

# 部署工具说明

文件编号:

版本号:

## 变 更 记 录

序号	文档版本	修改内容	章节	修改人/ 日期	批准人/日期
1	20130805	新建		张子萌	
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

注：对该档内容增加、删除或修改均需填写此变更记录，详细记载变更信息，以保证其可追溯性。

## 目录

1.介绍 .....	4
2.安装介绍.....	4
2.1 安装环境要求.....	4
2.2 安装步骤.....	5
2.3 初步调试.....	6
3.快速开始.....	7
4.账户管理.....	9
5.项目管理和设置.....	9
5.1 新建项目.....	9
5.2 单个项目的详细设置.....	9
5.2.1 全局配置.....	9
5.2.2 解锁操作.....	10
6.上传包操作.....	10
6.1 打包上传.....	10
6.2 单文件上传.....	11
6.3 文件目录列表.....	11
7 部署操作.....	11
8.删除项目文件.....	11
9.删除部署包.....	11
10 单机检测 .....	12
11 环境搭建 .....	12
12.全局设置 .....	12
13.个人信息 .....	12
14.特殊需求 .....	12
14.1 限制 ip 登陆.....	12
14.2 登录信息保存.....	13
15 注意事项重点说明.....	13
15.1 回退工作.....	13
15.2 删除后的回退工作.....	13

# 1.介绍

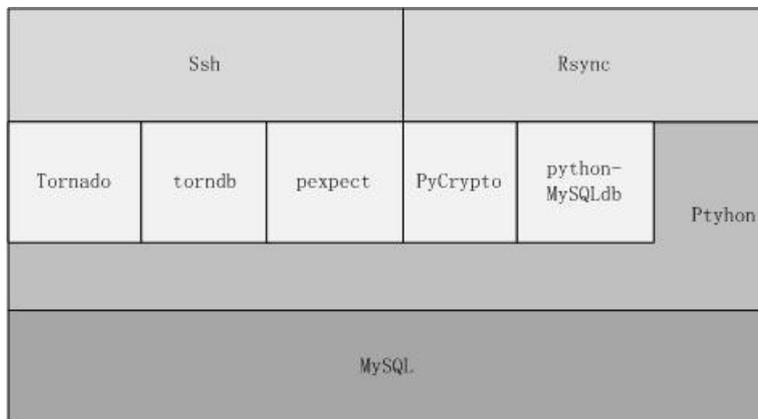
Web 部署工具。主要针对小型网络环境进行，有部署、回退、权限管理、日志记录等功能。

此部署系统是用 Python2.6(以上)+Tornado 开发，以调用命令的方式进行工作，用 WEB 界面形式进行管理。该系统需要用到第三方工具，如 ssh、rsync 等。所使用到的第三方工具的安装，都已经编写在安装脚本中，只要在能连接互联网的机器上可以使用安装脚本自动完成。但是基础的 python 环境需要手动安装，如果系统自带的版本过低，建议安装新版本（该系统还不能使用 python3.x）。

该工具使用推送和远程调用命令方式，所以没有 agent 端，部署配置后既可以使用。

由于属于个人开发，所以我的操作系统为 CentOS、raspberry pi、pcduino。

所用到的模块示意图如下：



该系统可以单进程或多进程进行部署，多进程是使用线程方式进行控制，建议为 3 个不要超出。在单进程情况下 10 个 Tomcat 服务部署完毕（传送文件网络耗时按 3 秒钟算）大约一分钟，因为 tomcat 需要重启耗时比较长，所使用时间大约是 php、python、ruby 类代码的 3 倍。使用多线程部署 Tomcat 类大约要 35-40 秒。此系统对于 JAVA 依赖于 tomcat 的服务适合在 20 个左右的服务环境中使用。脚本类的大约在 50 到 60 个服务环境使用。如果想以更快的方法，部署跟多的服务器，可以使用推送部署脚本的方式进行，后面在做介绍。

注意两点：1) 以上所说服务环境是指服务程序部署的套数，而不是物理服务器的台数。

2) 如果部署 tomcat 下的服务，为了避免因为环境变量造成的问题，需要编写一个适合远程使用脚本。可以参考 <http://www.simonzhang.net/?p=1951>

## 2.安装介绍

### 2.1 安装环境要求

操作系统：CentOS 或 Rasperry pi 。基础运行环境：python2.6 或 python2.7。数据库：mysql。

原则上可以运行在任何 python2.6 以上版本搭建的 linux 服务器上，但是本人只测试了 CentOS 和 Raspberry pi。

演示环境为 Raspberry pi。

## 2.2 安装步骤

首先创建数据。数据库创建部分不详细记录。如获得数据库信息：账号：deploy、密码：123456、库名为：deploytest、数据库 IP 为：192.168.1.130。数据库使用 UTF-8 编码。

将 deploy 代码放到相应目录中。(提醒：因为此系统安装需要安装第三方的框架和工具，建议在 root 权限下安装，并且装过程需要连接互联网)。测试过程将代码放置到 /program/deploytest 目录中。

开始安装。进入部署工具下的 install 目录，给 install.sh 可执行权限后并运行。执行效果如下：

```
root@raspberrypi: /program/deploytest
MySQL IP
192.168.1.130
MySQL Port
3306
MySQL DB Name
deploytest
MySQL User
deploy
MySQL password
123456
title
test
Domain Name
192.168.1.130:8889
local global ip
192.168.1.130
local global port
8889
=====
Please input your system id
  id  system name
   1   CentOS
   2  Raspberry pi
please write system id:
2
rasberrv
```

红字部分为系统的安装提示，黑色字为输入信息。信息输入完点击回车。当系统 id 输入点击回车后系统开始安装。

需要注意的部分“Domain Name”、“local global ip”和“local global port”的关系。“Domain Name”可以是外部访问域名，也可以是直接访问的 IP+端口的地址，如果为域名需要前端有使用 80 端口 WEB 服做映射，如 apache、nginx 等。“local global ip”为 WEB 服务映射的地址，“local global port”为 WEB 服务映射的端口，在部署系统启动时此端口会被部署系统占用。

如安装脚本没有报错，则安装完成，系统自动启动。如果自动系统失败，可以尝试下面的操作。

退到部署系统的根目录，使用命令“chmod 755 main.sh”给启停脚本可执行权限。使用命令“./main.sh start”启动服务，如看到“start ok!”则为启动成功，可以开始初步调试工作。启动如下图：

```
pi@raspberrypi /program/deploytest $ ./main.sh start
start ok !
pi@raspberrypi /program/deploytest $
```

## 2.3 初步调试

在浏览器中输入刚才设置的 url。演示地址系统所创建的地址为“http://192.168.1.130:8889”。第一次登陆使用管理员账户，默认用户名“admin”，密码为“1”。进入部署系统如图：



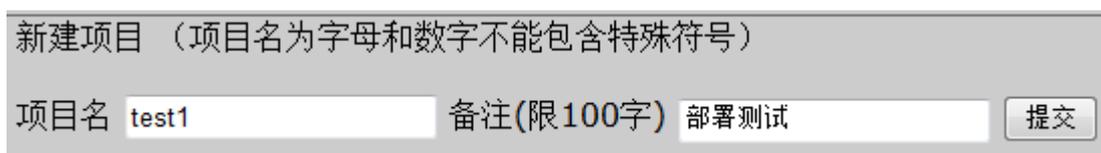
进入“全局设置”，可以修改初始化安装的域名和 Title 部分。进入“个人信息”，可以修改自己的账户密码。建议修改管理员密码。如无需修改则可以开始使用本系统了。

### 3.快速开始

在“全局设置中”设置的“添加用户”中添加部署操作的用户。如设置用户名为“test1”，密码自拟填写，后“提交”。如下图



然后在“新建项目”中创建一个项目，如设置“test1”，最好有备注，可以方便记录项目的功能。填好后“提交”。如下图



修改新用户的权限。首先通过下拉菜单选要修改的用户“test1”，另其可以管理项目“test1”，并可以使用部署工的所有功能。悬着完成后“提交”。如图：



管理员默认可以用所有功能，但如果不分配项目，则不能管理任何项目。所以点击“注销”退出管理员权限，使用“test1”用户登录。

登录后在左侧栏目中选择“单项目设置”，进入页面后通过“项目设置”的下拉列表选择相应项目。如图



进入如“test1”项目后开始对该项目服务器的具体信息进行配置。配置是以导入文件形

式进行，文件模板可以在线点击“配置文件模板”进行下载。如图：

上传配置(建议用UE或EditPlus保存为UTF-8编码)

[配置文件模板](#)

配置文件

模板是在 linux 下编辑的在 windows 下看可能会产生乱码，如果在 windows 下编辑推荐使用 UE 或 EditePlus 工具。配置编辑完毕 提交配置文件。如提示“修改成功”后在此进入该页面，会在最下面看到所提交的信息。如图：

当前服务列表。提交要删除的服务。

- p1 服务地址: 192.168.1.130
- p2 服务地址: 192.168.1.130
- p3 服务地址: 192.168.1.130
- p4 服务地址: 192.168.1.130
- p5 服务地址: 192.168.1.130
- p6 服务地址: 192.168.1.130
- p7 服务地址: 192.168.1.130
- p8 服务地址: 192.168.1.130
- p9 服务地址: 192.168.1.130

基础配置已经完毕，开始可以部署了。将代码在根目录级别压缩为 ZIP 文件。点击“代码打包”，选择则项目并上传压缩包。如图

代码打包 1

打包上传 单文件上传 svn上传 git上传 ftp基线上传 文件及目录列表

上传增量部署包(不能超过80M。包为在根目录下打的zip包)

选择 test1 上传部署文件 C:\Users\Administrator\De   全量包

备注(90个字)   4

如果压缩包上传成功，系统将自动跳转到“部署或回退”页面。选择要部署的项目，然后“提交”系统将开始部署。如下图：

部署工作

test1  部署  回退  3

部署完成是否重启服务  重新部署

单线程部署, 两次部署间隔10秒 配置替换暂未开启

<input checked="" type="radio"/> test1	20130805121053	备注:	未部署
--	----------------	-----	-----

部署操作为后台操作，即使页面退出部署也不会停止。为保同一个项目的上传和部署只能一个用户操作，在上传包和部署期间均有锁操作。后面讲具体介绍。

开始部署会有部署状态框显示，在部署状态框中看到“【ok】”则部署完成。到此部署系统已经正常运行，并成功的部署了一个项目。

## 4.账户管理

为了保留数据，本系统在“全局设置”中只有新添加帐户、禁用账户和恢复用户默认密码。在恢复账户默认密码的时候，如是被禁用的账户也同时开启。

具体位置如图：

代码打包	本站全局设置
部署和回退	域名 192.168.1.130:8889 本站 Title test 提交
单项目设置	添加用户
删项目文件	添用户名 设置密码 用户邮箱 提交
删部署包	恢复用户默认密（默认密码为 1234567） 禁用账户。如要启动账户，对其初始密码即可。
单机检测	用户名 当前用户列表 提交 用户名 当前用户列表 提交
环境搭建	新建项目（项目名为字母和数字不能包含特殊符号）
全局设置	项目名 备注(限100字) 提交
个人信息	
注销	

## 5.项目管理和设置

项目管理分为两部分：一、在整个系统中添加新建项目。二、针对单个项目做详细设置。

### 5.1 新建项目

进入“全局设置”设置页面，在新建项目中添加项目名和备注后提交即可。添加完毕可以分配给相应的用户进行管理。

### 5.2 单个项目的详细设置

进入“单项目设置”中。选择自己可以管理的项目。当前只有“全局配置”和“解锁”两个功能可以使用。

#### 5.2.1 全局配置

单项目的全局配置可进行部署过程默认配置，配置信息批量处理，管理当前服务。

部署过程默认配置中，针对服务可以只部署代码，如 python、php、ruby 等，如 java、系统服务等部署完毕必须重启。重启操作也是在配置文件中需要填写，如果没有启停命令、或命令是错误的，则重启会失败。所以启停脚本需要进行测试后再使用。针对 java 代码启停比较复杂，在使用过程中对环境变量需要做较多编写。针对 tomcat 启停脚本的改造可以参考 <http://www.simonzhang.net/?p=1951>。

部署代码过程分为中，第一种为多线程，每个线程会在后台调用一个部署进程。建议最高 10 个线程，如果 100 个服务，会分 10 批进行部署，20 个以上服务在使用此方法。第二种为单进程，系统会更具数据库的记录信息单步部署。单进程缺点是慢，优点是可以循环启停，前端负载根据启停服务会中断正在部署的连接，这样部署过程不会让客户端感觉到服务宕机。

配置文件的上传是批量加载服务器信息，模板可以在线上下载，填写要使用绝对路径。下载部分可以作为服务器资料备份的功能使用。如有信息批量变更，只需要先下载修改后直接上传即可。

如有部分服务下架需要删除，可以在列表中直接复选后点击“删除”即可。

## 5.2.2 解锁操作

为了避免同一个项目同时多人操作，上传或部署造成冲突，所以在开始上传或者部署时系统会进行锁操作，待上传或部署完毕，自动解锁。如果因为特殊原因自动解锁失败，可以通过此功能进行手动解锁。需要注意，需要确认部署已经过了一段时间，当前确实没有人在操作该项目再解锁。

# 6.上传包操作

点击“代码打包”，该页为代码上传，“打包上传”，“单文件上传”，“文件目录列表”可用。

## 6.1 打包上传

部署包需要时，在项目根目录下打的 ZIP 包。如果项目为 test1，包含 1.txt、2.txt 文件需要在 test1 目录下打包，包名可以自拟。如图：

```
test
├── 1.txt
├── 2.txt
└── test.zip

0 directories 3 files
```

第一次上传最好是全量包，这样在部署系统中就能保存全量代码，在需要回退时比较方便。如系统已经有全量包，但是文件变更较多，需要再次上传全量包时，可以选“全量包”在上传后系统将会将现有的全量包删除。上传包最大不用超过 80M，如果超过可以打多个 ZIP 包，分此上传，这样最后一次上传的就是全量包。

## 6.2 单文件上传。

如只有一个或少数文件需要上传，可以不用打包直接上传。需要注意如果所传文件在项目跟目录中，需要在“上传文件目录”点输入“.”，多级的目录，不用输入根目录的“/”。

## 6.3 文件目录列表

如需要知道当前系统中全量包的文件树，可以通过“文件目录列表”中选择相应项目进行查看。

# 7 部署操作

部署过程比较简单。默认值均会自动加载，如只是临时修改直接在此页修改即可，所做修改只对本次部署有效。执行顺序如图：

代码打包	部署工作	
部署和回退 <sup>1</sup>	test1 <input type="radio"/> 部署 <input type="radio"/> 回退 <input type="button" value="提交"/>	<sup>4</sup>
单项目设置	<input checked="" type="checkbox"/> 部署完成是否重启服务	<input type="checkbox"/> 重新部署
删项目文件	单线程部署，两次部署间隔10秒	配置替换暂未开启
删部署包	<input checked="" type="radio"/> test1_20130805121053 备注:	未部署
单机检测	<sup>3</sup>	

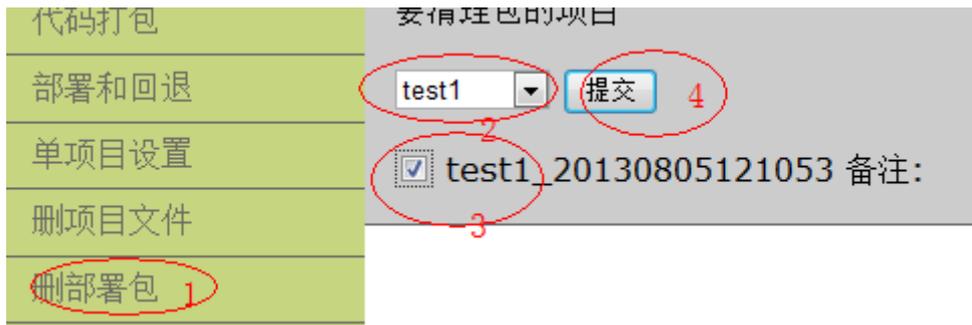
## 8.删除项目文件

在需要对线上文件进行删除时进入“删项目文件”页面，然后选择项目，填写要删除的文件后提交即可。举例要删除项目根目录下 123 目录和根目录下 456 目录中的 456.txt,在删除文件列表中直接输入'123'换行后输入'456/456.txt'，输入完成点击'提交'。删除项目根目录下的文件或文件夹，开头不用打'/'。文件名和目录名中间不能有空格。

## 9.删除部署包

部署操作系统会保留每次的部署代码和信息，时间长了会占用大量磁盘。可以通过“删除部署包”页面进行清理。**需要注意：在清理完部署包，部署的操作记录也会被删除。**

具体操作部署如图：



## 10 单机检测

在部署完毕系统会对部署的文件做 MD5 检查，如果 MD5 检查失败会认为是部署失败，并进行记录。检查此页需要看当前版本是否与最新版本相符。如果是 HTTP 服务，需要上传检查的 URL，可以检查服务是否启动，不能检查业务逻辑。

## 11 环境搭建

未开通，还停留在在手动脚本阶段

## 12.全局设置

## 13.个人信息

可以修改个人的密码，和邮件地址。修改完毕需要重新登录。

## 14.特殊需求

### 14.1 限制 ip 登陆

对于安全考虑，不希望所有地址都能访问部署系统，系统设置有白名单。将可以访问的 IP 地址写到，部署系统根目录的 key\_ip.txt 文件中即可。如果有多个 IP 地址，回车后换行添加即可。

## 14.2 登录信息保存

登录成功后，在没有注销的情况下账户信息可以保存一天。在此期间登录不会在需要输入账户密码。

# 15 注意事项重点说明

## 15.1 回退工作

在部署完新项目后发现有问题需要回退时需要注意。回退是用之前部署的文件进行覆盖，回退后如果有新文件将会保留在当前系统中，如果要删除可以使用“删除文件”功能。回退功能比较复杂，建议在部署系统上没有完整的系统或没有可以覆盖当前版本的老版本时，不要使用回退功能。

## 15.2 删除后的回退工作

在完成“删除文件”操作功能后，不能使用“回退”功能对已删除的文件进行回退。只能重新上传后进行部署操作。