

福建水务营收系统-部署设计

系统设计团队

2024年12月19日

- 1 [福建水务营收系统部署设计文档](#)
 - 1.1 [文档信息](#)
 - 1.2 [章节导航 \(精简\)](#)
 - 1.2.1 [部署目标](#)
 - 1.2.2 [部署原则](#)
 - 1.2.3 [银行文件传输配置部署约束](#)
 - 1.2.4 [总体部署架构](#)
 - 1.2.5 [环境架构设计](#)
 - 1.2.5.1 [生产环境架构](#)
 - 1.2.5.2 [灾备环境架构](#)
 - 1.2.6 [硬件配置要求](#)
 - 1.2.6.1 [负载均衡服务器](#)
 - 1.2.6.2 [Web服务器](#)
 - 1.2.6.3 [应用服务器](#)
 - 1.2.6.4 [数据库服务器](#)
 - 1.2.7 [软件配置要求](#)
 - 1.2.7.1 [操作系统](#)
 - 1.2.7.2 [应用环境](#)
 - 1.2.7.3 [中间件配置](#)
 - 1.2.8 [容器化部署方案](#)
 - 1.2.8.1 [核心服务组件](#)
 - 1.2.8.2 [部署配置说明](#)
 - 1.2.9 [部署流程](#)
 - 1.2.9.1 [环境准备阶段](#)
 - 1.2.9.2 [应用部署阶段](#)
 - 1.2.9.3 [测试验证阶段](#)
 - 1.2.10 [版本管理和发布](#)
 - 1.2.10.1 [版本管理策略](#)
 - 1.2.10.2 [发布流程](#)
 - 1.2.11 [网络架构安全](#)
 - 1.2.11.1 [网络分区设计](#)
 - 1.2.11.2 [访问控制策略](#)
 - 1.2.12 [应用安全](#)

- [1.2.12.1 身份认证安全](#)
- [1.2.12.2 数据传输安全](#)
- [1.2.13 数据安全](#)
 - [1.2.13.1 数据备份策略](#)
 - [1.2.13.2 数据恢复方案](#)
- [1.2.14 系统监控](#)
 - [1.2.14.1 基础设施监控](#)
 - [1.2.14.2 应用监控](#)
 - [1.2.14.3 数据库监控](#)
- [1.2.15 运维管理](#)
 - [1.2.15.1 日常运维](#)
 - [1.2.15.2 故障处理](#)
 - [1.2.15.3 变更管理](#)
- [1.2.16 运维工具](#)
 - [1.2.16.1 自动化运维](#)
 - [1.2.16.2 运维平台](#)

1 福建水务营收系统部署设计文档

1.1 文档信息

项目信息	详情
项目名称	福建水务营收系统
文档类型	概要设计文档
技术框架	RuoYi-Vue-Pro + yudao-ui-admin-vue3
文档版本	v1.0
编写日期	2024-12-19
文档状态	已完成

1.2 章节导航 (精简)

- [部署概述](#)
- [部署架构](#)
- [环境配置](#)
- [部署方案](#)
- [网络安全](#)
- [监控运维](#)

- [总结](#)

部署概述

福建水务营收系统采用集中化部署模式，基于集团私有云环境进行部署，构建高可用、可扩展的系统架构，满足集团及下属各水务公司的业务需求。

1.2.1 部署目标

- 实现系统的集中部署和统一管理
- 确保系统高可用性和可靠性
- 支持业务动态扩展和弹性伸缩
- 保障数据安全和业务连续性
- 降低运维成本，提高管理效率

1.2.2 部署原则

- **集中部署**：核心业务系统集中部署在集团数据中心，统一管理和维护
- **多级架构**：采用多级架构设计，实现前端负载分担和后端高可用
- **灾备设计**：重要系统和数据实现灾备，确保业务连续性
- **安全防护**：多层次安全防护体系，确保系统和数据安全
- **弹性扩展**：支持系统横向和纵向扩展，适应业务增长需求

1.2.3 银行文件传输配置部署约束

- `sw-business-bank` 继续沿用 `application-{profile}.yaml + Nacos` 加载环境默认文件传输规则，不在代码中硬编码送盘、回盘、对账目录。
- 环境默认规则承接
`protocol/host/port/username/credentialRef/sendDir/backDir/reconcileDir/archiveDir/localTempDir` ;
银行通道、租户、租户-通道覆盖通过 `bk_channel_api_config` 的 `FILE_TRANSFER_CONFIG` 记录补充。
- 配置切换只影响新发起文件动作；已落库批次继续沿用已固化的协议、目录和文件路径审计结果，不要求重算旧批次。

部署架构

1.2.4 总体部署架构

```

graph
    subgraph EXTERNAL["外部用户访问"]
        direction TB
        A[移动端用户]
        B[PC端用户]
        C[第三方系统]
    end

    subgraph LOADBALANCER["负载均衡层"]
        direction TB
        D[主负载均衡器]
        E[备负载均衡器]
    end

    subgraph WEBSERVICE["Web服务层"]
        direction TB
        F[Web服务器1]
        G[Web服务器2]
        H[Web服务器3]
    end

    subgraph APPSERVICE["应用服务层"]
        direction TB
        I[应用服务器1]
        J[应用服务器2]
        K[应用服务器3]
        L[应用服务器4]
    end

    subgraph DATASERVICE["数据服务层"]
        direction TB
        M[(主数据库<br/>达梦数据库 8.0+)]
        N[(备数据库<br/>达梦数据库 8.0+)]
        O[缓存服务器<br/>Redis]
        P[文件服务器<br/>MinIO]

        M -.->|数据同步| N
    end

    %% 层级间调用关系
    EXTERNAL -.->|用户请求<br/>接口调用| LOADBALANCER
    LOADBALANCER -.->|负载均衡<br/>故障切换| WEBSERVICE
    WEBSERVICE -.->|业务请求<br/>服务调用| APPSERVICE
    APPSERVICE -.->|数据访问<br/>缓存读写<br/>文件操作| DATASERVICE

```

1.2.5 环境架构设计

系统部署分为四个环境：开发环境、测试环境、预生产环境和生产环境。

1.2.5.1 生产环境架构

- **负载均衡**：双机热备，支持故障自动切换
- **Web服务器**：3台服务器，提供前端页面服务
- **应用服务器**：4台服务器集群，处理业务逻辑
- **达梦数据库**：主备架构，数据实时同步
- **Redis缓存**：主从架构，提高系统响应速度
- **MinIO文件存储**：分布式文件存储，支持高并发访问

1.2.5.2 灾备环境架构

- **异地部署**：在备用数据中心部署完整的系统环境
- **数据同步**：定时同步生产环境数据
- **切换机制**：支持快速切换到灾备环境

环境配置

1.2.6 硬件配置要求

1.2.6.1 负载均衡服务器

- **CPU**：8核 2.4GHz
- **内存**：16GB
- **存储**：100GB SSD
- **网络**：双千兆网卡

1.2.6.2 Web服务器

- **CPU**：8核 2.4GHz
- **内存**：16GB
- **存储**：200GB SSD
- **网络**：双千兆网卡

1.2.6.3 应用服务器

- **CPU**：16核 2.4GHz
- **内存**：32GB
- **存储**：300GB SSD
- **网络**：双千兆网卡

1.2.6.4 数据库服务器

- **CPU**：24核 2.4GHz
- **内存**：64GB
- **存储**：2TB SSD (RAID 10)
- **网络**：双万兆网卡

1.2.7 软件配置要求

1.2.7.1 操作系统

- **服务端**：CentOS 7.9+ 或 openEuler 20.03+
- **数据库**：达梦数据库 8.0+ 企业版

1.2.7.2 应用环境

- **Java运行环境**：JDK 17
- **应用服务器**：内置Tomcat 9.0
- **Web服务器**：Nginx 1.20+
- **缓存服务器**：Redis 6.2+

1.2.7.3 中间件配置

- 消息队列：基于Redis的轻量级队列
- 任务调度：Quartz集群模式
- 工作流引擎：Flowable 6.8.0
- 文件存储：MinIO 分布式存储
- 配置中心：Nacos，承接 profile 对应的银行文件传输默认规则

部署方案

1.2.8 容器化部署方案

系统采用Docker容器化部署，使用Docker Compose进行服务编排。

1.2.8.1 核心服务组件

```
graph
  direction TB
  subgraph FRONTEND["前端服务层"]
    A[前端容器<br/>Nginx + Vue3]
  end

  subgraph BACKEND["后端服务层"]
    B[后端容器<br/>Spring Boot]
  end

  subgraph DATASTORAGE["数据存储层"]
    direction TB
    C[数据库容器<br/>达梦数据库 8.0+]
    D[缓存容器<br/>Redis]
    E[文件存储容器<br/>MinIO]
  end

  %% 容器服务调用关系
  FRONTEND -.->|API请求<br/>静态资源| BACKEND
  BACKEND -.->|数据操作<br/>缓存访问<br/>文件存储| DATASTORAGE
```

1.2.8.2 部署配置说明

- 前端容器：包含Vue3应用和Nginx反向代理
- 后端容器：Spring Boot应用，连接数据库和缓存
- 数据库容器：达梦数据库，数据持久化存储
- 缓存容器：Redis缓存服务，提升系统性能
- 文件存储容器：MinIO对象存储，管理系统文件
- 银行文件交换目录：应用节点需具备本地临时目录与归档目录访问权限；外部银行 SFTP/FTP 白名单、账户与凭据由环境运维统一开通

1.2.9 部署流程

1.2.9.1 环境准备阶段

1. 服务器准备：配置服务器硬件和网络环境
2. 操作系统安装：安装和配置操作系统
3. Docker环境部署：安装Docker和Docker Compose

4. **网络配置**: 配置服务器网络和防火墙规则

1.2.9.2 应用部署阶段

1. **镜像构建**: 构建应用Docker镜像
2. **服务编排**: 配置Docker Compose服务编排
3. **数据初始化**: 初始化数据库表结构和基础数据
4. **服务启动**: 启动所有容器化服务

1.2.9.3 测试验证阶段

1. **功能测试**: 验证系统各功能模块正常运行
2. **性能测试**: 测试系统并发处理能力
3. **安全测试**: 验证系统安全防护措施
4. **集成测试**: 测试与外部系统的接口集成

1.2.10 版本管理和发布

1.2.10.1 版本管理策略

- **Git版本控制**: 使用Git管理代码版本
- **分支管理**: 采用GitFlow分支管理模型
- **标签管理**: 为每个发布版本打标签

1.2.10.2 发布流程

1. **代码合并**: 将开发分支合并到主分支
2. **构建镜像**: 自动构建Docker镜像
3. **环境部署**: 自动部署到目标环境
4. **健康检查**: 验证服务健康状态

网络安全

1.2.11 网络架构安全

1.2.11.1 网络分区设计

- **DMZ区**: 部署面向外部的Web服务, 与内网隔离
- **应用区**: 部署业务应用服务, 限制网络访问
- **数据区**: 部署数据库和文件服务, 严格访问控制
- **管理区**: 部署运维管理工具, 独立网络环境

1.2.11.2 访问控制策略

- **防火墙规则**: 配置严格的防火墙访问规则
- **端口管理**: 只开放必要的服务端口
- **IP白名单**: 对敏感服务配置IP访问白名单

1.2.12 应用安全

1.2.12.1 身份认证安全

- **多因子认证**: 支持密码+短信验证码双因子认证
- **会话管理**: 安全的会话创建、维护和销毁机制
- **密码策略**: 强密码策略和定期密码更换

1.2.12.2 数据传输安全

- **HTTPS通信**: 所有外部通信使用HTTPS加密
- **API安全**: API接口使用JWT令牌认证
- **数据加密**: 敏感数据传输和存储加密

1.2.13 数据安全

1.2.13.1 数据备份策略

- **自动备份**: 每日自动增量备份, 每周全量备份
- **异地存储**: 备份数据异地存储, 防止灾难性损失
- **备份验证**: 定期验证备份数据的完整性和可用性

1.2.13.2 数据恢复方案

- **快速恢复**: 提供数据快速恢复机制
- **恢复演练**: 定期进行数据恢复演练
- **灾难恢复**: 制定完整的灾难恢复预案

监控运维

1.2.14 系统监控

1.2.14.1 基础设施监控

- **服务器监控**: CPU、内存、磁盘、网络等资源监控
- **网络监控**: 网络连通性和带宽使用情况监控
- **存储监控**: 磁盘空间和IO性能监控

1.2.14.2 应用监控

- **服务状态监控**: 监控各服务的运行状态
- **性能监控**: 监控应用响应时间和吞吐量
- **业务监控**: 监控关键业务指标和异常情况
- **日志监控**: 实时监控应用日志和错误信息

1.2.14.3 数据库监控

- **数据库性能监控**: 监控SQL执行效率和资源使用
- **连接监控**: 监控数据库连接数和连接状态
- **存储监控**: 监控数据库存储空间和增长趋势

1.2.15 运维管理

1.2.15.1 日常运维

- **系统巡检**: 定期检查系统运行状态

- **性能优化**: 根据监控数据优化系统性能
- **容量规划**: 预测资源需求, 提前进行容量规划
- **安全检查**: 定期进行安全风险评估和加固

1.2.15.2 故障处理

- **故障监控**: 7×24小时故障监控和告警
- **应急响应**: 建立快速响应机制和处理流程
- **故障分析**: 详细记录和分析故障原因
- **预防措施**: 制定故障预防和改进措施

1.2.15.3 变更管理

- **变更审批**: 建立变更申请和审批流程
- **风险评估**: 评估变更对系统的影响和风险
- **回滚预案**: 为每次变更制定回滚预案
- **变更记录**: 详细记录所有系统变更操作

1.2.16 运维工具

1.2.16.1 自动化运维

- **部署自动化**: 自动化应用部署和配置管理
- **监控自动化**: 自动化监控配置和告警处理
- **备份自动化**: 自动化数据备份和验证
- **巡检自动化**: 自动化系统健康检查

1.2.16.2 运维平台

- **统一运维平台**: 提供统一的系统管理界面
- **监控大屏**: 实时展示系统运行状态
- **告警中心**: 集中管理和处理系统告警
- **知识库**: 建立运维知识库和操作手册

总结

部署设计以“集中部署、分层防护、可观测运维”为主线, 覆盖了环境规划、架构设计、部署实施与运维治理。后续若进入实施细节阶段, 建议将脚本、参数模板和操作手册沉淀到附录或运维手册, 不在本主文档中重复维护。