目录

[一、 服务器环境准备 3](#_Toc204638741)

[1. 更新系统 3](#_Toc204638742)

[2. 安装必要软件 3](#_Toc204638743)

[二、 前端部署 3](#_Toc204638744)

[1. 安装Node.js环境 3](#_Toc204638745)

[(1) 安装NodeSource仓库 3](#_Toc204638746)

[(2) 验证安装 3](#_Toc204638747)

[3. 在本地（开发环境）构建Vue项目 4](#_Toc204638748)

[(1) 安装依赖 4](#_Toc204638749)

[(2) 构建生产环境版本 4](#_Toc204638750)

[4. 安装并配置Nginx 4](#_Toc204638751)

[(1) 安装Nginx 4](#_Toc204638752)

[(2) 启动Nginx 4](#_Toc204638753)

[5. 上传前端构建文件 4](#_Toc204638754)

[(1) 在本地将dist目录打包 4](#_Toc204638755)

[(2) 上传到服务器 4](#_Toc204638756)

[(3) 在服务器上解压 4](#_Toc204638757)

[(4) 为Nginx设置文件权限 5](#_Toc204638758)

[(5) 设置SELinux上下文标签 5](#_Toc204638759)

[6. 配置Nginx 5](#_Toc204638760)

[7. 重启Nginx 6](#_Toc204638761)

[三、 后端部署 6](#_Toc204638762)

[1. 安装JDK 11 6](#_Toc204638763)

[(1) 安装OpenJDK 11 6](#_Toc204638764)

[(2) 验证安装 7](#_Toc204638765)

[2. 上传并部署Spring Boot应用 7](#_Toc204638766)

[(1) 创建应用目录 7](#_Toc204638767)

[(2) 将IDEA中打包好的jar文件上传到服务器 7](#_Toc204638768)

[3. 创建Systemd服务 7](#_Toc204638769)

[(1) 创建服务文件以便管理应用 7](#_Toc204638770)

[(2) 添加以下内容 7](#_Toc204638771)

[(3) 启用并启动服务 8](#_Toc204638772)

# 服务器环境准备

## 更新系统

yum update -y

## 安装必要软件

yum install -y wget curl git vim net-tools

# 前端部署

## 安装Node.js环境

### 安装NodeSource仓库

curl -fsSL https://rpm.nodesource.com/setup\_16.x | sudo bash -

sudo yum install -y nodejs

### 验证安装

node -v

npm -v

## 在本地（开发环境）构建Vue项目

### 安装依赖

npm install

### 构建生产环境版本

npm run build

## 安装并配置Nginx

### 安装Nginx

yum install -y epel-release

yum install -y nginx

### 启动Nginx

systemctl start nginx

systemctl enable nginx

## 上传前端构建文件

### 在本地将dist目录打包

tar czvf farm.tar.gz .\dist

### 上传到服务器

scp farm.tar.gz user@server-ip:/tmp/

### 在服务器上解压

mkdir -p /var/www/frontend

tar -xzf /tmp/farm.tar.gz -C /var/www/frontend/

### 为Nginx设置文件权限

chown -R nginx:nginx dist

### 设置SELinux上下文标签

# 设置文件标签为httpd\_sys\_content\_t，允许Web服务器访问

semanage fcontext -a -t httpd\_sys\_content\_t '/var/www/frontend/dist(/.\*)?'

# 应用标签到文件

sudo restorecon -R -v /var/www/frontend/dist/

## 配置Nginx

vim /etc/nginx/conf.d/farm.conf

添加以下内容

|  |
| --- |
| server {  listen 80;  server\_name your-domain.com localhost; # 建议明确指定域名  root /var/www/frontend/dist;  index index.html index.htm;  # 静态资源优化（建议放在前面）  location ~\* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2|ttf|eot)$ {  expires 1y;  add\_header Cache-Control "public, immutable";  try\_files $uri =404;  }  # API 代理  location /api/ {  proxy\_pass http://localhost:8080/; # 确保后端服务在此地址运行  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  }  # 处理Vue Router的history模式 - 放在最后  location / {  try\_files $uri $uri/ /index.html;  }  } |

## 重启Nginx

nginx -t *# 测试配置文件*

systemctl restart nginx

# 后端部署

## 安装JDK 11

### 安装OpenJDK 11

yum install java-11-openjdk-devel -y

### 验证安装

java -version

javac -version

## 上传并部署Spring Boot应用

### 创建应用目录

mkdir -p /opt/springboot-app

chown -R farm:farm /opt/springboot-app

### 将IDEA中打包好的jar文件上传到服务器

*scp target/farm.jar user@server-ip:/opt/springboot-app/*

## 创建Systemd服务

### 创建服务文件以便管理应用

vim /etc/systemd/system/farm.service

### 添加以下内容

|  |
| --- |
| [Unit]  Description=Farm Spring Boot Application  After=network.target  [Service]  Type=simple  User=farm  Group=farm  WorkingDirectory=/opt/springboot-app/Farm  ExecStart=/bin/java -jar /opt/springboot-app/Farm/Farm.jar  Restart=on-failure  RestartSec=10  [Install]  WantedBy=multi-user.target |

### 启用并启动服务

systemctl daemon-reload

systemctl start farm.service

systemctl enable farm.service