

?????

- `git commit -m "Add SSL support"`
- #1. `git push origin master`
- #2. `www.example.com 503 Service Unavailable`
- #3. DB connection & query failed
- #4. `git pull origin master`
- #5. Git READ.md file push
- #6. The current branch master has no upstream branch.

? ???? ?? & ?? ?? ???

?? ???? ??

- Docker 安装 BookStack + MariaDB

```
bookstack/
├── bookstack/          # BookStack 应用目录
├── ...                  # 其他文件
├── bookstack_db/       # MariaDB 数据库
├── ...                  # (.frm, .ibd 等 DB 文件)
├── certs/              # SSL 证书 (可选)
├── html/               # nginx 静态文件
├── vhost.d/            # 虚拟主机配置 (可选)
└── docker-compose.yml  # Docker 配置
```

bookstack container

```
/app/www/
├── .env                # Laravel 环境变量
├── artisan             # Laravel CLI 脚本
├── config/             # 配置文件
│   ├── app.php
│   └── database.php
├── routes/            # 路由文件
│   └── web.php
├── resources/         # 资源文件, 视图等
│   ├── views/
│   └── lang/
├── storage/           # 存储, 缓存, 日志等
│   ├── logs/
│   ├── app/
│   └── framework/
└── public/            # 静态文件
    └── uploads/       # 用户上传文件
```

bookstack_db container

```

./
├── var/
│   ├── lib/
│   │   ├── mysql/
│   │   │   # ↳  MariaDB  DB  files
│   │   │   ├── bookstack/
│   │   │   │   # BookStack  DB  files
│   │   │   │   ├── users.ibd
│   │   │   │   ├── pages.ibd
│   │   │   │   └── ...
│   │   │   ├── mysql/
│   │   │   │   #  DB  (root,  user)
│   │   │   │   ├── performance_schema/
│   │   │   │   └── ibdata1, ib_logfile0
│   └── etc/
│       ├── my.cnf
│       │   # MariaDB  config
│       └── Linux  files

```

1. SSH? ??? ?? ??

Linux, macOS, WSL,  , VSCode 

```
ssh -i ~/xxx.pem ubuntu@your-lightsail-ip
```

 (Windows PowerShell / CMD)

```
ssh -i "C:\Users\aweso\pem\####.pem" ubuntu@your-lightsail-ip
```

2. home ??

- `.yaml`` `□□` `□□`
- `docker` `□□□□` `□□□□` `□□` (`docker ps`, `docker-compose up -d`. `docker-compose down` `□`)

```
cd DainWiki
```

3. ?? ???? ??? ???? ?

```
docker exec -it bookstack bash
```

4. Laravel ? ??? ??

- ``.env`` 文件 文件

```
docker exec -it bookstack bash
cd /app/www
ls -al .env
```

5. Laravel ?? ?? ?? ? ??????? ?? ??

- DB 文件 文件
- DB 文件 (文件) 文件

```
php artisan config:clear
php artisan cache:clear
php artisan view:clear
php artisan migrate --force
```

6. maria db ??? ?? ?

```
docker exec -it bookstack_db bash 文件 文件
mysql -u root -p
文件 rootpassword

文件文件
Welcome to the MariaDB monitor...
MariaDB [(none)]>
```

7. ?? ??? ?? ??

```
df -h
```


?? ? ?

- Docker **AWS Lightsail**
- HTTPS ** ' ' **

- 浏览器使用 **SSL** 加密通信，防止数据被窃听。
- HTTPS 使用 **SSL/TLS** 协议，由 **CA** 颁发证书，验证服务器身份。
- 浏览器会检查证书是否有效，如果无效，会提示警告。

- Docker [] [] **SSL** [] [] [] [] [] [] [] []
- [] [] Docker [] [] docker-compose.yml [] [] [] []

☐ nginx-proxy ☒ , Let's Encrypt ☐ SSL ☐☐☐☐

```

└─ DainWiki/
   └─ bookstack/
      └─ bookstack_db/
         └─ docker-compose.yml

```

```

└─ DainWiki/
├─ bookstack/          # BookStack のインストール
├─ bookstack_db/       # MariaDB のインストール
└─ certs/              # SSL 証明書 (Docker コンテナ)

```

```
└─ /html/ # nginx 静态 资源
└─ /vhost.d/ # 静态 资源 (静态 资源)
└─ docker-compose.yml # 静态 资源
```

```
ubuntu@ip-172-26-5-34:~/DainWiki$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  NAMES                CREATED        STATUS
d25363af4d9b   lscr.io/linuxserver/bookstack:late "/init"                  bookstack            6 minutes ago  Up 6 minutes
e3662714bf27   jrcs/letsencrypt-nginx-proxy-companion "/bin/bash /app/entr... letsencrypt          6 minutes ago  Up 6 minutes
fa3cf91211e6   jwilder/nginx-proxy                "/app/docker-entrypo... nginx-proxy           6 minutes ago  Up 6 minutes
:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
613018760a99   mariadb:10.4                       "docker-entrypoint.s... bookstack_db         6 minutes ago  Up 6 minutes
```

① Let's Encrypt 静态 资源

```
service:
- VIRTUAL_HOST=dainwiki.com
- LETSENCRYPT_HOST=dainwiki.com
- LETSENCRYPT_EMAIL=your@email.com
```

② APP_URL 从 http 到 https 静态 资源

```
- APP_URL=http://dainwiki.com
+ APP_URL=https://dainwiki.com
```

→ 静态 资源 https 静态 资源 .

③ nginx-proxy 静态 资源

```
nginx-proxy:
  image: jwilder/nginx-proxy
  container_name: nginx-proxy
  restart: always
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - /var/run/docker.sock:/tmp/docker.sock:ro
    - ./certs:/etc/nginx/certs
    - ./vhost.d:/etc/nginx/vhost.d
    - ./html:/usr/share/nginx/html
  networks:
    - bookstack_net
```

```
letsencrypt:
  image: jrcs/letsencrypt-nginx-proxy-companion
  container_name: letsencrypt
  restart: always
  environment:
    - NGINX_PROXY_CONTAINER=nginx-proxy
    - DEFAULT_EMAIL=your@email.com
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock:ro
    - ./certs:/etc/nginx/certs
    - ./vhost.d:/etc/nginx/vhost.d
    - ./html:/usr/share/nginx/html
  depends_on:
    - nginx-proxy
  networks:
    - bookstack_net
```

→ ☐ ☐ + SSL ☐ ☐

#2. www ??? 503 ????? ??

??

- `www` : <https://dainwiki.com>, `http://dainwiki.com` `www` `www`
- `http://www.dainwiki.com`, `https://www.dainwiki.com` `www` `www` `www`
- DNS `www` `www` (`A` `www` `www` `@` `www`(`www`) `www` `www` `www`)

503 Service Temporarily Unavailable

nginx

타입 ▼ ⓘ	호스트	값/위치	TTL	우선 순위	서비스 ▼	상태
A	www	43.202.6.70	1800		DNS 설정	<button>수정</button> <button>삭제</button>
A	@	43.202.6.70	1800		DNS 설정	<button>수정</button> <button>삭제</button>

?? ??

- Docker `nginx-proxy` + `letsencrypt-nginx-proxy-companion` `nginx-proxy` `nginx-proxy` `nginx-proxy`
- `.env` `docker-compose.yml` `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com`
- `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com`

????

- Docker `nginx-proxy` + `letsencrypt-nginx-proxy-companion` `nginx-proxy` `nginx-proxy` `nginx-proxy`
- `.env` `docker-compose.yml` `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com` `www.dainwiki.com`

```
VIRTUAL_HOST=dainwiki.com,www.dainwiki.com
LETSencrypt_HOST=dainwiki.com,www.dainwiki.com
LETSencrypt_EMAIL=you@example.com
```

`docker-compose down`

```
docker-compose down
docker-compose up -d
```

□ □□ □□

- www.dainwiki.com □ □□ SSL □□□ □□ □□
- [www □□ □ https:// □ □ □□□](https://www.□□.https://□□□□)

#3. DB ??? & ??? ?? ?????

??

- 1000 BookStack API 100 100 1000 ((Apache + PHP + MySQL) 100 100 DB 1000)
- 503 100 1000 1000 100 100 1000 100 100 100 100 1000

?? ??

- MySQL 1000 100 1000 htdocs 100 100 1000
- 100 100 100 100 100 1000 100 100
- 100 100 100 (volume, data 1000) 100

?? ??

- 100 : AWS Lightsail 100 100 , Docker + Volume 1000 100 100 100
- 100 : docker-compose up -d 100 100 100 , 1000 1000 volume: 100 100
- 100 100 :
100 100 60GB 100 100 1000 1000 100 100 100 100
100 mysqldump 100 100 100 1000 100 100
- 100 : AWS Lightsail \$10 100 (2GB RAM, vCPU 1, SSD 60GB)

? ??

```
└─ DainWiki/
├─ bookstack/      # BookStack 100 100, 1000
├─ bookstack_db/   # MariaDB 1000 (100 1000, 100 100)
├─ docker-compose.yml
└─ .env
```

? yml. ?? ??? ? ? ?

```
bookstack:
  volumes:
    - ./bookstack_data:/config

bookstack_db:
  volumes:
    - ./db_data:/var/lib/mysql
```

容器 名称 (bind mount) 容器 名称 .

- 容器 名称 (./bookstack_data) 容器 名称 (/config) 容器 名称
- 容器 名称 (=Lightsail 容器 名称) 容器 名称 容器 名称 .
- 容器 名称 (/) 容器 名称 ~/DainWiki/bookstack_data, db_data 容器 名称 (容器 名称 : /home/ubuntu/DainWiki/bookstack_data)

? ? ? ? ? ? ? ?

```
docker inspect bookstack
```

- 容器 名称 Mounts 容器 名称 容器 名称

? ? ? ? ? ? ? ?

```
df -h
```

```
ubuntu@ip-172-26-5-34:~/DainWiki$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        58G   3.9G   54G   7% /
tmpfs            956M   0    956M   0% /dev/shm
tmpfs            383M   1.3M  381M   1% /run
tmpfs            5.0M   0    5.0M   0% /run/lock
efivarfs         128K   3.6K  120K   3% /sys/firmware/efi/efivars
/dev/nvme0n1p16  881M  136M  684M  17% /boot
/dev/nvme0n1p15  105M   6.1M   99M   6% /boot/efi
tmpfs            192M   12K   192M   1% /run/user/1000
```

#4. ??? ?? ??? ????? ?? ??

??

- `https://www.dainwiki.com` 的 的 的 的
- `www` 的 的 `yaml` 的 的 的 → `www` 的 的 的 的 的
- `docker logs letsencrypt` 的 的 的 的 的 的 , 的 的 的 的

?? ??

- `.env` 的 `LETSencrypt_HOST` 的 `VIRTUAL_HOST` 的 的 的 (`www.dainwiki.com` 的)
- 的 的 的 的 ,
- Let's Encrypt Companion 的 的 的 / 的 的 的
- 的 的 的 的 的 的 的 的 的 的
- 的 的 的 的 5 的 的 的 的

?? ??

SSL 的 的 的 的

```
docker logs letsencrypt
```

```
Reloading nginx proxy (nginx-proxy)...
2025/05/18 04:30:43 Generated '/etc/nginx/conf.d/default.conf' from 4 containers
2025/05/18 04:30:43 [warn] 63#63: "ssl_stapling" ignored, no OCSP responder URL in the certificate "/etc/nginx/certs/www.dainwiki.com.crt"
2025/05/18 04:30:43 [notice] 63#63: signal process started
Reloading nginx proxy (nginx-proxy)...
2025/05/18 04:30:43 Contents of '/etc/nginx/conf.d/default.conf' did not change. Skipping notification ''
2025/05/18 04:30:43 [warn] 77#77: "ssl_stapling" ignored, no OCSP responder URL in the certificate "/etc/nginx/certs/www.dainwiki.com.crt"
2025/05/18 04:30:43 [notice] 77#77: signal process started
```

的 的 的

```
docker exec -it nginx-proxy sh
```

```
docker exec -it nginx-proxy bash
```

的 的 的 的 的 的 的

```
ls /etc/nginx/certs/
```

- `dainwiki.com/`, `www.dainwiki.com/` 的 的 的
- 的 的 `fullchain.pem`, `cert.pem`, `key.pem` 的 的 的

```
ubuntu@ip-172-26-5-34:~/DainWiki$ docker exec -it nginx-proxy bash
root@c44cb4cfd4ae:/app# ls /etc/nginx/certs/
dainwiki.com          dainwiki.com.dhparam.pem  www.dainwiki.com.chain.pem  www.dainwiki.com.key
dainwiki.com.chain.pem dainwiki.com.key          www.dainwiki.com.crt
dainwiki.com.crt      dhparam.pem               www.dainwiki.com.dhparam.pem
```

#5. Git? README.md ??? ??? ????

?? push ?? ??

??

- GitHub에 README.md를 올렸을 때, Git에서는 `git push`를 해야 합니다.
- GitHub에서 `pull`할 때, `--allow-unrelated-histories` 옵션을 추가하여 `push`할 수 있습니다.
- `git pull --rebase`를 사용하여 `rebase`를 할 수 있습니다.

??

명령어	설명	결과
<code>git pull origin main --allow-unrelated-histories</code>	원본 브랜치에서 1회 가져옴	원본 브랜치에서 가져온 코드가 Git에 업로드됨
<code>git pull origin main --rebase</code>	원본 브랜치에서 (재)가져옴	원본 브랜치에서 가져온 코드가 <code>rebase</code> 되어 <code>push</code> 됨

1회 `pull`할 때 `--allow-unrelated-histories` 옵션을 추가합니다.

원본 브랜치에서 '원본 브랜치'를 가져옴. Git에서는 `--rebase` 옵션을 추가합니다.

`--allow-unrelated-histories` 옵션을 추가하여 원본 브랜치에서 가져온 코드가 `push`됨.

원본 브랜치에서 `X → A → B → C`를 가져옴. 원본 브랜치에서 가져온 코드가 `push`됨.

?? ??

```
# 1. 원본 브랜치 추가
git remote add origin https://github.com/owner/repo.git
```

```
# 2. 원본 브랜치에서 코드를 가져옴
git pull origin main --allow-unrelated-histories
```

```
# 3. 分支合并到主分支
```

```
# 4. 推送代码到远程仓库
```

```
git push origin main
```

或者使用 `git pull` 命令来拉取最新代码并合并到本地分支。

- `git pull --rebase origin main` → 将本地分支与远程分支合并，并基于最新的主分支进行重新合并。
- `git pull origin main` → 将本地分支与远程分支合并，并基于最新的主分支进行重新合并。

#6. The current branch master has no upstream branch. ??

??

- Git Bash 打开 终端 窗口 运行 命令

The current branch master has no upstream branch.

\$ git push fatal: The current branch master has no upstream branch. To push the current branch and set the remote as upstream, use git push --set-upstream origin master To have this happen automatically for branches without a tracking upstream, see 'push.autoSetupRemote' in 'git help config'.

??

- 在 本地 创建 分支 (master) 并 切换 到 该 分支
- master 分支 设置 上游 分支 --set-upstream 指定 上游 分支
- main 分支 设置 上游 分支 并 推送 到 远程

?? ??

- master 分支 设置 上游 分支 并 推送 到 远程
- 在 本地 创建 分支 + GitHub(远程 origin) 分支 并 推送 到 远程

```
git branch -m master main
git push -u origin main
# (GitHub 默认分支 是 main)
git push origin --delete master # (删除)
```

1. 在 本地 创建 分支 并 推送 到 远程

```
git branch -m master main
```

- -m = move (rename)

2. `origin` 分支 名称

```
git push -u origin main
```

- -u `main` 分支 名称 `upstream` 分支 名称

3. GitHub 分支 名称 `main` 分支 名称

1. GitHub 分支 名称 → 分支 名称 名称
2. Settings → Branches 分支 名称
3. Default branch `master` → `main` 分支 名称

4. `master` 分支 名称

```
git push origin --delete master
```

分支 名称

`master` 分支 名称 `--set-upstream` 分支 名称 分支 名称 .

```
git push --set-upstream origin master
```

分支 名称 `origin/master` 分支 名称 `master` 分支 名称 `(origin)` 分支 名称 .

分支 名称 `master` 分支 名称 , 分支 名称 分支 名称 分支 名称 ,

分支 名称 分支 名称 `git push, git pull` 分支 名称 `origin/master` 分支 名称 .