

K8s集群资源使用率统计分析工具 - 开发文档

文档信息

- **版本**: 1.0.0
- **创建日期**: 2026-02-25
- **技术栈**: Python 3.10 + FastAPI + openpyxl
- **作者**: 骆玉宝
- **邮箱**: ybluo@alauda.io
- **使用AI工具**: OpenCode (big-pickle模型)

1. 项目概述

1.1 背景与目标

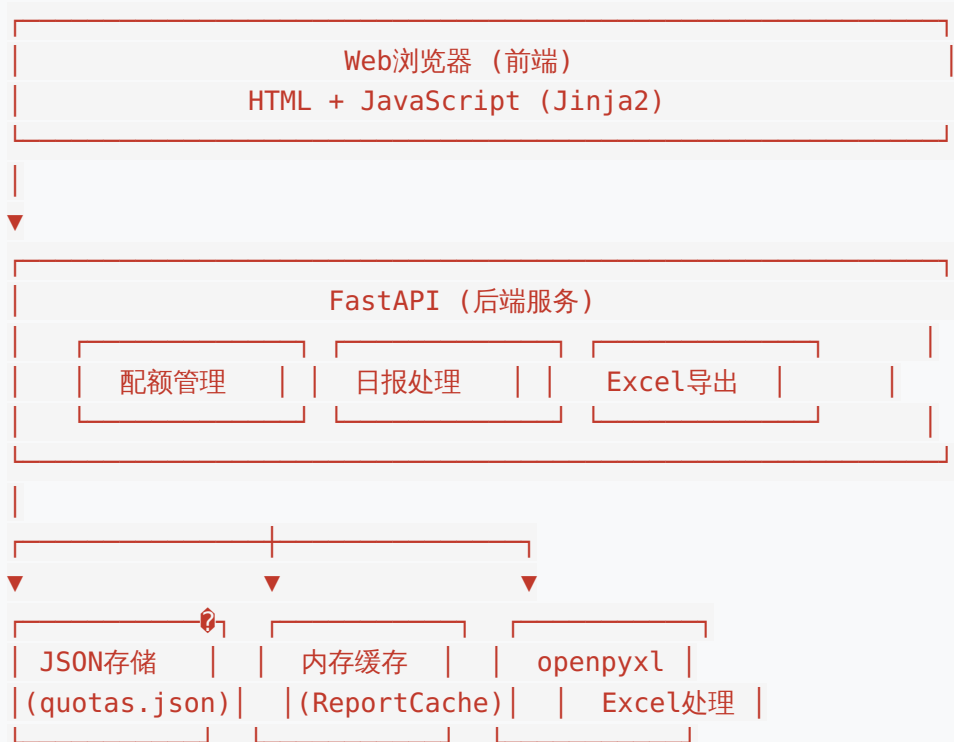
随着Kubernetes集群规模的不断扩大，对集群资源使用情况进行有效监控和分析变得尤为重要。本工具旨在实现多集群、多维度的资源使用数据自动化统计与分析，支撑项目级资源配额使用率的日常监控与管理。

1.2 核心功能

- 资源层级管理**: 项目 → 命名空间的层级关系，支持CPU、内存配额配置
- 日报数据解析**: 从文件读取各命名空间的CPU、内存使用数据
- 使用率计算**: 按项目汇总资源使用量，结合配额计算使用率
- 数据导出**: 生成标准化Excel报告

2. 系统架构

2.1 技术架构



2.2 项目结构

```
k8s-resource-analyzer/
├── app/
│   ├── __init__.py          # 包初始化
│   ├── main.py              # FastAPI入口、路由定义
│   ├── models.py            # Pydantic数据模型
│   ├── storage.py           # JSON存储与内存缓存
│   ├── parser.py            # 日报文件解析器
│   ├── quota_manager.py     # 配额管理 (CRUD+Excel导入)
│   ├── calculator.py        # 使用率计算逻辑
│   └── exporter.py          # Excel导出功能
├── templates/
│   └── index.html           # 前端页面
├── data/
│   └── quotas.json          # 配额持久化存储
└── uploads/                 # 上传文件临时目录
```

```
└─ requirements.txt      # Python依赖
└─ README.md             # 项目说明
```

3. 安装与运行

3.1 环境要求

- Python 3.10+
- pip 包管理器

3.2 安装步骤

```
# 1. 进入项目目录
cd k8s-resource-analyzer

# 2. 安装依赖
pip install -r requirements.txt

# 3. 启动服务
uvicorn app.main:app --host 0.0.0.0 --port 8000
```

3.3 访问方式

- 本地访问: `http://localhost:8000`
- 局域网访问: `http://<服务器IP>:8000`

4. 使用指南

4.1 配额管理

4.1.1 导入配额

点击"下载模板"获取配额导入模板, 模板格式如下:

| 云序号 | 项目名称 | CPU配额(核) | 内存配额(GB) |
|-----|------|----------|----------|
|-----|------|----------|----------|

| | | | |
|----------|-----|----|----|
| inst_xxx | 项目A | 10 | 50 |
|----------|-----|----|----|

注意事项：

- 云序号 + 项目名称为唯一键
- 支持增量导入（相同键更新，不同键新增）

4.1.2 手动修改配额

在配额列表中，每个项目都有"修改"按钮，点击后可以单独修改该项目的：

- 项目名称
- CPU配额（核）
- 内存配额（GB）

修改完成后点击"保存"即可生效。

4.1.3 删除配额

在配额列表中点击"删除"按钮即可移除对应项目的配额信息。

4.2 日报上传

4.2.1 文件格式要求

文件名格式：_Day_report_YYYY-MM-DD （包含日期即可）

文件内容格式：

项目名称;命名空间名称;CPU使用总量(核);内存使用总量(bytes)

示例项目A;namespace-a;3.5;16106127360

示例项目A;namespace-b;2.1;9663676416

4.2.2 上传方式

1. 点击"下载模板"获取示例文件
2. 点击上传区域或拖拽文件到上传区域
3. 支持同时上传多个文件
4. 同日期数据会自动累加

4.3 报表查看

1. 切换到"报表查看"标签
2. 选择日期
3. 查看各项项目的CPU/内存使用量及使用率
4. 使用率按颜色区分：高($\geq 80\%$)、中($\geq 50\%$)、低($< 50\%$)

4.4 导出报告

- 1. 切换到"导出"标签
 - 2. 勾选要导出的日期
 - 3. 点击"导出Excel"按钮
- 导出文件格式：

| 云序号 | 日期 | cpu使用率 | 内存使用率 |
|----------|----------|--------|--------|
| inst_xxx | 2026/2/9 | 0.8000 | 0.6500 |

注意：

- 使用率为小数形式（如80% = 0.8000）
- 保留4位小数
- 自动过滤无配额的项目

5. API接口文档

5.1 配额管理

| 方法 | 路径 | 说明 |
|--------|------------------------|-----------|
| GET | /api/quotas | 获取所有配额列表 |
| POST | /api/quotas | 添加单个配额 |
| PUT | /api/quotas/{cloud_id} | 修改配额 |
| DELETE | /api/quotas/{cloud_id} | 删除配额 |
| POST | /api/quotas/import | Excel导入配额 |

5.2 日报管理

| 方法 | 路径 | 说明 |
|------|---------------------|--------|
| POST | /api/reports/upload | 上传日报文件 |

| | | |
|-----|---------------------|-----------|
| GET | /api/reports | 获取已上传日期列表 |
| GET | /api/reports/{date} | 获取指定日期数据 |
| GET | /api/reports/export | 导出Excel报告 |

5.3 模板下载

| 方法 | 路径 | 说明 |
|-----|-----------------------|----------|
| GET | /api/templates/quota | 下载配额导入模板 |
| GET | /api/templates/report | 下载日报模板 |

6. 核心模块说明

6.1 models.py - 数据模型

```
class Quota(BaseModel):
    cloud_id: str          # 云序号 ()
    project_name唯一键: str    # 项目名称
    cpu_quota: float        # CPU配额 (核)
    memory_quota: float     # 内存配额 (GB)

class ProjectUsage(BaseModel):
    project_name: str
    cloud_id: Optional[str]
    cpu_usage: float
    memory_usage: float
    cpu_rate: Optional[float]
    memory_rate: Optional[float]
```

6.2 parser.py - 日报解析

- 从文件名提取日期（支持多种格式）
- 解析文本格式的日报数据
- 字节单位转换为GB

6.3 calculator.py - 计算器

- 按项目聚合命名空间使用量
- 计算CPU/内存使用率
- 使用率上限100%

6.4 exporter.py - 导出器

- 生成Excel工作簿
- 日期格式转换 (Y/M/D)
- 使用率转换为小数格式

7. 部署建议

7.1 生产环境部署

```
# 使用Gunicorn替代uvicorn
pip install gunicorn
gunicorn -w 4 -k uvicorn.workers.UvicornWorker app.main:app --bind 0.0.0.0:8000
```

7.2 Nginx反向代理（可选）

```
server {
    listen 80;
    server_name your-domain.com;

    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

8. 未来优化方向

8.1 自动获取K8s配额

目标：通过Token+API对接K8s集群，自动获取项目配额，无需手动维护

实现方案：

1. 配置K8s集群访问凭证（kubeconfig或Token）
2. 调用Kubernetes API获取Namespace资源配额
3. 定时同步配额数据到本地存储
4. 区分自动获取的配额与手动维护的配额

涉及API：

- ``GET /api/v1/namespaces/{namespace}/resourcequota``
- ``GET /api/v1/namespaces``

8.2 Prometheus数据自动采集

目标：通过Prometheus接口获取监控数据，自动汇总命名空间资源使用量

实现方案：

1. 配置Prometheus连接信息
2. 定义查询语句获取CPU/内存使用量

- CPU： ``sum(rate(container_cpu_usage_seconds_total{namespace!=""}[1d])) by (namespace)``
- 内存： ``sum(container_memory_working_set_bytes) by (namespace)``
 3. 定时任务自动拉取数据
 4. 生成日报文件或直接存储到数据库

涉及API：

- Prometheus Query API
- Prometheus Query Range API

8.3 容器化部署

目标：将工具制作成Docker镜像，方便部署和分发

Dockerfile示例：

```
FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt
```



```
COPY . .
```

```
EXPOSE 8000
```

```
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

构建与运行：

```
docker build -t k8s-resource-analyzer .
```

```
docker run -d -p 8000:8000 -v ./data:/app/data k8s-resource-analyzer
```

8.4 其他优化方向

1. **数据持久化**：从内存存储改为SQLite/PostgreSQL
2. **用户认证**：增加登录功能，支持多用户
3. **告警功能**：使用率超过阈值时发送通知
4. **历史趋势**：展示历史使用率变化趋势图
5. **多集群支持**：支持同时管理多个K8s集群

9. 附录

9.1 依赖清单

```
fastapi==0.109.0
uvicorn==0.27.0
openpyxl==3.1.2
python-multipart==0.0.6
pydantic==2.5.3
jinja2==3.1.3
```

9.2 配置文件示例

无配置文件，所有参数通过代码或Web界面管理。

9.3 常见问题

Q1: 导出时提示"Report not found"

A: 确保已上传日报文件，且该日期有匹配配额的项目。

Q2: 上传文件解析失败

A: 检查文件格式是否符合要求，每行格式为：`项目名称;命名空间;CPU;内存(bytes)`

Q3: 使用率显示为空

A: 确保配额中的项目名称与日报中的项目名称完全匹配。

文档结束