

任务四 项目上线部署发布

1. 前言

1.1 服务器与操作系统

- 服务器是计算机的一种，它比普通计算机运行更快、负载更高、价格更贵。
- 服务器从硬件上等同于电脑PC。而服务器跟PC都是由CPU、内存、主板、硬盘、电源等组成；但服务器的性能要远远超过PC，因为它要保证全年无休。



- 操作系统: 操作系统是作为应用程序与计算机硬件之间的一个接口
 - 没有安装操作系统的计算机，被称为裸机, 如果想在裸机上运行自己的程序，就需要使用机器语言
 - 安装操作系统之后，就可以配置一些高级语言的环境，进行高级语言的开发
- Linux操作系统
 - Linux系统是最具稳定性的系统
 - Linux是天生就比Windows更具安全性
 - 免费, Linux服务器在应用开发上更能节约成本

1.2 项目的发布部署

- 项目的开发流程大致要经过一下几个步骤:
 - 项目立项
 - 需求分析阶段
 - 原型图设计阶段
 - 开发阶段
 - 测试阶段
 - 系统上线

2. 后台项目部署

2.1 安装虚拟机

- 在Linux阶段我们已经安装过了虚拟机,使用的是 Linux操作系统 **CentOS 7** 版本

2.2 安装软件环境

以下软件,在Linux阶段都已安装完成,具体操作详见该阶段安装文档.

软件	版本
JDK	11
Tomcat	8.5
MySQL	5.7

1. 查看Java版本

```
java -version
```

```
[root@localhost ~]# java -version
java version "11.0.7" 2020-04-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.7+8-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.7+8-LTS, mixed mode)
```

2. 查看tomcat是否能够正常启动

```
# 进入到tomcat目录
cd /usr/tomcat/

# 启动tomcat
./bin/startup.sh

# 关闭tomcat
./bin/shutdown.sh
```

3. 关闭防火墙

```
#查看已经开放的端口:
firewall-cmd --list-ports

#开启端口
firewall-cmd --zone=public --add-port=8080/tcp --permanent

#命令含义:
-zone #作用域
-add-port=8080/tcp #添加端口,格式为: 端口/通讯协议
-permanent #永久生效,没有此参数重启后失效

#重启防火墙
firewall-cmd --reload #重启firewall

#关闭防火墙
systemctl stop firewalld.service #停止firewall
systemctl disable firewalld.service #禁止firewall开机启动
firewall-cmd --state #查看默认防火墙状态 (关闭后显示notrunning, 开启后显示running)
```

4. 登录MySQL,检查数据库连接是否正常

-- 创建用户

```
CREATE USER 'JiuYuan'@'%' IDENTIFIED BY 'JiuYuan@123';
```

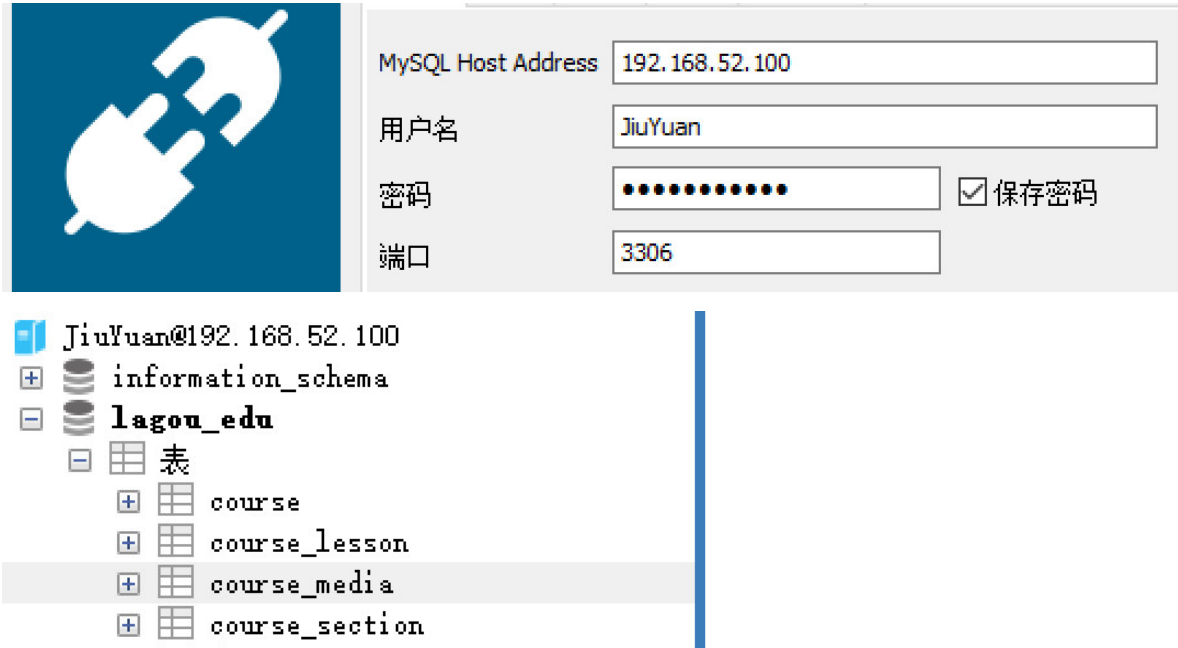
-- 授予所有权限

```
GRANT ALL PRIVILEGES ON *.* TO 'JiuYuan'@'%' IDENTIFIED BY 'JiuYuan@123';
```

-- 刷新

```
FLUSH PRIVILEGES;
```

5. 使用SQLYog连接Linux上的MySQL, 导入SQL脚本 创建项目所需的数据库



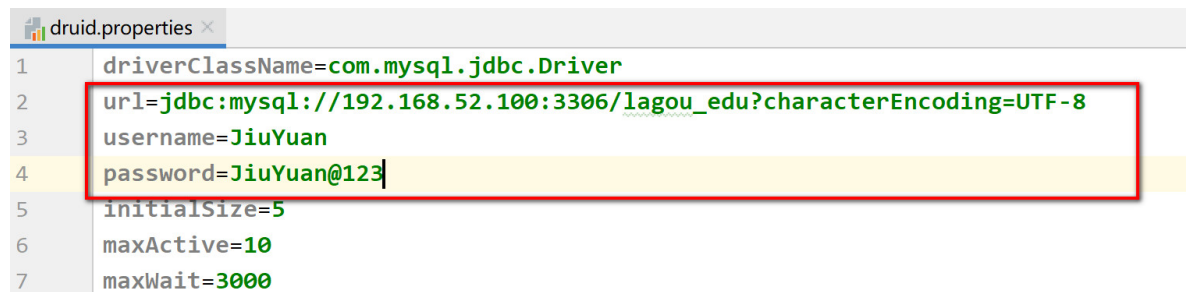
2.3 项目打包 发布

1. 修改项目的数据库配置文件, 数据库的IP, 用户名 密码都要修改.
2. 修改 Constants常量类中的项目URL

//生产环境地址

```
public static final String LOCAL_URL = "http://192.168.52.100:8080";
```

3. 修改后启动项目,测试一下 保证数据库连接没有问题



4. 检查POM文件,打包方式必须是war,编译版本为JDK11

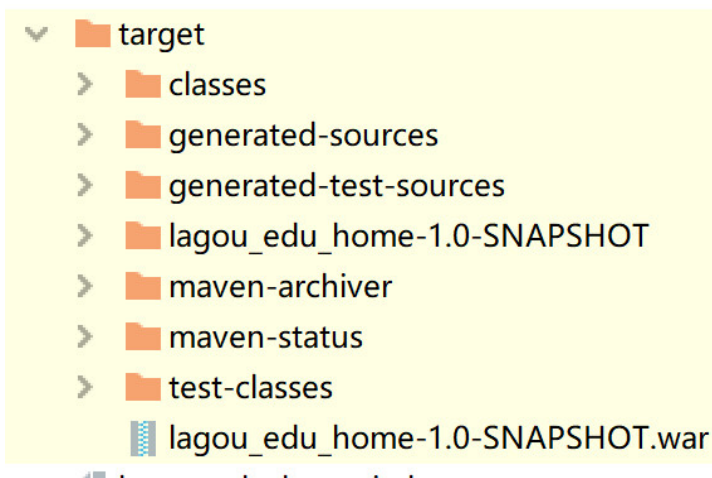
```
<packaging>war</packaging>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>11</maven.compiler.source>
  <maven.compiler.target>11</maven.compiler.target>
</properties>
```

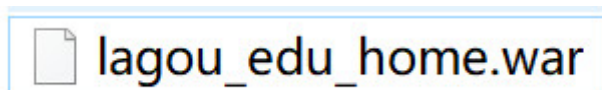
4. 执行打包命令

```
//清除target文件夹
clean
//打包 ,跳过测试
package
```

5. 复制出target目录下的 war包



6. 修改一下项目名称



7. 上传到tomcat中,启动测试

```
[root@localhost webapps]# ll
total 3500
drwxr-x---. 16 root root    4096 Jul 11 08:02 docs
drwxr-x---.  6 root root     78 Jul 11 08:02 examples
drwxr-x---.  5 root root     82 Jul 11 08:02 host-manager
-rw-r--r--.  1 root root 3572056 Jul 23 02:46 lagou_edu_home.war
drwxr-x---.  5 root root     97 Jul 11 08:02 manager
drwxr-x---.  3 root root    4096 Jul 11 08:02 ROOT
```

```
//访问接口
http://192.168.52.100:8080/lagou_edu_home/course?methodName=findCourseList
```

3.前端项目部署

3.1 修改配置文件

1. 前端项目的配置文件有两个,一个是开发环境的配置文件,一个是生产环境的配置文件.



2. 我们先修改一下开发环境文件的 后端服务器访问地址,然后进行一下测试

```
.env.development 文件
VUE_APP_API_BASE = http://192.168.52.100:8080/lagou_edu_home
```

3. 修改生产环境的配置文件

```
// .env.production
VUE_APP_API_BASE = http://192.168.52.100:8080/lagou_edu_home
```

3.2 前端项目打包

1. 修改 **vue.config.js** 配置文件

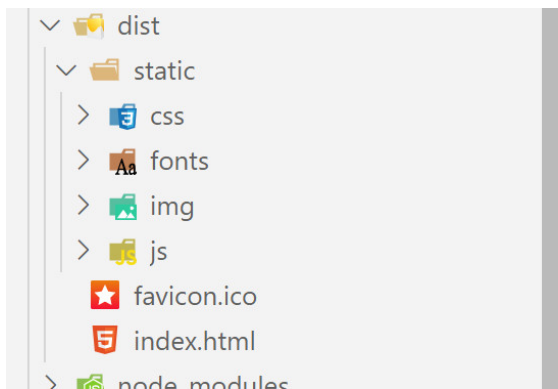
复制下面内容即可

```
module.exports = {
  // relative path for dev
  publicPath: process.env.NODE_ENV === "production" ? "/edu-boss/" : "./",
  // for gh-pages
  indexPath: "index.html",
  assetsDir: "static",
  lintOnSave: process.env.NODE_ENV !== "production",
  productionSourceMap: false,
  css: {
    // sourceMap: process.env.NODE_ENV !== 'production'
  },
  devServer: {
    open: true,
    port: 8081
  }
};
```

2. 执行下面的打包命令

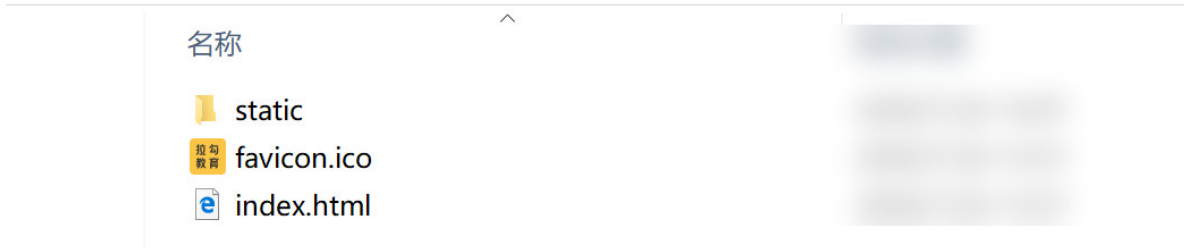
```
npm run build
```

3. 在项目下会生成一个 dist 目录



4. 在本地tomcat的webapps目录下,创建一个edu-boss文件夹,将dist目录中的文件拷贝到里面

APTECH (H:) > software > tomcat-ui > apache-tomcat-8.5.55 > webapps > edu-boss



5. 启动本地tomcat ,访问前端项目 路径为:

`http://localhost:8081/edu-boss/`

3.3 前端项目发布

1. 验证没有问题后,将edu-boos项目压缩,上传到tomcat服务器

```
//复制一份tomcat
cp -r /usr/tomcat/ /usr/tomcat2

//上传 edu-boss.zip ,并解压
unzip edu-boss.zip

//删除edu-boss.zip
rm -rf edu-boss.zip
```

2. 修改tomcat2的server.xml 配置文件,修改3个端口,避免与tomcat1冲突

```
<Server port="8006" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
```

```
<Connector port="8081" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8444" />
```

3. 在部署后端项目的tomcat1的 webapps目录下创建一个 upload文件夹,保存图片

```
webapps]# ll

root root      4096 Jul 11 08:02 docs
root root        78 Jul 11 08:02 examples
root root        82 Jul 11 08:02 host-manager
root root        92 Jul 23 21:05 lagou_edu_home
root root 3572056 Jul 23 02:46 lagou_edu_home.war
root root        97 Jul 11 08:02 manager
root root      4096 Jul 11 08:02 ROOT
root root        56 Jul 23 23:48 upload
```

4. 运行前端项目

```
//进入tomcat2,启动项目
./bin/startup.sh

//动态查看日志
tail -f logs/catalina.out
```

5. 运行后端项目

```
//进入tomcat1,启动项目
./bin/startup.sh

//动态查看日志
tail -f logs/catalina.out
```

6. 前后端都启动后,进行测试