemctl restart jenkins
```

五、安装插件

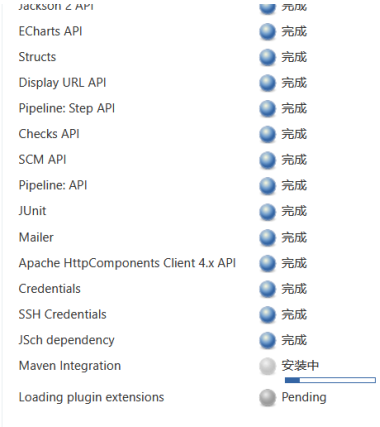
插件作用说明:

```
1. Maven Intergration 集成编译工具
2. Publish Over SSH 基于ssh协议向web服务器发布项目
3. git
4. gitlab
```

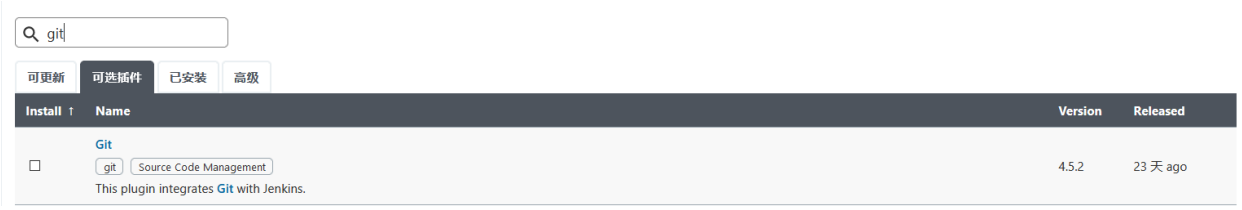
1、安装界面汉化插件



2、安装maven Intergration插件

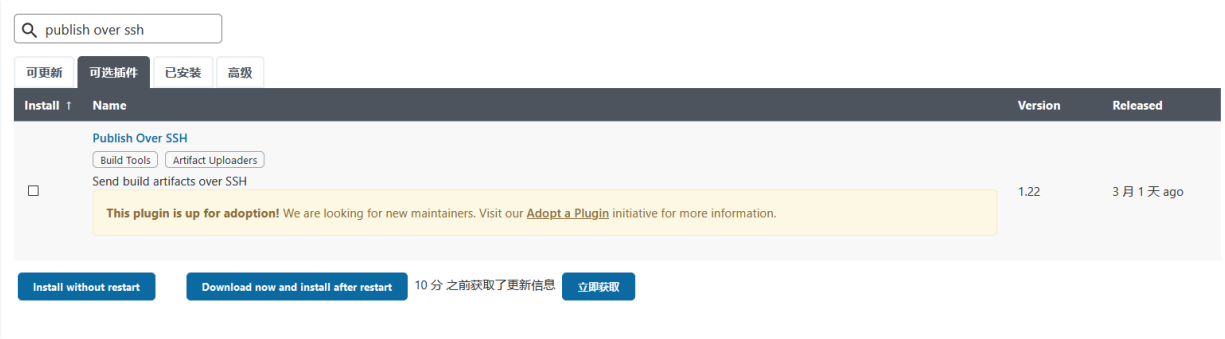


3、安装git插件



4、安装gitlab插件

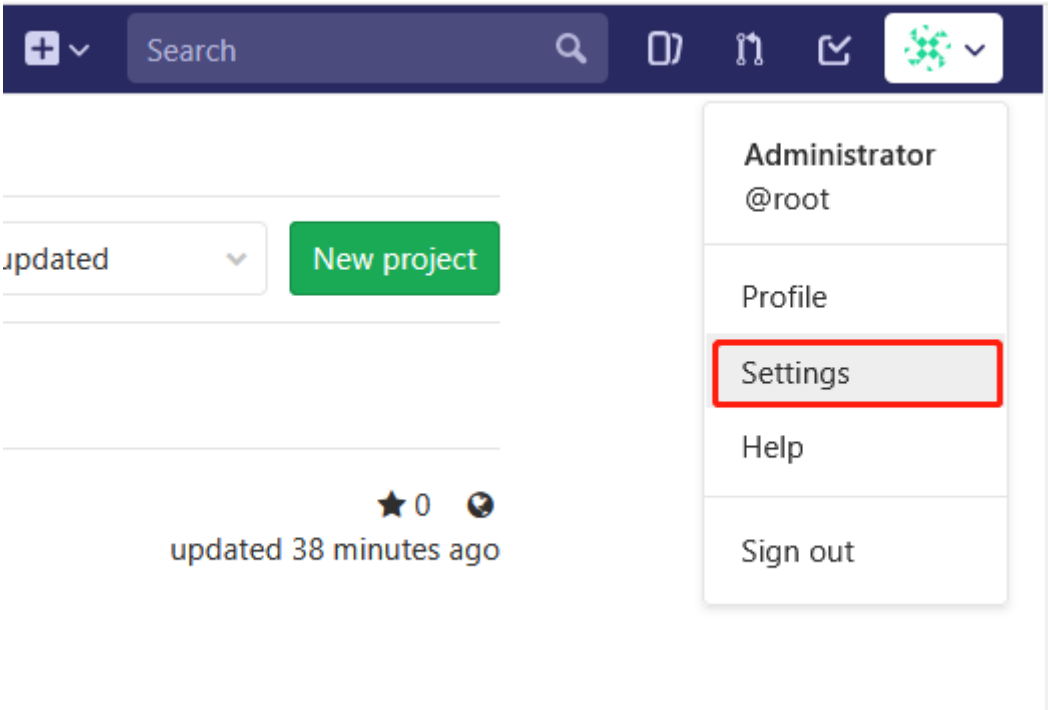
5、安装Publish over ssh插件

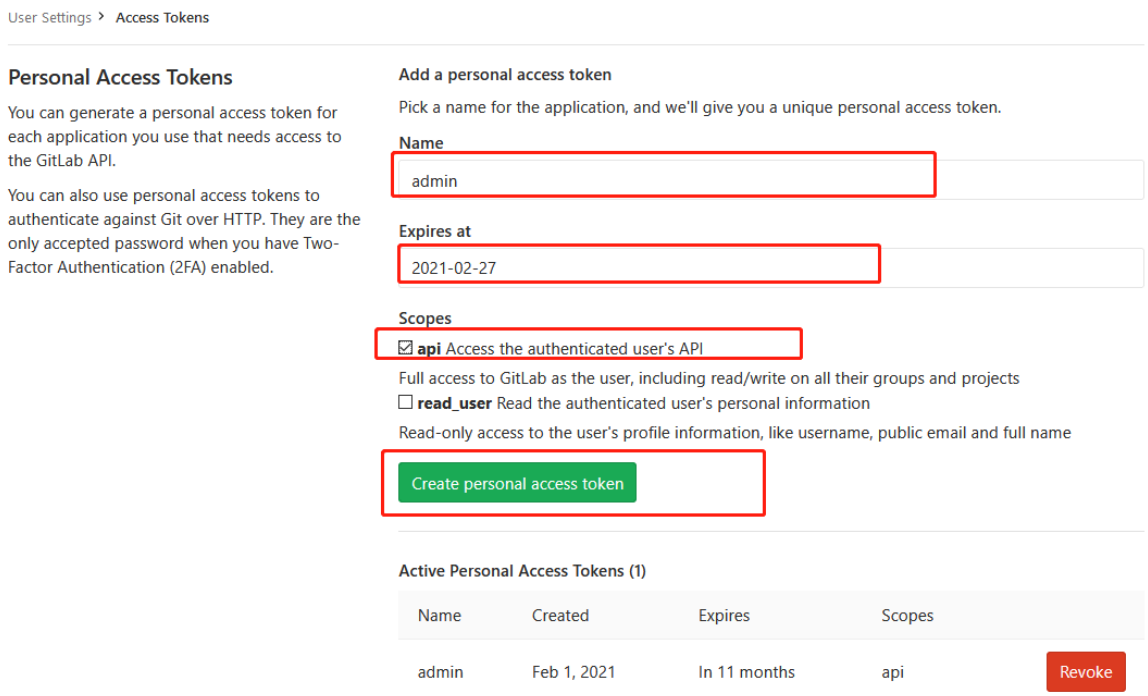


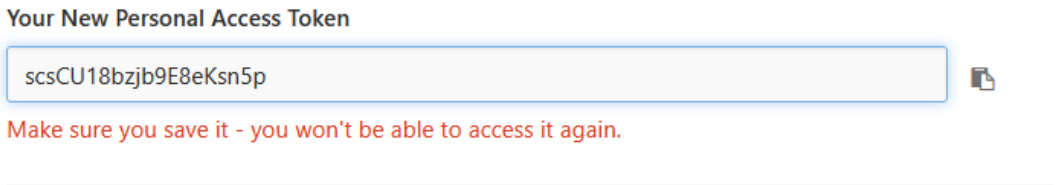
6、所有插件安装完成后，重启jenkins服务

六、配置jenkins连接gitlab

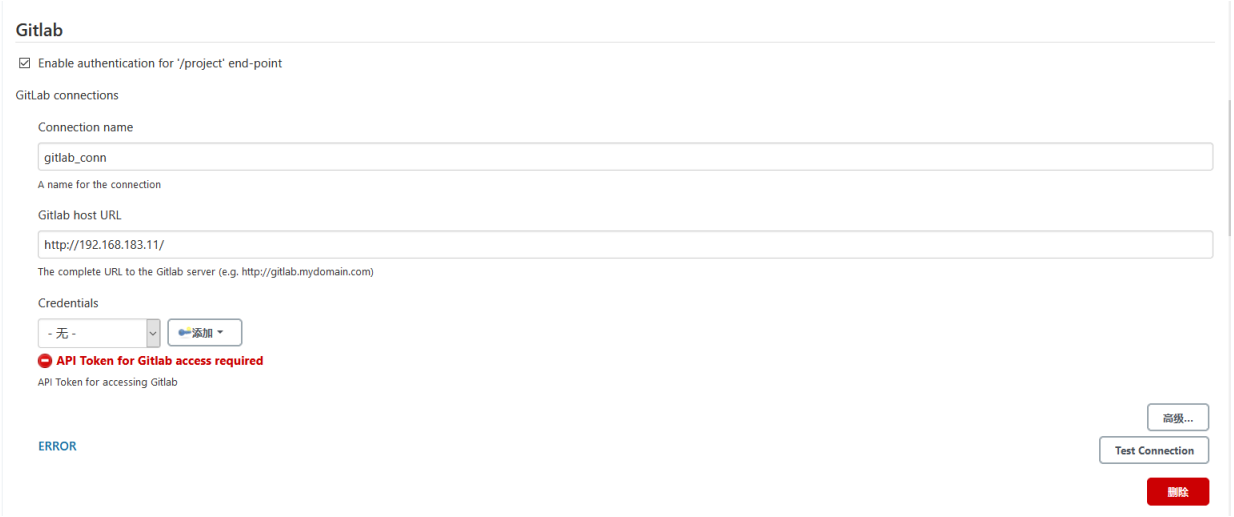
1、在gitlab上生成认证需要的令牌

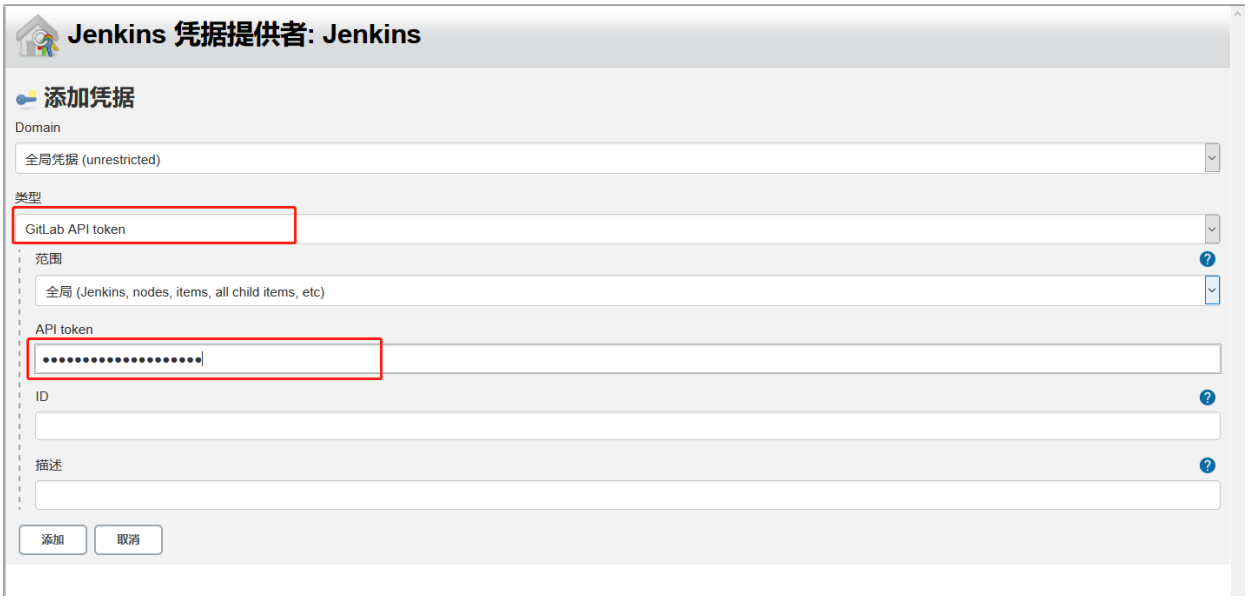


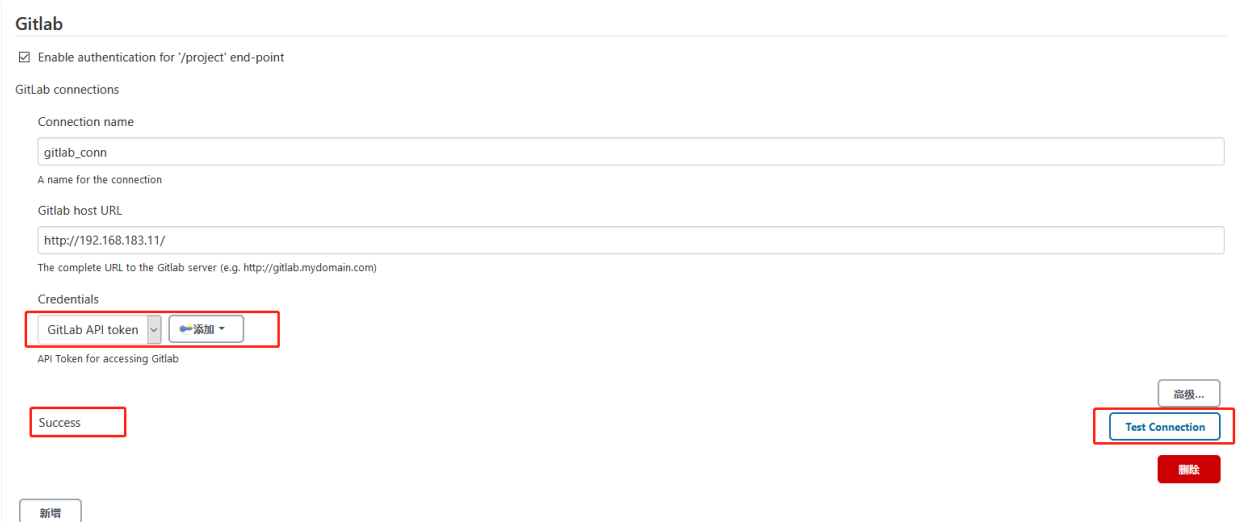




2、在系统设置中找到gitlab

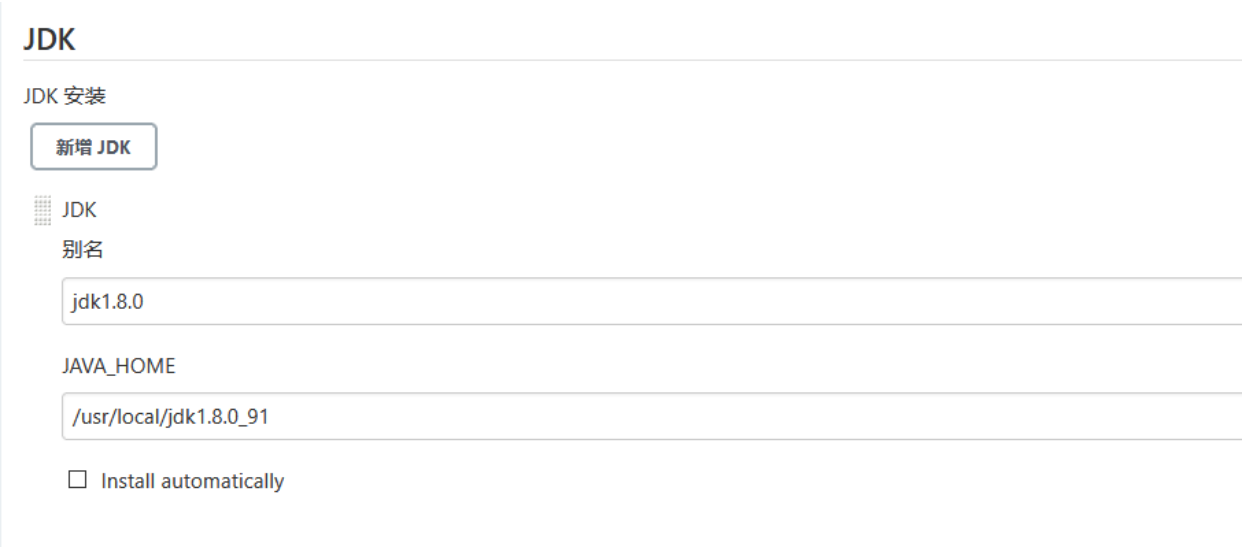




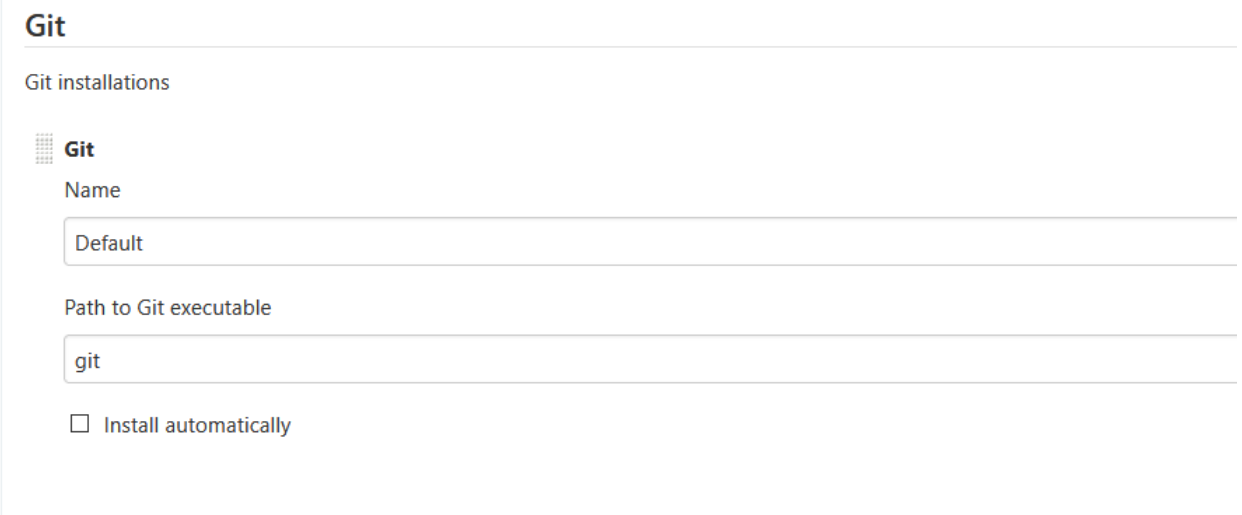


七、配置jenkins全局工具

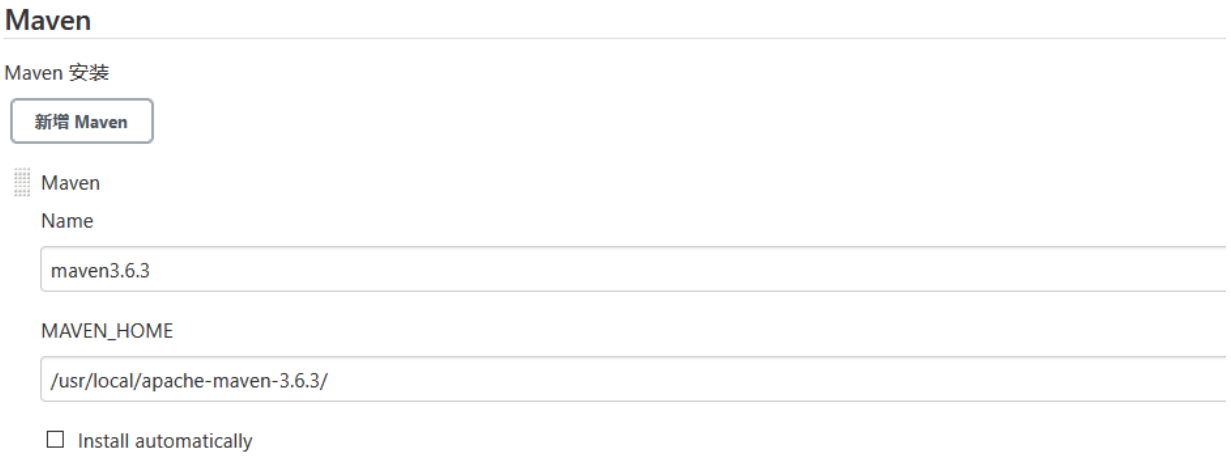
1、配置jdk安装路径



2、配置git客户端



3、配置maven工具

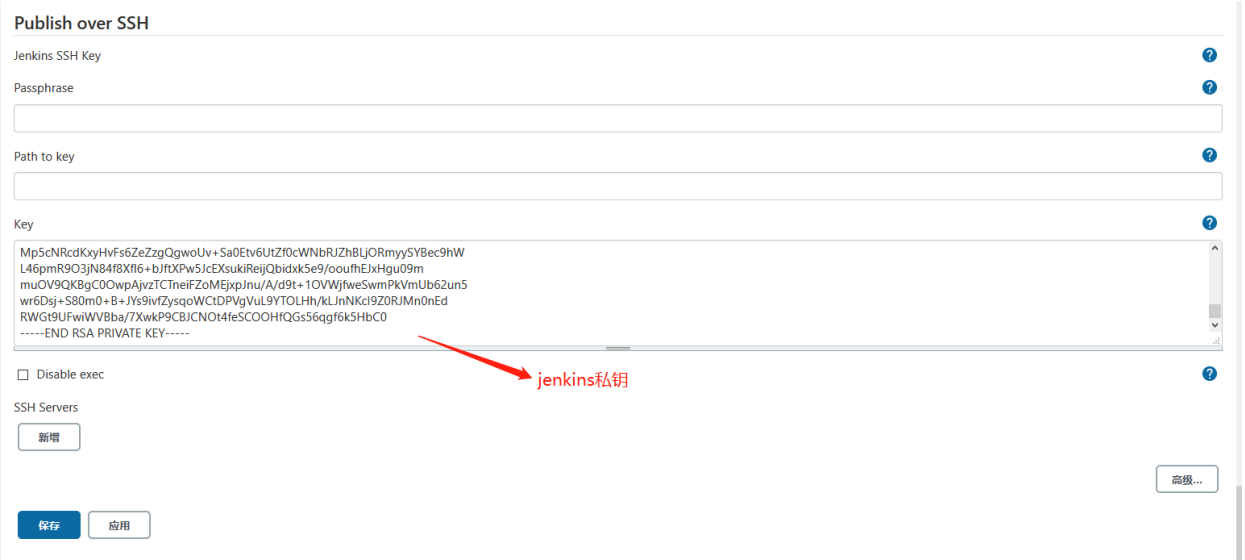


八、创建任务编译项目

1、配置jenkins与web server间的免密ssh

```
[root@jenkins ~]# ssh-keygen -t rsa
[root@jenkins ~]# ssh-copy-id root@192.168.183.12
```

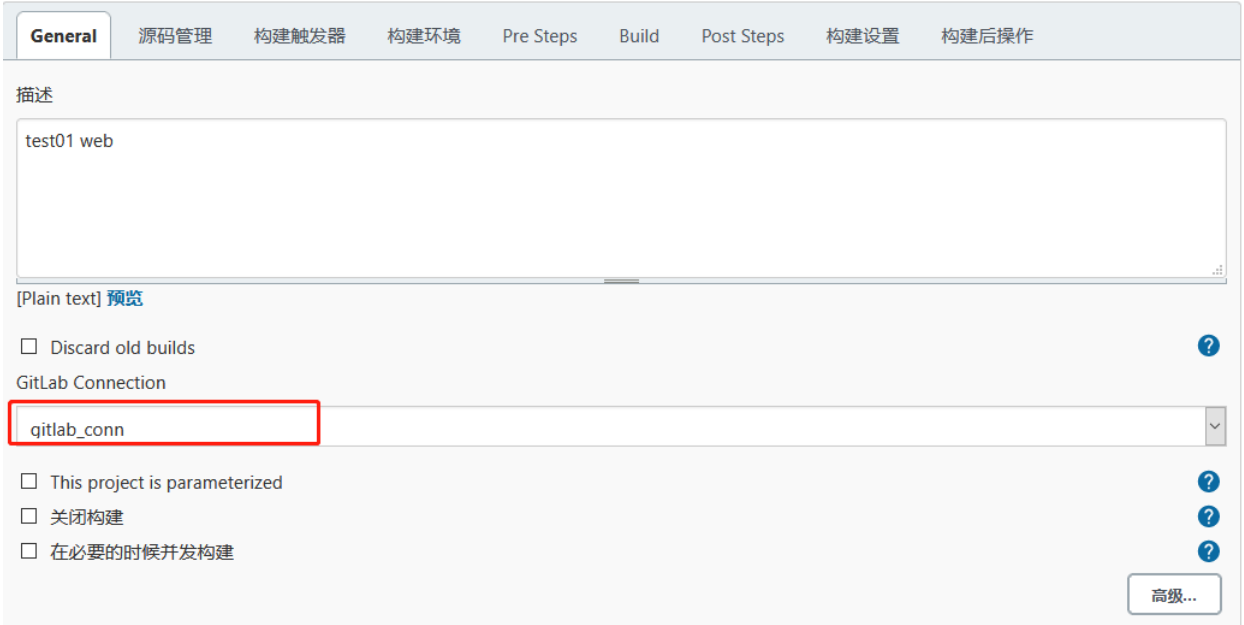
2、配置Publish over ssh插件



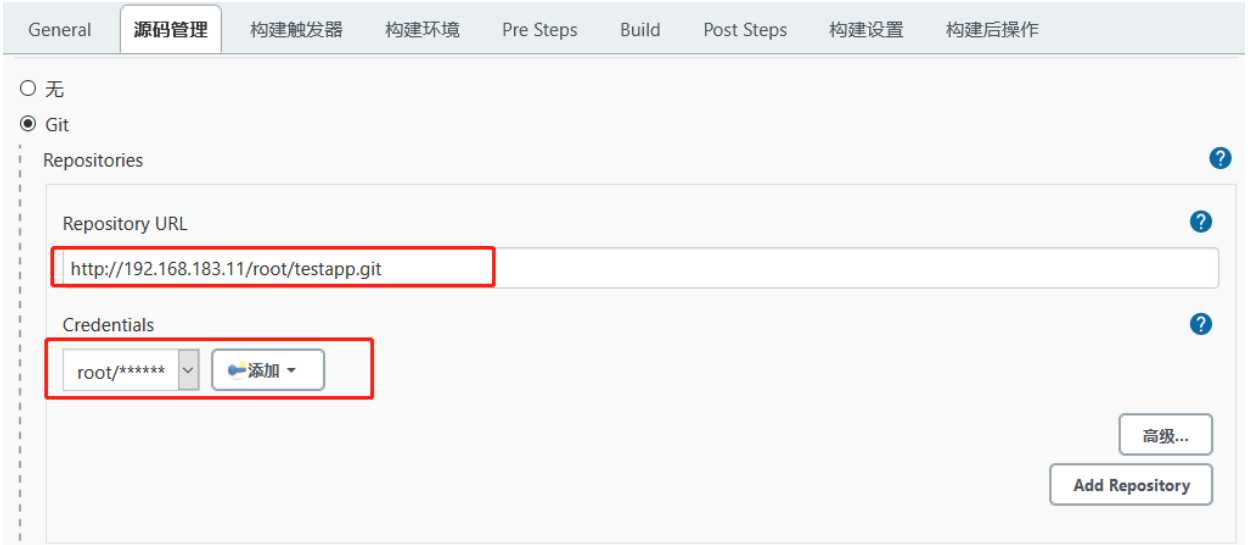
3、添加远程web服务器

4、创建任务

5、设置gitlab连接



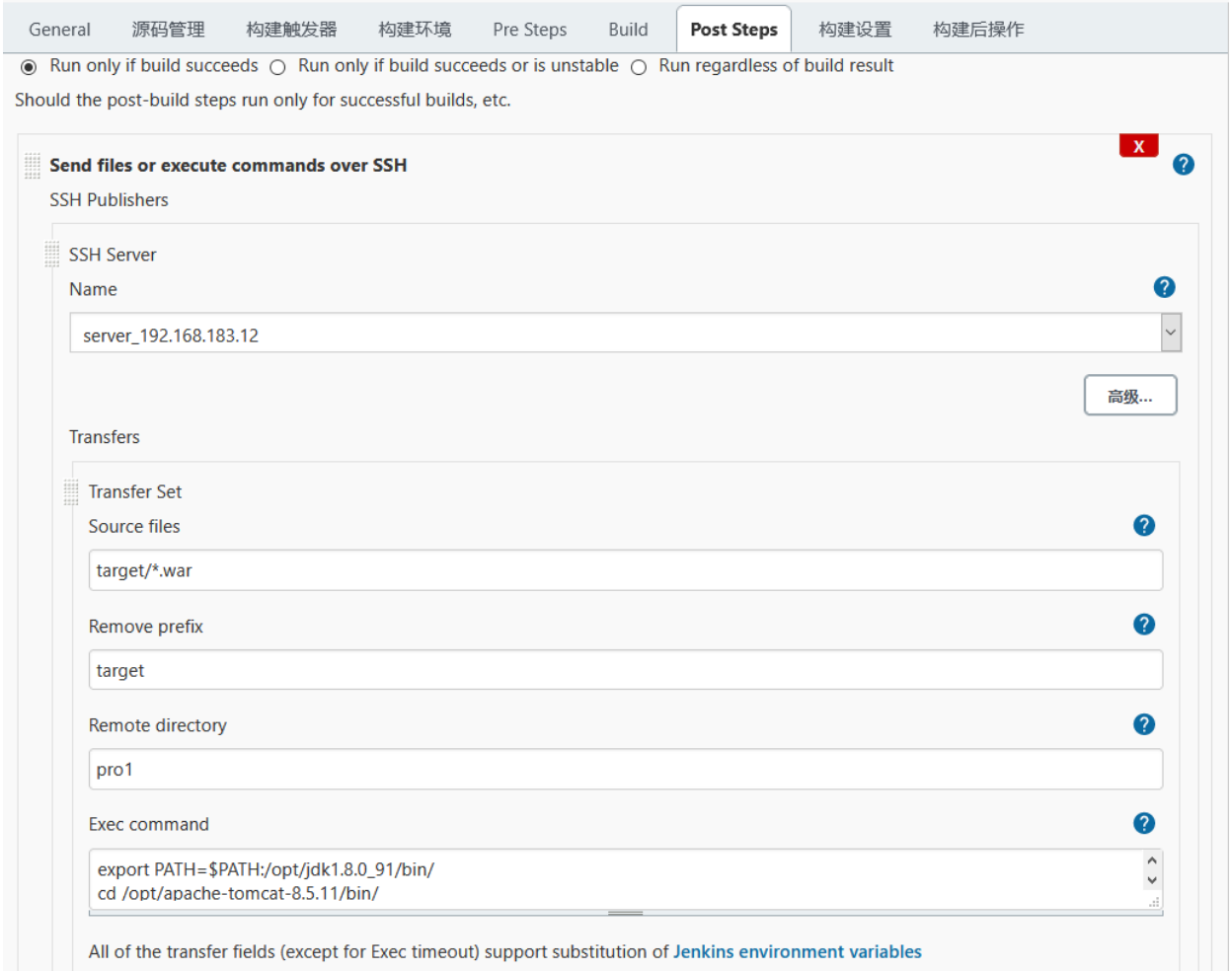
6、设置项目仓库地址



7、设置项目构建时间

8、设置pom.xml文件位置，相对于任务的存储位置

9、设置构建后执行的动作



执行命令参考

```
export PATH=$PATH:/opt/jdk1.8.0_91/bin/
cd /opt/apache-tomcat-8.5.11/bin/
./shutdown.sh
/bin/cp -rf /app/jenkins/pro1/*.war cd /opt/apache-tomcat-8.5.11/webapps
./startup.sh
```

九、测试

1、在jenkins服务器上查看构建的项目

```
[root@jenkins ~]# ls /var/lib/jenkins/workspace/test01
pom.xml src target
[root@jenkins ~]# ls /var/lib/jenkins/workspace/test01/target/
hello hello.war maven-archiver
```

2、查看发布到web server上的项目

```
[root@web_server webapps]# ls /app/jenkins/pro1/
hello.war

[root@web_server webapps]# ls /opt/apache-tomcat-8.5.11/webapps/
docs examples hello hello.war host-manager manager ROOT
```

3、客户端访问web server