



链滴

Aries - 基于 Go + Vue 的现代化博客系统

作者: [zyk](#)

原文链接: <https://ld246.com/article/1614086809236>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

项目介绍

主题管理功能还在开发中!

Aries 是基于 Gin + GORM + MySQL + Vue + H5 开发的现代化博客系统，博客主要分为**后台管理端**和**示端**。

Github 仓库地址: <https://github.com/zhaoyangkun/aries>

- **后台管理端**采用**前后端分离**的开发模式，通过 **JSON** 格式数据进行前后端交互。
- **展示端**基于传统模式开发，采用 **Go Template** 模板引擎来渲染页面，有利于 **SEO** 优化。
- 数据库层没有设置物理外键，所有外键操作都在业务层处理。
- 评论功能通过 **评论组件**方式实现。

预览

预览地址: <https://cangmangai.cn>

功能特性

1. 文章

- 编辑器采用 **vditor**，提升中文 **markdown** 使用体验
- 文章标签、分类管理
- 文章置顶
- 文章加密
- 自定义文章链接
- 文章排序
- 草稿箱和回收站
- 从文件导入文章

2. 外观

- 自定义菜单
- 主题导入
- 主题切换
- 主题设置
- 主题编辑

3. 评论

- 回收站
- 评论审核
- 邮件通知

4. 图床

- 支持 sm.ms, imgbb 和 腾讯云 cos 存储

- 多图片上传
- 图片管理

5. 页面

- 日志管理
- 自定义页面
- 图库管理

6. 友情链接

- 添加、修改友链
- 友链分类

7. 用户

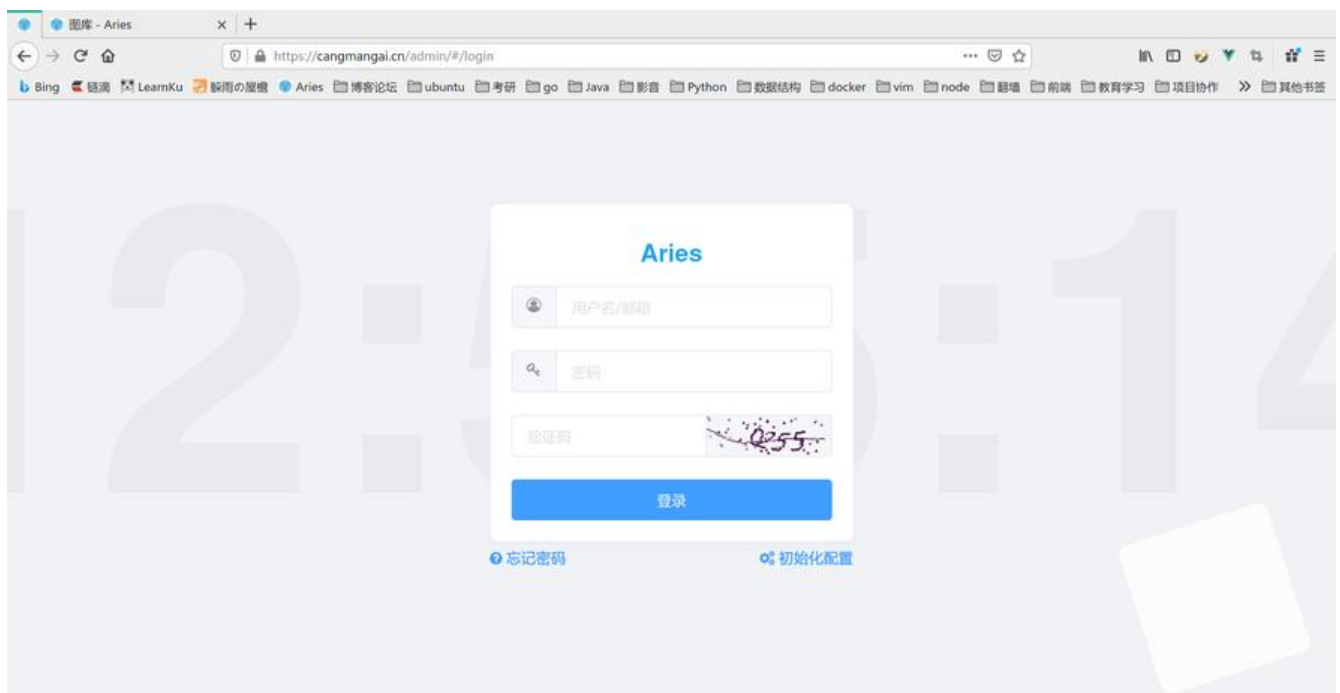
- 修改用户信息
- 修改密码

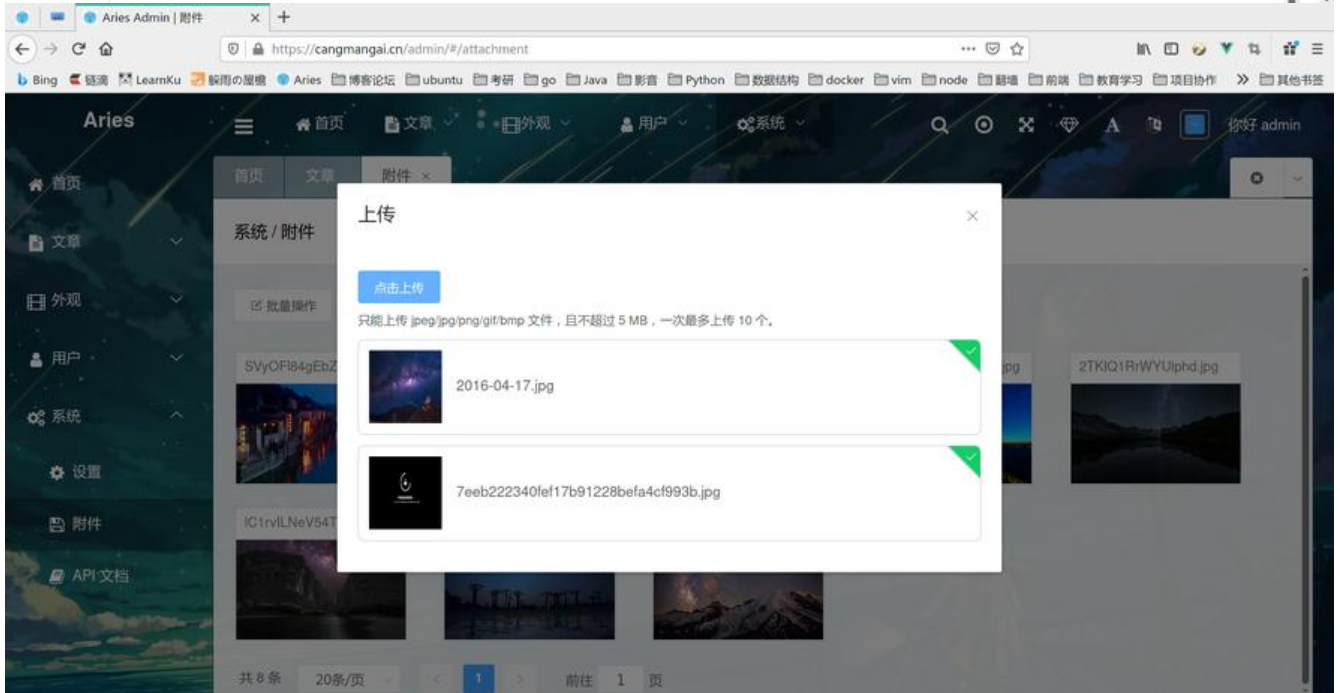
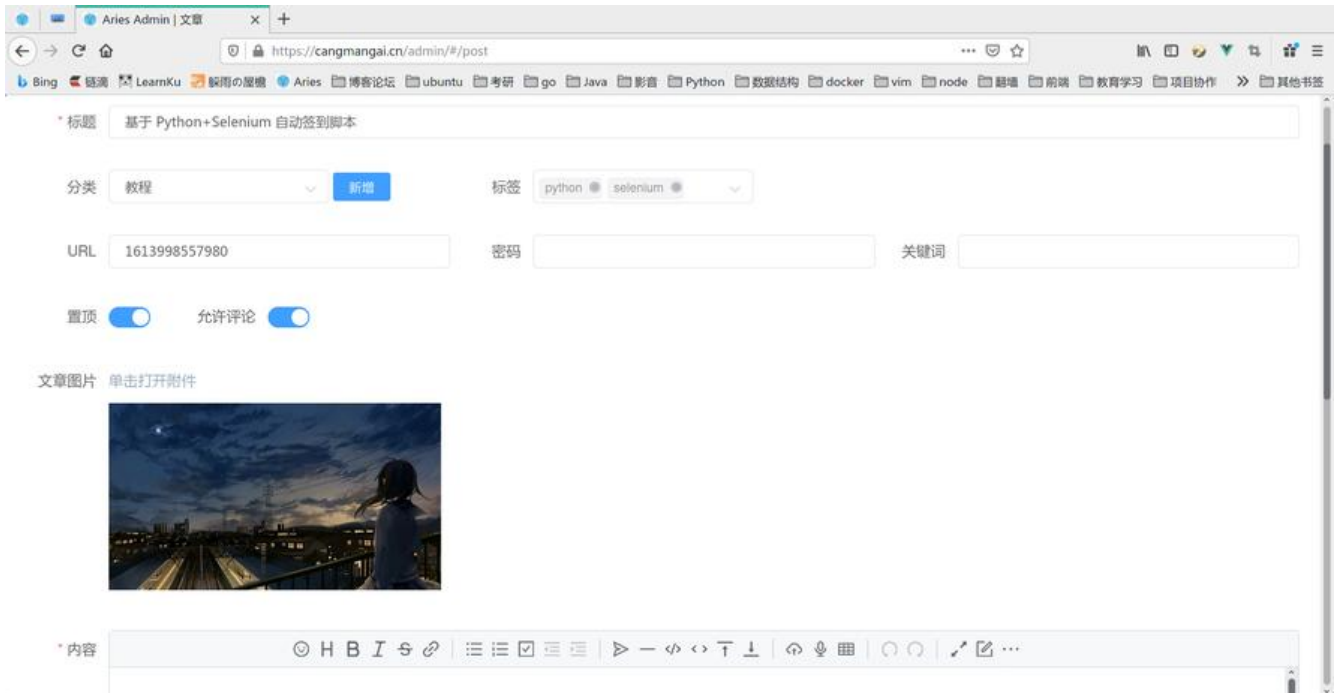
8. 设置

- 网站设置
- 邮件设置
- 图床设置
- 评论设置
- 参数设置

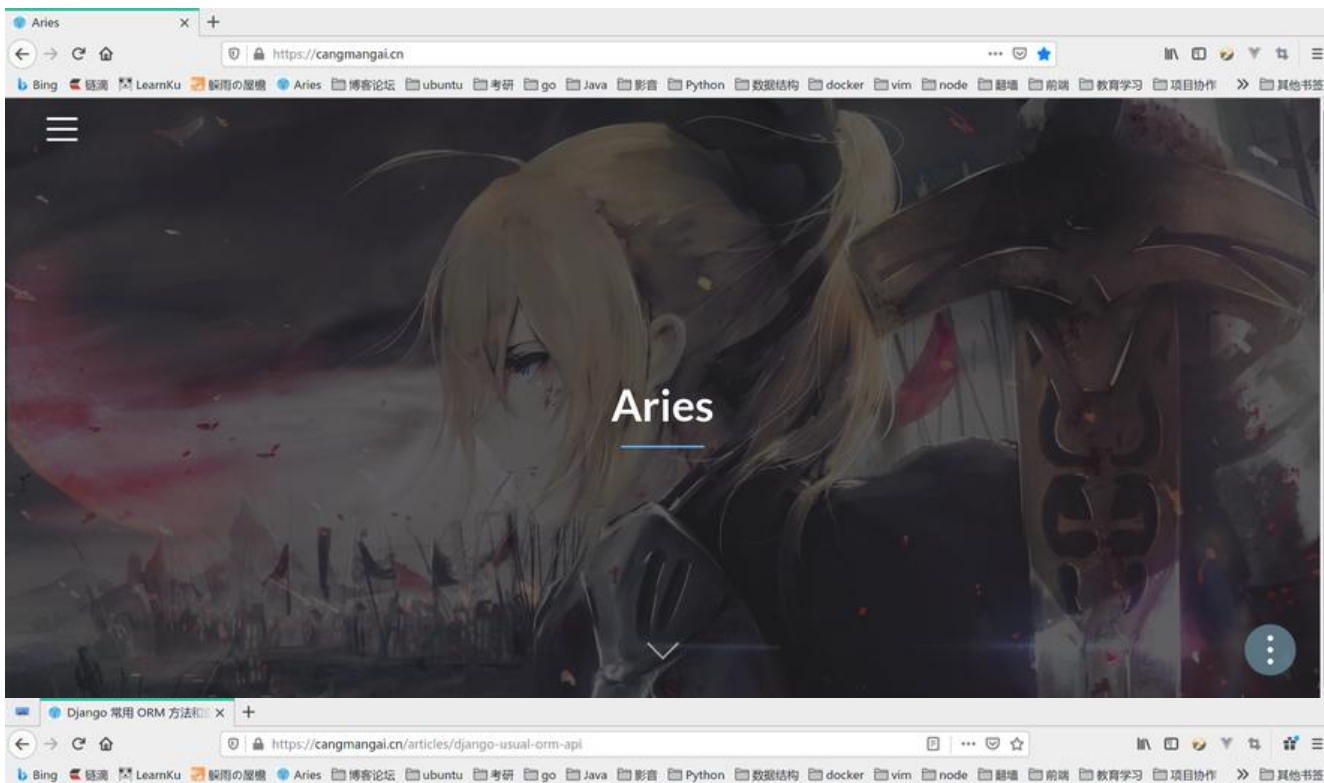
部分截图

- 后台管理端





● 博客展示端



首页 分类 标签 归档 友链 日志 图库

例如，重复利用索引来调用查询集中的对象，会导致每次都查询数据库：

```
>>> queryset = Entry.objects.all()
>>> print(queryset[5]) # 查询数据库
>>> print(queryset[5]) # 再次查询数据库
```

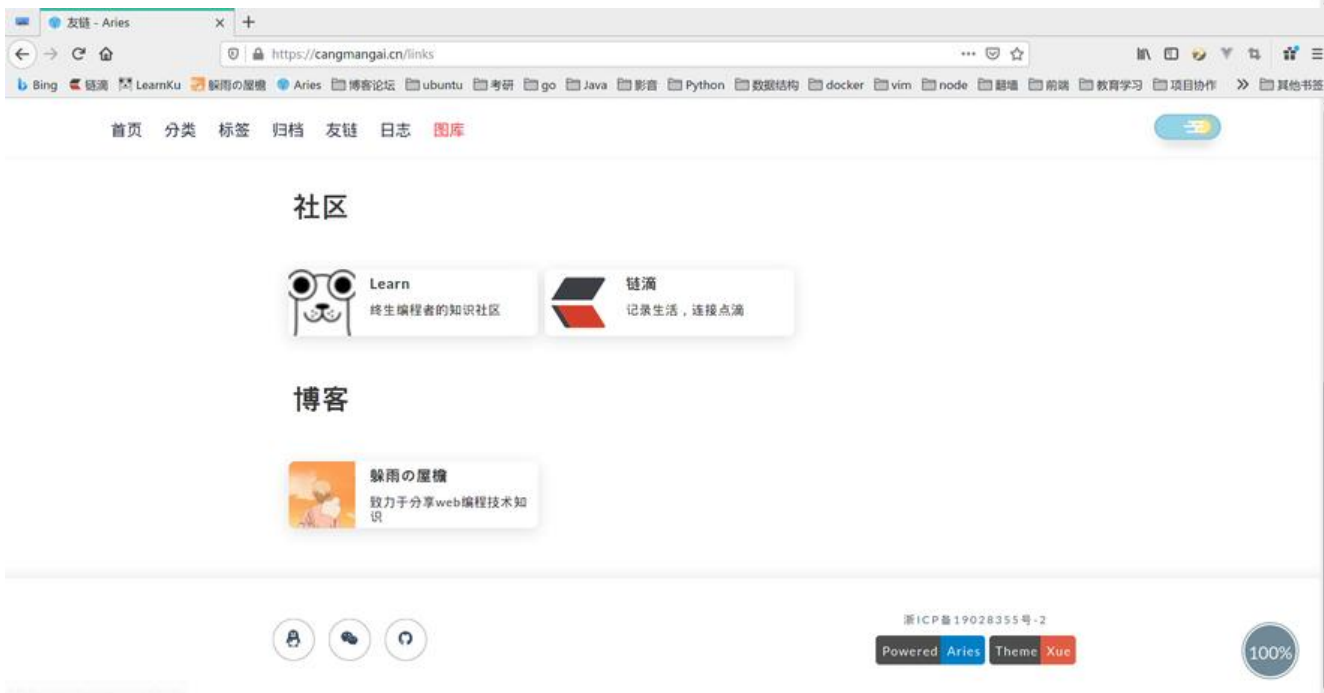
不过，如果已对查询结果集进行检出操作（例如循环遍历操作），就会直接调用缓存中的数据：

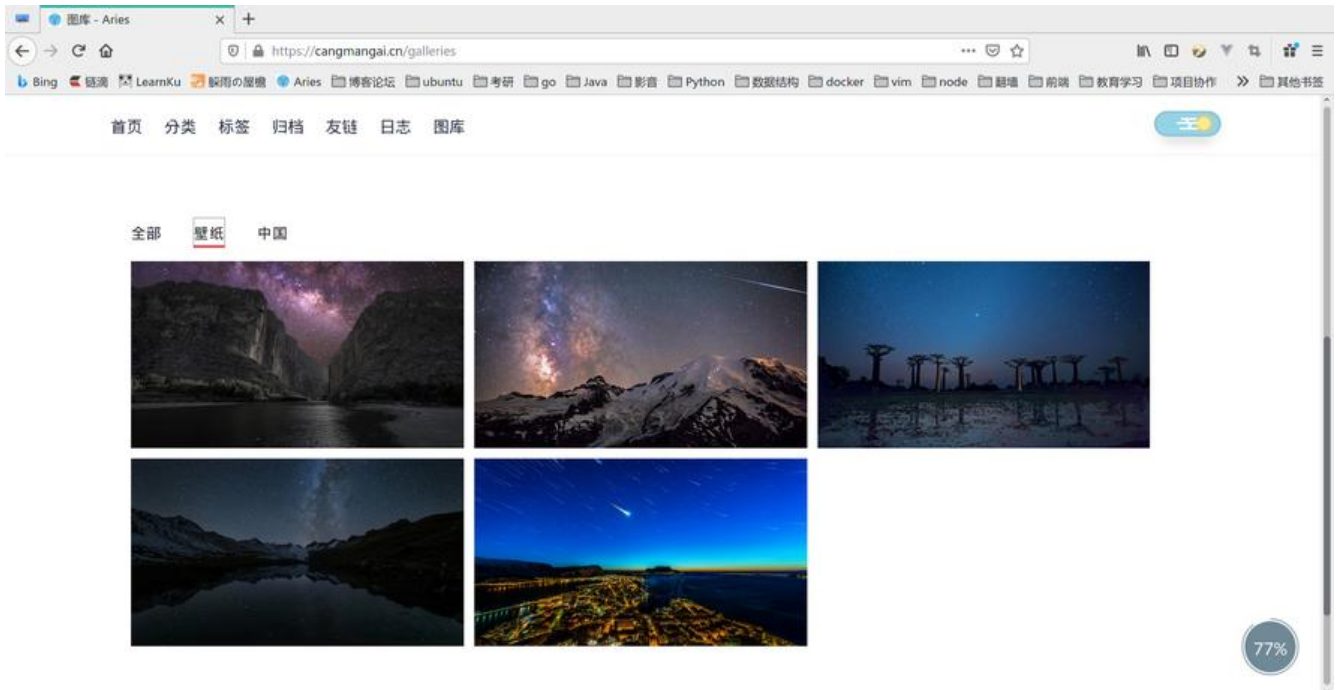
```
1 >>> queryset = Entry.objects.all()
2 >>> [entry for entry in queryset] # 查询数据库，写入缓存
3 >>> print(queryset[5]) # 调用缓存
4 >>> print(queryset[5]) # 调用缓存
```

以下动作会触发全部查询结果集，并写入缓存。

```
1 >>> [entry for entry in queryset]
2 >>> bool(queryset)
3 >>> entry in queryset
4 >>> list(queryset)
```

93%





如何在本地运行 Aries

- 运行环境:

请确保已安装好以下环境，配置好 **go mod** 代理和 **npm** 国内镜像源

- Go 1.15
- Node.js v12+
- MySQL 5.7+

如何配置 **go mod** 代理: <https://goproxy.io/zh/>

如何配置 **npm** 国内镜像源: <https://www.cnblogs.com/luckyhui28/p/12268313.html>

- 克隆项目代码到本地:

`git clone https://github.com/zhaoyangkun/aries.git`

- 新建名称为 **aries** 的数据库，注意字符集为 **utf8m64**，字符编码为 **utf8m64_general_ci**。
- 修改项目目录下 `config/develop.yaml` 配置文件中的数据库连接密码:

服务器配置

server:

```
mode: "debug"           # 运行模式
port: "8088"            # 端口
token_expire_time: 3600 # JWT token 过期时间 (单位: 秒)
allowed_refers: ["localhost", "127.0.0.1"] # 允许的 referer
```

数据库配置

db:

```
host: "127.0.0.1"      # 主机地址
user_name: "root"      # 用户名
password: "19960331"   # 密码
database: "aries"      # 数据库名
```

```
port: "3306"      # 端口
time_zone: "Local" # 时区
max_idle_conn: 10 # 最大空闲连接数
max_open_conn: 20 # 最大打开连接数
```

- 进入项目根目录，安装 `gin` 项目相关依赖：

```
go mod download
go get -u github.com/swaggo/swag/cmd/swag
```

- 进入项目根目录，启动 `gin` 项目：

```
go run main.go
```

- 进入项目中的 `d2-admin` 目录，安装 `npm` 依赖：

```
npm install
```

- 待 `npm` 依赖安装完毕后，启动 `Vue` 项目：

```
npm run serve
```

- 待 `gin` 和 `Vue` 项目启动完毕后，在浏览器中访问 <http://localhost:8080> 即可进入后台管理。
- 在浏览器中访问 <http://127.0.0.1:8088> 可进入博客展示端。

部署

仅支持 `docker` 部署，需要具备一定的 `linux` 基础。

确保服务器已配置好 `docker` 运行环境，并安装好 `mysql`，`mysql` 版本不低于 5.7。

- 在 `mysql` 中新建名称为 `aries` 的数据库，注意字符集为 `utf8m64`，字符编码为 `utf8m64_general_ci`。
- 在主目录下创建 `.aries` 文件夹：

```
mkdir ~/.aries
```

- 创建配置文件 `aries.yaml`：

```
touch ~/.aries/aries.yaml
```

`aries.yaml` 配置条目具体可以参考：<https://github.com/zhaoyangkun/aries/blob/master/config/product.yaml>

可先将 `product.yaml` 所有内容复制到 `aries.yaml` 中再做修改。

注意：

- `mode` 要设置为 `release`，表示生产环境；
 - 注意校对数据库的 **主机地址**，**帐号**和**密码**；
 - 使用 **重置密码**功能的话，需配置 `smtp`。
- 拉取 `aries` 镜像：

```
docker pull zhaoyangkun/aries
```

- 运行容器：


```
docker run -p 8088:8088 --name aries --restart=always --network=host -v ~/.aries:/root/.aries
-d zhaoyangkun/aries
```

- 反向代理:

推荐使用 `nginx` , 由于 `aries` 默认运行在 `8088` 端口上, 需要在云服务器**安全组**开放 `8088` 端口, 同时 `ginx` 反代 `8088` 端口到 `80` 端口, 这里给出一段 `nginx`配置文件的示例:

```
upstream aries_server {
    server cangmangai.cn:8088; # cangmangai.cn 为域名, 也可以改为公网 IP, 8088 表示监听端口
}

# http 重定向到 https
server {
    listen 80; # 80 为 http 默认端口
    server_name cangmangai.cn;
    rewrite ^(.*) https://$host$1 permanent;
}

# https 配置
server {
    charset utf-8;
    listen 443 ssl; # 443 为 https 默认端口
    server_name cangmangai.cn;
    ssl_certificate /ssl/cn.pem; # .pem证书路径
    ssl_certificate_key /ssl/cn.key; # .key证书路径
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;
    location / {
        proxy_pass http://aries_server$request_uri; # 反向代理, 将 8088 端口转发到 443 端口
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        client_max_body_size 10m;
    }
    location /bdunion.txt {
        alias /ssl/bdunion.txt;
    }
}
```