

# 基于Docker的异常值检测服务

## 1. 引言

Docker环境中创建一个Python环境，使用Sanic框架对外提供HTTP接口服务。

## 2. 目标

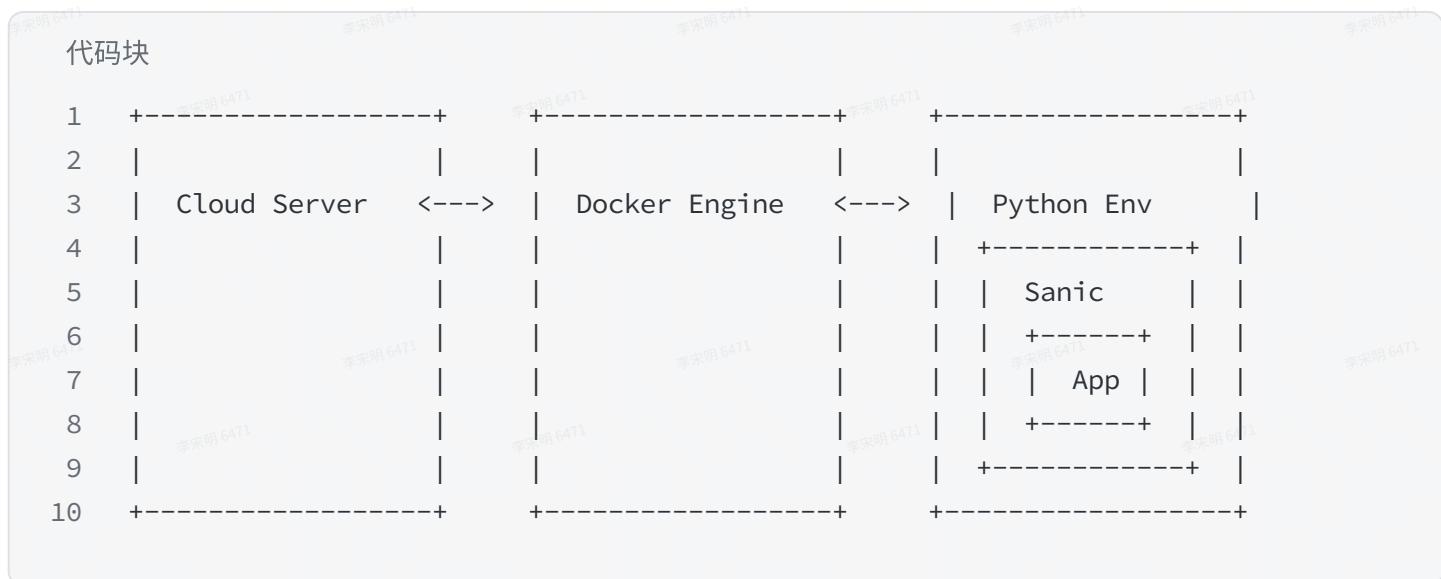
- 在Docker容器中搭建Python运行环境。
- 通过Sanic框架提供稳定的HTTP接口服务。
- 在云服务器上部署和运行Demo工程。

## 3. 系统架构

### 3.1 组件

- Docker Engine
- Python 3.x
- Sanic Web Framework

### 3.2 架构图



## 4. Docker环境搭建

## 4.1 Dockerfile

创建一个Dockerfile来定义容器环境。

代码块

```
1 使用官方Python运行时作为父镜像
2 FROM python:3.8-slim
3 设置工作目录
4 WORKDIR /usr/src/app
5 安装Sanic和其他依赖
6 COPY requirements.txt .
7 RUN pip install --no-cache-dir -r requirements.txt
8 复制项目文件到工作目录
9 COPY ..
10 暴露端口
11 EXPOSE 8000
12 运行Sanic服务器
13 CMD ["python", "app.py"]
```

## 4.2 requirements.txt

定义Python环境和依赖。

代码块

```
1 sanic==21.3.4
2 #其他依赖...
```

## 5. Sanic Web服务

创建一个Sanic应用来提供HTTP接口。

代码块

```
1 #app.py
2 from sanic import Sanic, response
3 from algorithm import async_my_algorithm
4 app = Sanic("ChageeService")
5
6 @app.route("/algorithm", methods=["POST"])
7 async def algorithm(request):
8     data = request.json
9     result = await async_my_algorithm(data)
10    return response.json(result)
11
```

```
李宋明 6471  
12 if name == "__main__":  
13     app.run(host="0.0.0.0", port=8000)
```

## 6. 部署到云服务器

### 6.1 构建和推送Docker镜像

在本地构建Docker镜像并推送到Docker Hub。

代码块

```
1 docker build -t myalgorithm-service .  
2 docker push myalgorithm-service
```

### 6.2 在云服务器上运行容器

登录到宋明提供的云服务器，并运行以下命令来启动容器。

代码块

```
1 docker pull myalgorithm-service  
2 docker run -d --name myalgorithm-service -p 8000:8000 myalgorithm-service
```

## 7. 安全和日志

- 确保Docker容器以非root用户运行。
- 配置日志记录，以便于监控和调试。

## 8. 测试

在本地和云服务器上对服务进行彻底测试，确保所有算法和接口按预期工作。

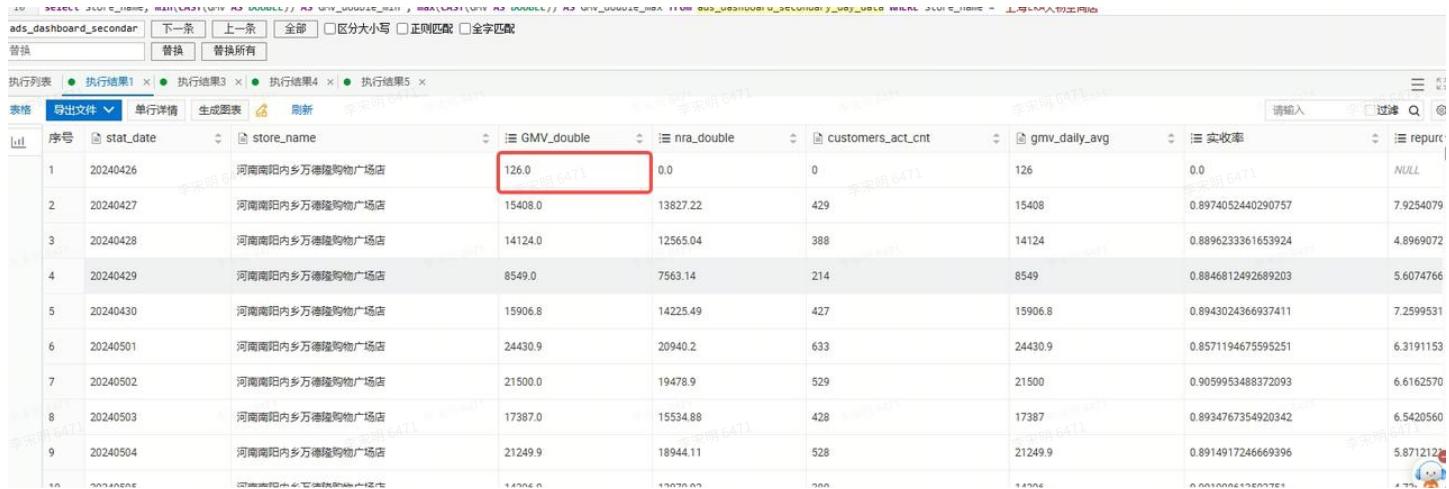
## 9. 维护和升级

- 定期更新Docker镜像和依赖。
- 按需添加新算法或更新现有算法。

## 10. 结论

本文档提供了一个在Docker中创建Python环境，实现算法并通过Sanic提供HTTP服务的指南。按照上述步骤，可以在云服务器上成功部署和运行。

## 11. 输入参数与输出参数



The screenshot shows a database query results page with a table titled 'ads\_dashboard\_secondary'. The table contains data for various stores, including their names, GMV, and other metrics. A red box highlights the 'GMV\_double' column for the first row.

序号	stat_date	store_name	GMV_double	nra_double	customers_act_cnt	gmv_daily_avg	实收率	repu
1	20240426	河南南阳内乡万德隆购物广场店	126.0	0.0	0	126	0.0	NULL
2	20240427	河南南阳内乡万德隆购物广场店	15408.0	13827.22	429	15408	0.8974052440290757	7.9254079
3	20240428	河南南阳内乡万德隆购物广场店	14124.0	12565.04	388	14124	0.8896233361653924	4.8969072
4	20240429	河南南阳内乡万德隆购物广场店	8549.0	7563.14	214	8549	0.8846812492689203	5.6074766
5	20240430	河南南阳内乡万德隆购物广场店	15906.8	14225.49	427	15906.8	0.8943024366937411	7.2599531
6	20240501	河南南阳内乡万德隆购物广场店	24430.9	20940.2	633	24430.9	0.8571194675595251	6.3191153
7	20240502	河南南阳内乡万德隆购物广场店	21500.0	19478.9	529	21500	0.9059953488372093	6.6162570
8	20240503	河南南阳内乡万德隆购物广场店	17387.0	15534.88	428	17387	0.8934767354920342	6.5420560
9	20240504	河南南阳内乡万德隆购物广场店	21249.9	18944.11	528	21249.9	0.8914917246669396	5.8712124

### 输入参数：

{

"指标板块": "当日实时脉搏",

"指标": {

    "GMV": 126.0

    "实收": 0.0

    "新增会员数": 4.0,

    "出杯数": 10.0,

    "门店名称":

}

}

### 输出参数：

{

"指标板块": "当日实时脉搏",

"指标": {

    "GMV异常率": 1.0,

    "实收异常率": 1.0,

    "新增会员数异常率": 1.0,

    "出杯数异常率": 1.0,

李宋明 6471

}

李宋明 6471

李宋明 6471

## ■ 经营驾驶舱资产清单

李宋明 6471

## ■ 驾驶舱输出表样

## 13.代码仓库

李宋明 6471

[https://gitlab.bwcjxt.com/data\\_algorithm/abnormaldetection](https://gitlab.bwcjxt.com/data_algorithm/abnormaldetection)

李宋明 6471

-1

-2

-3

-4

李宋明 6471

李宋明 6471