

# 软件配置与运维文档

---

## 系统要求

### 硬件要求

- **处理器**: 推荐双核处理器, 2GHz或更高。处理器的性能直接影响系统的响应速度和处理能力, 因此选择高性能的处理器可以确保系统稳定运行。
- **内存**: 最小4GB, 推荐8GB或更高。充足的内存可以提高系统的多任务处理能力, 减少因内存不足导致的性能问题。
- **存储**: 至少500MB可用空间。存储空间主要用于存放应用程序、日志文件和临时数据, 确保有足够的可用空间可以避免磁盘满导致的系统故障。
- **网络**: 支持TCP/IP的网卡, 推荐千兆网卡。高带宽和低延迟的网络连接可以保证系统的数据传输效率, 提升整体性能。

### 软件要求

- **操作系统**: Windows 10/11, macOS, 或主流Linux发行版。选择合适的操作系统版本可以提高系统的兼容性和稳定性, 不同操作系统对应用程序的支持也有所不同。
- **依赖软件**:
  - Node.js:用于前端开发和构建, 提供运行时环境。
  - Java 21:用于后端开发, 支持最新的Java特性和性能改进。
  - vue3:用于构建用户界面, 具有高效、响应式的特点。

### 网络要求

- **局域网环境**:支持多播。确保所有节点能够快速可靠地相互通信, 减少延迟。
- **开放特定端口**: 8080和8081, 这些端口用于前后端应用程序的通信和访问。

## 安装指南

### 网络配置

- **防火墙设置**: 确保防火墙允许应用所需的端口和服务通信, 避免因网络阻断导致的应用无法正常运行。
- **路由器和交换机配置**: 确保网络设备配置正确, 支持局域网中的所有节点, 避免网络瓶颈和数据包丢失。

### 环境变量设置

- **JAVA\_HOME**: 指向Java安装路径, 确保Java相关工具和应用能够找到Java环境。
- **NODE\_HOME**: 指向Node.js安装路径, 确保Node.js相关工具和应用能够找到Node.js环境。
- **其他环境变量**: 根据具体应用需求设置其他必要的环境变量, 确保应用能够正确读取配置信息。

# 配置说明

## 应用配置文件

- 前端配置

- `vue.config.js`: 用于配置Vue项目的打包、代理、路径别名等信息。
- `jest.config.js`: 用于配置Jest测试框架的测试路径、测试环境等信息。
- `babel.config.js`: 用于配置Babel编译器的转码规则、插件等信息。

- 后端配置

- `application.properties`: 用于配置Spring Boot等Java应用的数据库连接、日志级别、端口等信息。

## 运行维护

### 启动应用

1. 启动前端应用: 首先安装必备的依赖 `npm install`, 安装完毕后 `npm run serve` 启动。确保所有依赖项已安装, 启动应用之前先进行依赖项安装, 随后启动服务, 确保前端页面能够正常访问。
2. 启动后端应用: `java -jar your-app.jar`。确保Java环境配置正确, 启动应用后检查日志, 确保服务正常运行并可以响应请求。

### 日常维护任务

- **日志检查**: 定期查看应用日志, 及时发现和解决潜在问题, 确保系统运行稳定。
- **数据库备份**: 定期备份数据库, 防止数据丢失, 并在需要时能够快速恢复数据。
- **性能监控**: 使用监控工具 (如Prometheus和Grafana) 监控系统性能, 及时调整配置以优化系统性能。

### 更新流程

1. 从版本控制系统获取最新代码, 确保包含所有最新的功能和修复。
2. 在测试环境中进行全面测试, 确保代码没有引入新的问题。
3. 更新生产环境, 确保服务不中断, 通过蓝绿部署或滚动更新实现平滑过渡。

## 故障排除

### 常见问题

1. **无法登录**:
  - 检查网络连接, 确保客户端与服务器之间的通信正常。
  - 检查服务是否启动, 确保后端服务正常运行。
2. **消息发送失败**:
  - 检查后端服务日志, 确定问题的具体原因。
  - 确认前后端通信正常, 排除网络或配置问题。

## 日志分析

- 使用日志分析工具（如ELK：Elasticsearch, Logstash, Kibana）集中收集、存储和分析日志，快速定位和解决问题。

## 性能调优

- 数据库调优**：定期清理和优化数据库索引
- 代码优化**：定期审查和优化代码

## 备份和恢复

### 数据备份

- 定期备份数据库和应用配置文件，确保在数据丢失或系统崩溃时能够快速恢复。

### 数据恢复

- 使用备份文件进行数据恢复，确保最小化数据丢失，快速恢复系统运行。

## 安全措施

### 安全审计

- 定期进行安全审计，检查系统的安全配置和潜在漏洞。
- 使用安全工具（如OWASP ZAP）进行安全扫描，发现并修复安全漏洞。

## 监控和日志

### 监控设置

- 配置Prometheus和Grafana进行系统监控，实时监测关键性能指标。
- 设置关键指标的告警阈值，及时通知运维人员处理异常情况。

### 日志管理

- 配置日志轮转，避免日志文件过大影响系统性能。
- 定期备份和清理日志文件，确保日志数据的可用性和安全性。

### 告警配置

- 配置邮件或短信告警系统，确保在出现异常时及时通知相关运维人员。

## 升级流程

### 版本升级准备

- 检查新版本的兼容性说明，确保新版本与现有系统兼容。
- 在测试环境中进行全面测试，确保新版本没有重大问题。
- 制定回滚计划，确保在升级失败时能够快速恢复。

## 执行升级

1. 停止生产环境应用，确保数据一致性。
2. 部署新版本，按照预定的升级步骤进行操作。
3. 验证新版本运行正常，确保所有功能正常使用。

## 升级后检查

- 检查系统日志，确保没有错误或异常。
- 进行功能测试，确保系统正常运行，所有功能无误。

## 文档维护

- 定期检查并更新本文档，确保内容准确、完整。
- 记录所有配置变更和维护操作，保持操作的可追溯性。
- 保持一个已知问题和解决方案的知识库，方便快速解决常见问题。