

部署 Web 项目到 Linux 环境

本节要点

掌握 Java Web 项目的部署方式

1. 什么是部署

1.1 前置概念

开发环境: 程序猿写代码用的机器.

测试环境: 测试人员测试程序使用的机器. 一般是退役的部署机器.

部署环境(线上环境): 最终项目发布时所使用的机器. 对稳定性要求很高.

1.2 什么是部署

把程序拷贝安装到部署环境上, 这个过程称为 "部署". 也叫 "上线".

这个过程需要拷贝对应的 .class 文件和依赖的资源文件(各种配置文件, html等)

一旦程序部署成功, 那么这个程序就能被外网中千千万万的普通用户访问到.

部署过程至关重要, 属于程序开发中最重要的一环. 一旦部署出现问题, 极有可能导致严重的事故(服务器不可用之类的).

为了防止部署出错, 一般公司内部都有一些自动化部署工具(如 Jenkins 等). 当前我们先使用手工部署的方式来完成部署.

2. 使用 git 传输代码到服务器

2.1 将本地代码上传到 github 上

过程略, 参考之前的 git 使用过程.

2.2 在 Linux 上下载代码

```
# 进入用户家目录
cd ~
# 创建目录保存项目
mkdir project
cd project
# 使用 git 下载代码
git clone [项目链接]
```

备注:

除了使用 git 来完成代码同步之外, 也可以借助其他工具传输.

同学们可以自行了解 scp 命令和 rz/sz 命令.

3. 数据库建表

按照之前的数据库建表脚本, 在服务器上运行, 建立相同的表结构.

```
mysql -u[数据库用户名] -p[数据库密码] < [数据库脚本].sql
```

4. 构建项目并打 WAR 包

切换到项目的主目录(包含 pom.xml 的目录), 使用 mvn 打包项目

```
mvn package
```

此时在 target 目录下就会生成对应的 war 包.

曾经我们主要是在 IDEA 中通过图形化工具来使用 mvn. 事实上 mvn 本质是一个基于命令行的工具.

曾经在 IDEA 上进行的 mvn 操作在 Linux 中同样也可以通过命令行来完成.

几个常用操作:

1. mvn compile: 构建项目
2. mvn test: 运行单元测试(会先进行构建)
3. mvn package: 打包项目(会先运行单元测试)

5. 拷贝到 Tomcat 中

将 war 包拷贝到 Tomcat 的 webapps 目录中.

```
cp [war包名字] [tomcat webapps目录]
```

稍等片刻, Tomcat 会自动解压缩 war 包, 完成部署.

注意: 如果该项目之前部署过, 需要先删除之前的包.

6. 验证

在浏览器上访问项目的链接. (按照项目的功能进行验证).

1. 验证账户注册登录
2. 验证展示博客列表
3. 验证新增博客
4. 验证展示博客内容
5.

注意事项:

- 要开启 Tomcat 对应端口的安全组(8080)
- 数据库的地址和账号密码可能需要调整
- 可以使用 top 查看系统当前的资源使用情况.
- 还可以把项目链接生成一个二维码(百度搜 "二维码生成"), 就可以用手机扫码访问.