

# RWADurian 2.0 系统部署与运维成本方案

版本: 1.0 日期: 2026年1月22日 编制: 技术部

## 一、项目概述

### 1.1 系统简介

RWADurian 2.0 是一套基于微服务架构的数字资产挖矿与交易系统。用户通过贡献算力参与挖矿，系统按秒计算并分配资产份额，同时支持资产交易、销毁等功能。

### 1.2 业务场景

#### 场景一：实时挖矿与资产展示

用户通过安卓 App 参与挖矿，系统需要：

- 按秒计算挖矿收益**：根据用户算力占比，每秒计算并分配资产份额
- 实时推送资产变化**：用户在 App 上能实时看到资产数量和价值的变化
- 长连接保持**：用户打开 App 期间，通过 WebSocket 长连接保持数据同步
- 不间断运行**：挖矿服务全年 7×24 小时不中断

#### 场景二：资产交易与K线行情

用户可在平台内进行资产买卖：

- 实时K线行情**：展示分钟/小时/日K线图表
- 销毁记录实时更新**：每笔交易产生的销毁量实时显示
- 价格实时刷新**：资产价格随交易实时变动

### 1.3 需部署服务清单

服务名称	路径	功能描述	端口
mining-admin-web	frontend/mining-admin-web	管理后台前端	3100
auth-service	backend/services/auth-service	用户认证与授权	3024
contribution-service	backend/services/contribution-service	算力/贡献值管理	3020
mining-service	backend/services/mining-service	挖矿核心服务	3021
trading-service	backend/services/trading-service	交易与行情服务	3022
mining-admin-service	backend/services/mining-admin-service	管理后台API	3023

### 1.4 技术栈

组件	技术
前端	Next.js 14 + React 18 + Tailwind CSS + ECharts
后端	NestJS 10 (全部5个服务)
数据库	PostgreSQL 15

缓存	Redis 7
消息队列	Kafka
实时通信	Socket.io (WebSocket)

## 二、数据存储与带宽需求分析

### 2.1 数据存储需求

#### 核心数据结构

数据类型	单条大小	产生频率	说明
MiningRecord	152 bytes	每用户每分钟1条	挖矿记录
MiningTransaction	320 bytes	每笔交易1条	资产流水
BurnRecord	216 bytes	每分钟1条	销毁记录
MinuteKLine	152 bytes	每分钟1条	K线数据

#### 存储量估算 (按 10,000 活跃用户计算)

时间范围	原始数据量	含索引 (×1.5)
每日	~1.1 GB	~1.7 GB
每月	~33 GB	~50 GB

### 2.2 带宽需求

#### 实时推送数据量 (单用户)

推送类型	数据大小	推送频率	带宽占用
价格更新	200 bytes	每秒1次	1.6 kbps
余额更新	495 bytes	每秒1次	3.96 kbps
销毁/K线	425 bytes	每分钟1次	0.1 kbps
单用户合计			5.7 kbps

#### 带宽承载能力

带宽规格	理论承载	实际承载 (预留30%)
10 Mbps	1,754 用户	~1,200 用户
50 Mbps	8,771 用户	~6,000 用户
100 Mbps	17,543 用户	~12,000 用户

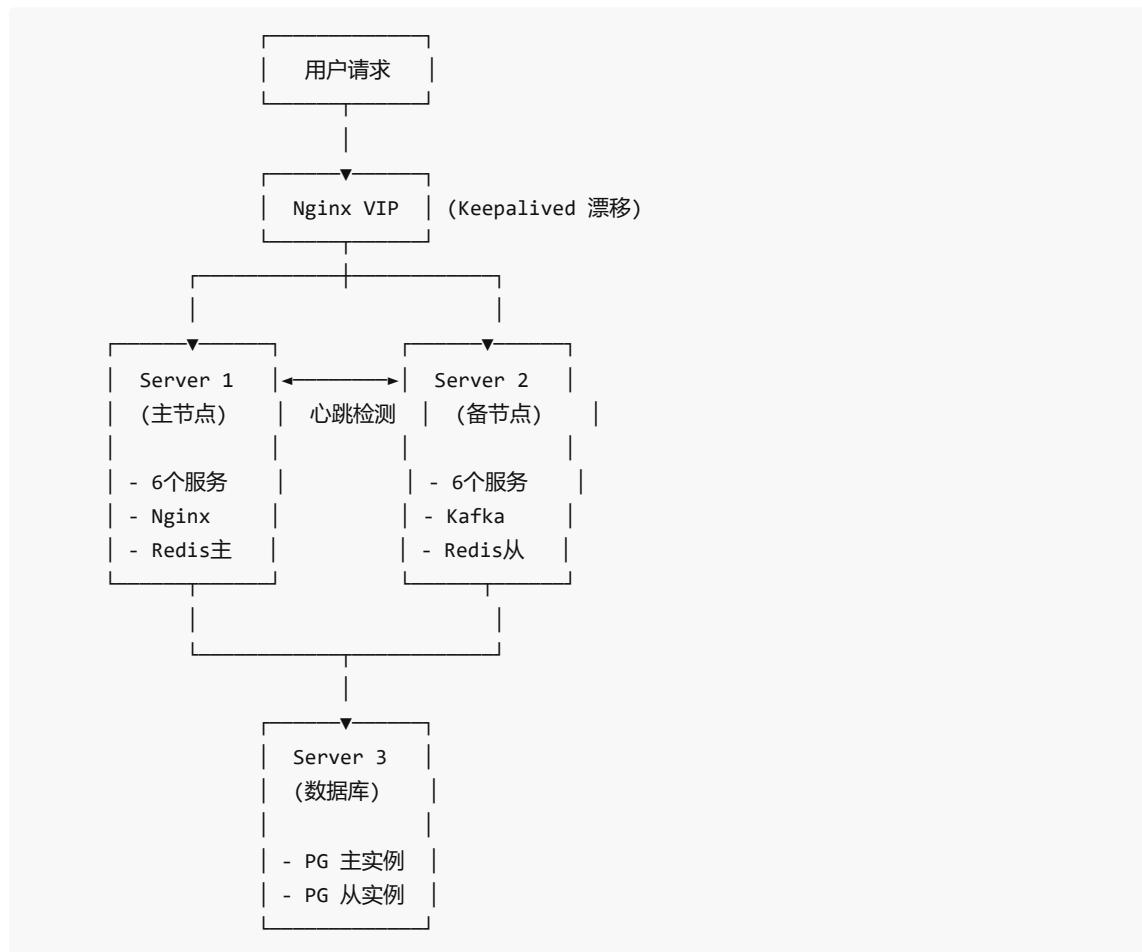
### 三、方案一：高可用架构

#### 3.1 适用场景

要求全年每一秒钟都不中断挖矿与销毁记录，用户通过安卓手机实时长连接，按秒显示资产数量与价值。

- 挖矿服务 7×24 不间断运行
- 用户 App 实时展示资产变化
- 主备自动切换，故障无感知

#### 3.2 架构设计



#### 3.3 服务器配置

服务器	角色	部署内容
Server 1	应用主节点	6个服务 + Nginx + Redis主
Server 2	应用备节点	6个服务(热备) + Kafka + Redis从
Server 3	数据库节点	PostgreSQL 主从复制

#### 3.4 高可用机制

故障场景	恢复策略	恢复时间
Server 1 宕机	Keepalived 自动切换至 Server 2	< 30 秒
Server 2 宕机	Server 1 继续服务	无影响
Server 3 宕机	从实例提升为主	< 5 分钟
单服务崩溃	PM2/Systemd 自动重启	< 10 秒

### 3.5 月度成本

费用项目	规格	月租金
应用服务器 ×2	主备双活	¥7,000
数据库服务器 ×1	PostgreSQL 主从	¥3,500
带宽	100 Mbps × ¥100/M	¥10,000
公网 IP	3 个	¥150
备份 + 监控	-	¥300
<b>合计</b>		<b>¥20,950/月</b>

### 3.6 支撑能力

指标	数值
同时在线用户	~12,000
实时推送频率	每秒
可用性	99.9%+

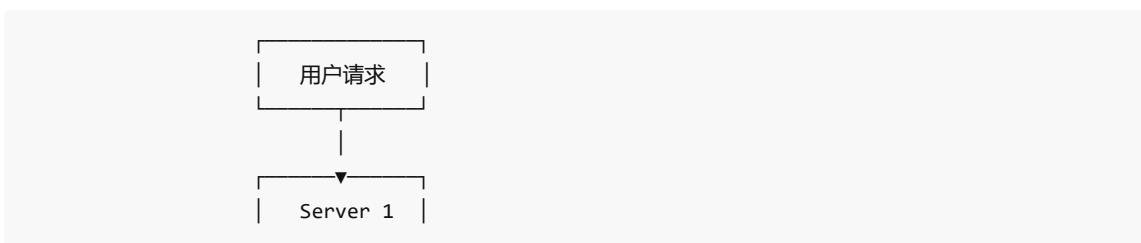
## 四、方案二：标准架构

### 4.1 适用场景

不考虑挖矿是否中断，中断后重启继续即可；不要求按秒实时显示，但销毁记录和K线需要实时刷新。

- 允许短暂服务中断
- K线行情、销毁记录实时更新
- 资产余额可通过轮询刷新

### 4.2 架构设计





#### 4.3 服务器配置

服务器	角色	部署内容
Server 1	全功能节点	6个服务 + 全部中间件 + 数据库

#### 4.4 月度成本

费用项目	规格	月租金
服务器 × 1	应用 + 数据库	¥3,500
带宽	10 Mbps × ¥100/M	¥1,000
公网 IP	1 个	¥50
机柜托管	2U	¥400
备份 + 监控	-	¥200
<b>合计</b>		<b>¥5,150/月</b>

#### 4.5 支撑能力

指标	数值
同时在线用户	~1,200
K线/销毁刷新	实时
可用性	99%+

### 五、方案对比

维度	方案一 (高可用)	方案二 (标准)
适用场景	秒级实时推送, 不中断	K线/销毁实时, 允许中断
月成本	¥20,950	¥5,150
服务器数量	3 台	1 台
带宽	100 Mbps	10 Mbps
同时在线用户	~12,000	~1,200

实时性	秒级推送	分钟级/轮询
可用性	99.9%	99%
挖矿中断处理	自动切换, 不中断	需手动重启

## 六、成本汇总

方案	月成本
方案一 (高可用)	¥20,950
方案二 (标准)	¥5,150
差额	¥15,800

## 七、建议

### 7.1 初期阶段

建议采用 **方案二 (标准架构)** :

- 月成本仅 ¥5,150, 降低初期投入
- 满足初期用户规模需求 (~1,200 在线)
- 快速上线验证业务模式
- K线和销毁记录仍可实时刷新

### 7.2 扩展阶段

当满足以下条件时, 升级至 **方案一 (高可用架构)** :

- 同时在线用户超过 1,000
- 用户对实时性要求提高
- 业务模式验证成功, 进入增长期

## 八、附录

### 8.1 服务端口分配

服务	端口
mining-admin-web	3100
contribution-service	3020
mining-service	3021
trading-service	3022
mining-admin-service	3023
auth-service	3024

PostgreSQL	5432
Redis	6379
Kafka	9092

## 8.2 数据库分配

服务	数据库名	Redis DB
auth-service	rwa_auth	14
contribution-service	rwa_contribution	0
mining-service	mining_db	1
trading-service	trading_db	2
mining-admin-service	mining_admin_db	3

---

文档结束