目录

1、安装

* 1.1、安装插件
* 1.2、配置

2、配置

* 2.1、配置jenkins支持maven自动编译
* 2.2、配置文件

3、发布

4、脚本

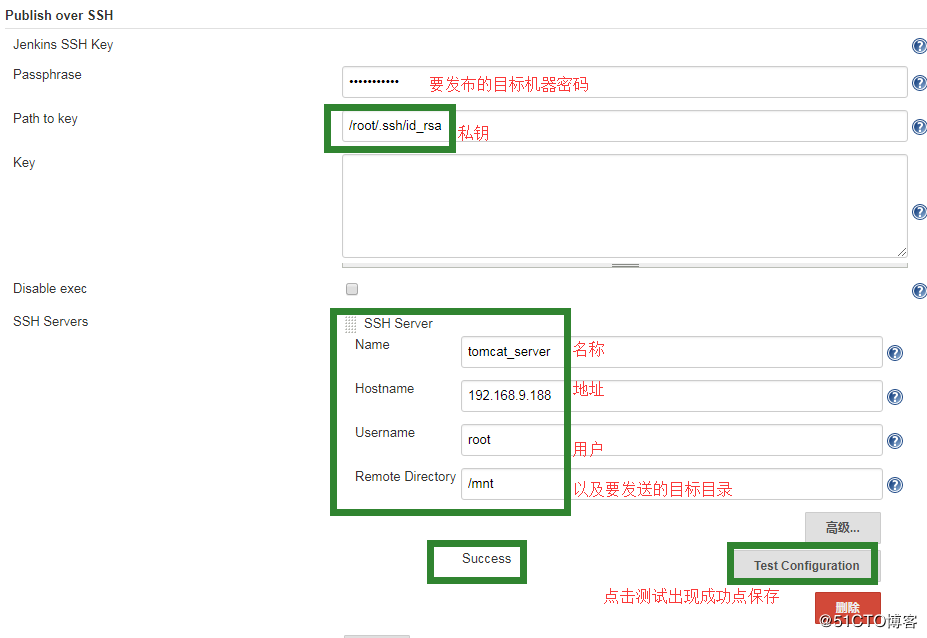
官方特性介绍：publish over ssh

* SCP - 通过SSH发送文件（SFTP）
* 在远程服务器上执行命令（可以禁用服务器配置或整个插件）
* 使用用户名和密码（键盘交互）或公钥认证
* 密码/密码短语在配置文件和用户界面中加密
* SSH SFTP / SSH Exec可用作构建过程中的构建步骤（Freestyle和Matrix项目）
* （maven）项目构建之前的SSH，或者在构建之后运行，无论构建是否成功（请参阅下面的[构建包装器](https://wiki.jenkins.io/display/JENKINS/Publish+Over+SSH+Plugin#PublishOverSSHPlugin-wrappers)）
* 该插件是“提升意识”（直接从正在升级的版本的工件目录发送文件），请参阅[促销](https://wiki.jenkins.io/display/JENKINS/Publish+Over#PublishOver-promotions)
* 可以为作业配置中的每个服务器覆盖认证凭证（或者在全局配置中未提供该服务器的情况下提供它们）
* 如果文件传输失败，可以选择重试（适用于flakey连接）
* 启用命令/脚本在伪TTY中执行

**1、安装**

* 官网介绍：https://wiki.jenkins.io/display/JENKINS/Publish+Over+SSH+Plugin
* **1.1、安装插件**
* 点击-系统管理-系统插件-可选插件中 直接搜 publish over ssh 点击安装，安装完之后，重启jenkins

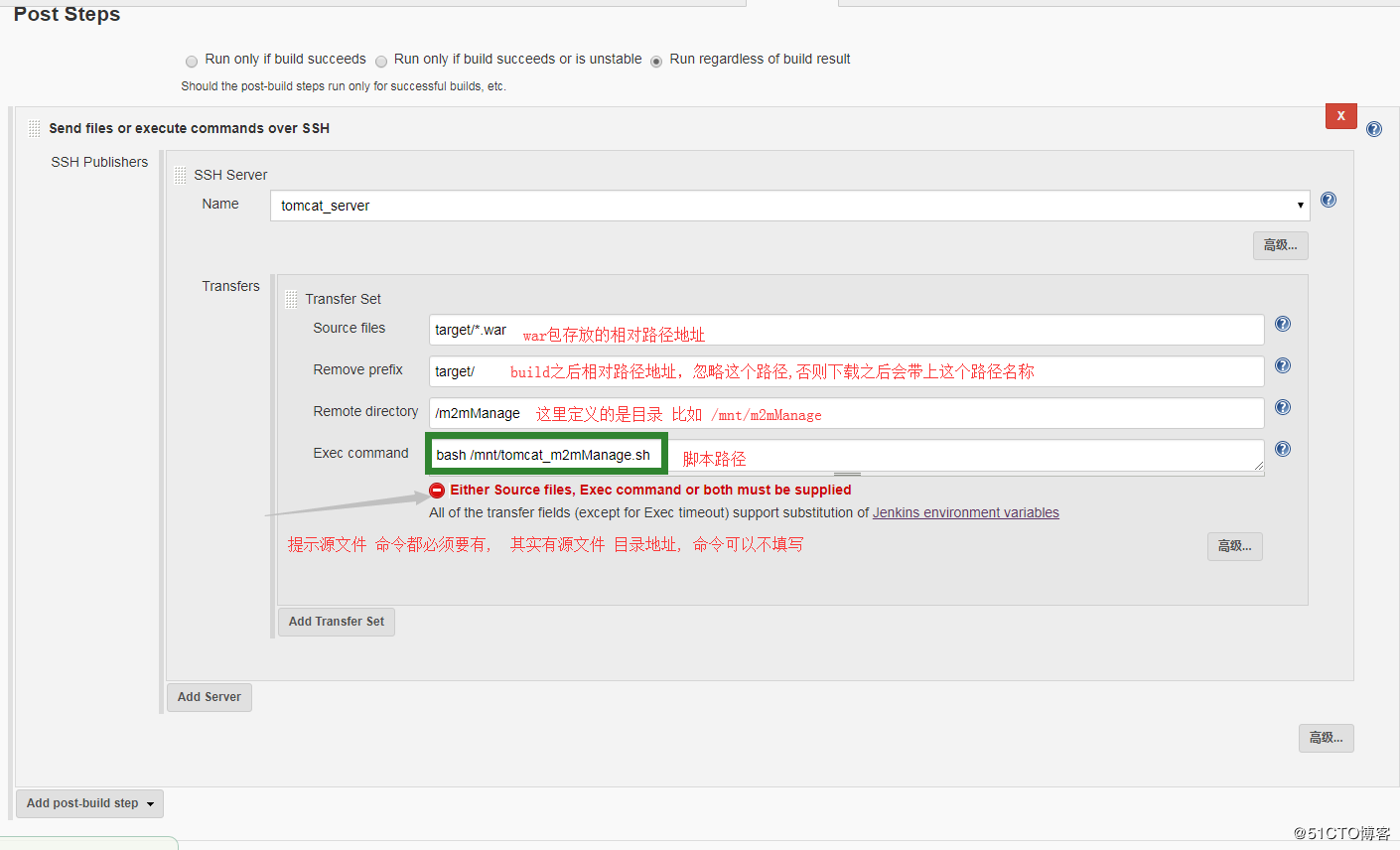
* **1.2、配置**
* 点击- 系统管理-系统设置-拉到最下面就有这个窗口了



* + Passphrase： 密码（目标机器的密码）
  + Path to key：key文件（私钥）的路径
  + SSH Server Name： 标识的名字（随便你取什么）
  + Hostname： 需要连接ssh的主机名或ip地址，此处填写应用服务器IP（建议ip）
  + Username： 用户名
  + Remote Directory： 远程目录(要发布的目录,比如/usr/local/tomcat/webapps/)

**2、配置**

* **2.1、配置jenkins支持maven自动编译**
  + 链接：实例3.2：  http://blog.51cto.com/xiong51/2093450
* **2.2、配置文件**



Name ： 这里是自定义的ssh远程服务器，如果有多台,点击add post-build step 继续添加

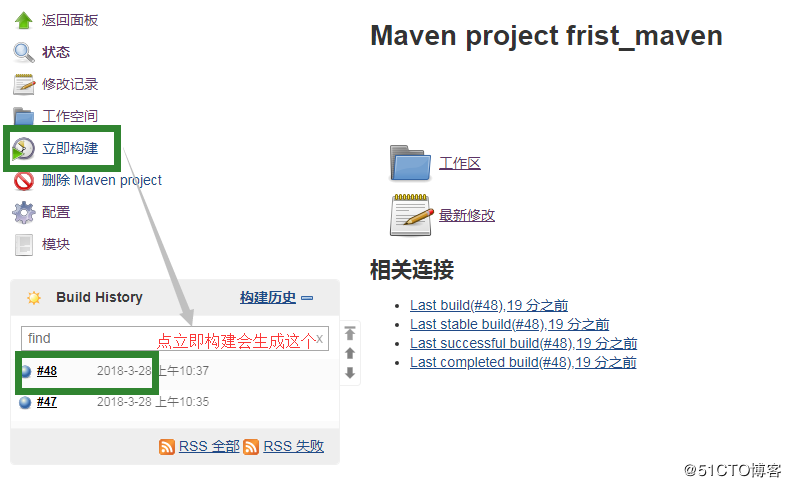
Source files ： 源文件地址，相对地址 比如 /root/.jenkins/workspace/jobname/target/xx.war 直接写成 target/xx.war就行

Remove prefix ： 这里指的是忽略前面的 target/ 远程服务器直接显示就是xx.war

Remote directory ： 远程服务器要保存的文件目录

Exec command ： 这里直接写 bash 脚本名称 xx.sh 这里也是远程服务器的脚本

**3、发布**





**4、脚本**

* 个人练习使用 ，直接修改 server\_name 是发布的项目名,由tomcat定义

     tomcat\_m2mManger是我定义的tomcat脚本名称，自带的脚本 进程可能会删除不了

war包地址 我定义的为/mnt，如果是别的更改一下目录就行

* **#!/bin/bash**
* #
* # \_\_auther\_\_: xiong
* # \_\_ time \_\_: 2018年03月28日 11:00:54
* # \_\_ 作用 \_\_: 用于jenkins build完成之后的最后操作shell脚本
* # \_\_ 使用 \_\_: 定义jenkins
* #
* #---------------------------------- 文件路径配置 ----------------------------------
* # 定义基础路径及目录
* server\_name=m2mManage
* # tomcat名称
* tomcat\_server\_name=tomcat\_m2mManage
* # tomcat 服务启动脚本
* tomcat\_serivce=/usr/local/$tomcat\_server\_name/sbin/$tomcat\_server\_name
* # 保存文件路径
* base\_dir=/usr/local/website
* work\_dir=$base\_dir/m2mManageDest
* # 以时间的方式来发布版本
* time=`date +%Y%m%d%H%M`
* time\_dir=$work\_dir/$server\_name$time
* # 链接路径
* link\_dir=$base\_dir/$server\_name
* # 配置文件路径
* conf\_dir=WEB-INF/classes
* #---------------------------------- 启动及判断配置 ----------------------------------
* # 创建目录,如果存在,直接退出重新发布一次
* if [ -d $time\_dir ];then
* exit 5
* else
* /bin/mkdir $time\_dir
* fi
* # 检测这个端口判断服务是否启动,
* if /usr/sbin/ss -tnl | grep 这里定义端口地址;then
* # 启动了那就直接停止它
* $tomcat\_serivce stop
* fi
* # 将打包上传的文件传到这个目录下
* /usr/bin/unzip /mnt/$server\_name/$server\_name.war -d $time\_dir/ &>/dev/null
* # 创建一个用于保存当前上线包的无效配置文件,并当无效配置文件复制到该文件中
* old\_conf\_dir=$time\_dir/$conf\_dir
* /bin/mkdir $old\_conf\_dir/old\_conf
* /bin/mv -i $old\_conf\_dir/\*.p\* $old\_conf\_dir/old\_conf/
* # 将上一个版本的配置文件复制到当前上线的文件中
* /bin/mv -i $link\_dir/$conf\_dir/\*.p\* $time\_dir/$conf\_dir
* # 判断上一个版本的软链接,用于回滚版本
* if [ -d $link\_dir ];then
* /bin/rm -irf $link\_dir
* fi
* # 重新定义新上线包的软链接
* /bin/ln -sv $time\_dir $link\_dir
* # 启动服务
* $tomcat\_serivce start