

使用docker自定义部署Supernote Private Cloud

使用此方法部署可实现一个高度自定义的Supernote Private Cloud

Supernote Private Cloud 端口使用说明

- **容器端口 8080:** 容器内的 8080 端口由 Nginx 监听，Nginx 作为反向代理，将请求转发到 supernote-service 容器，以提供同步和 Web 管理界面服务。要使用浏览器或 Supernote 设备访问私有云，必须将此端口映射到 Docker 主机，以允许从 Docker 主机外部访问。有关从 Docker 容器内部发布应用程序的说明，请参阅“公开端口”部分（参见“主机端口映射 19072”）。
- **容器端口 18072:** 此端口主要用于与 Supernote 设备进行自动同步。
- **容器端口 443:** 容器内的 443 端口由 Nginx 监听，其工作方式与 8080 端口相同。443 端口默认提供 HTTPS 服务。
- **主机映射端口 19072:** 此端口主要用于将 supernote-service 容器的 8080 端口映射到当前部署 Supernote 私有云的 Docker 主机。通过此端口映射，您可以访问 Docker 主机上部署的私有云服务，从而实现基于浏览器的管理和配置以及从 Supernote 设备上实现与私有云的同步。此映射会在 Docker 主机上打开一个端口（例如 19072），并创建一条规则，将所有发往该主机端口的流量直接转发到指定容器的内部端口（例如 8080），有关端口暴露的更多信息，请参阅 Docker 官方文档：<https://docs.docker.com/engine/containers/run/#exposed-ports>。
- **主机映射端口 19443:** 此端口主要用于将 supernote-service 容器的 443 端口映射到当前部署 Supernote 私有云的 Docker 主机。通过此端口映射，您可以经由 HTTPS（Supernote 自签名证书，浏览器不受信任）访问部署在 Docker 主机上的私有云服务，从而实现基于浏览器的管理和配置，以及 Supernote 设备与私有云之间的同步。此映射会在 Docker 主机上打开一个端口（例如 19443），并创建一条规则，将所有发往该主机端口的流量直接转发到指定容器的内部端口（例如 443）。有关端口暴露的更多信息，请参阅 Docker 官方文档：<https://docs.docker.com/engine/containers/run/#exposed-ports>

1. 为什么 supernote-service 容器需要暴露 8080 端口？

1. 容器内的 8080 端口由 Nginx 监听，Nginx 提供反向代理服务，将请求转发到 supernote-service 容器，以提供同步服务和 Web 管理界面。要使浏览器或 Supernote 设备能够访问私有云，必须将此端口映射到 Docker 主机以允许外部访问。
2. Docker 官方文档明确指出：“默认情况下，容器运行时不会向主机暴露任何端口。要从主机访问容器内部监听的端口，必须发布该端口。” 端口发布通常使用 `-p {host_port}:{container_port}` 格式。请参考：<https://docs.docker.com/engine/containers/run/#exposed-ports>

2. 为什么要将容器内的 443 和 8080 端口映射到 Docker 主机上的 19072 和 19443 端口？

1. Supernote 的私有云需要满足无法使用反向代理的用户的需求。因此，我们需要在 Docker 主机上提供一个默认端口作为私有云容器服务的访问点。端口映射（例如 `-p 19072:8080` 和 `-p 19443:443`）是 Docker 推荐的标准做法：它会在主机上打开指定的端口（例如 19072、19443），并将所有发往该端口的流量转发到容器的内部端口（例如 8080、443）。这是从主机外部访问容器服务的标准且必要的方法。参考：<https://docs.docker.com/engine/containers/run/#exposed-ports>

3. 为什么 Supernote 在容器内发布应用程序时，默认使用端口 19072 和 19443 而不是使用 80 和 443 端口？

1. Supernote 私有云目前默认启用 HTTPS 服务(自签名证书, 不被浏览器所信任)。80 和 443 作为服务常用端口, 直接将容器内的 8080 和 443 端口映射到 Docker 主机上的 80 或 443 端口容易导致端口冲突, 造成服务不可用。因此, 我们指定 19072 端口为默认 HTTP 映射端口, 19443 端口为默认 HTTPS 映射端口, 以降低冲突风险, 确保服务稳定性。

2. 要自定义端口, 请按照以下步骤操作:

1. **对于使用 Docker 部署的用户:** 直接修改 Docker 配置项 `-p <local http port>:8080` 和 `-p <local https port>:443`。
2. **对于使用 Docker Compose 部署的用户:** 只需将 `.env` 文件中的 `HTTPS_PORT` 和 `HTTP_PORT` 值修改为外部访问所需的端口即可。
3. **对于使用 `install.sh` 脚本部署的用户:** 目前, 端口 19072 和 19443 被固定为外部连接的默认入口点。如果您需要使用 HTTPS, 请参考反向代理配置说明。

4. 为什么 Supernote 不能提供可信的 HTTPS 服务?

1. Supernote 团队无法为每个用户部署的私有云实例颁发可信的 SSL 证书。证书颁发需要来自可信证书颁发机构的验证以及域名所有权证明, 这对于分布在众多用户自有服务器上的私有云服务而言并不实际。因此, Supernote 为需要 HTTPS 功能的用户提供了 Nginx 和 NAS 反向代理的示例配置, 并在部署手册中提供了 HTTPS 设置说明。
2. 我们已默认将对不受信任证书的支持集成到服务中, 并默认启用 SSL/TLS 配置。如果您需要为私有云服务配置受信任证书, 请参考以下部署方法。我们提供了 `docker run` 和 `docker-compose` 两种部署方法的文档, 其中均包含 HTTPS 配置说明。

5. 我们将在[常见问题解答](#)部分提供有关私有云相关问题的更多更新和解释。

- **主机映射端口 18072:** 此端口主要用于将 `supernote-service` 容器内的自动同步功能映射到当前主机。通过此端口, 您可以使用 Supernote 设备上通过 WebSocket 协议实现的自动同步功能。如果您确定不会使用自动同步功能, 则可以选择不开启此端口。有关端口配置的详细说明, 请参阅 Docker 官方文档: <https://docs.docker.com/engine/containers/run/#exposed-ports>

1. 为什么需要端口 18072?

1. 18072 端口主要用于启用 Supernote 设备与私有云之间的自动同步。如果您需要使用自动同步功能, 则必须在 Docker 主机上开放此端口。如果您确认不会使用自动同步, 则可以关闭此端口。

1. 基于 docker 部署的用户: 可直接删除 Docker 或 Docker Compose 配置中的 `-p 18072:18072` 条目
2. 使用 `install.sh` 脚本部署的用户: 目前端口固定为 18072, 并且默认开启。

容器组件描述

- MariaDB: 使用官方 MariaDB 10.6.24 镜像进行部署, 用于存储用户注册信息、文件同步和其他信息。
- Redis: 使用官方 Redis 7.4.7 镜像进行部署, 用于存储登录信息和其他数据。我们建议将 Redis 数据目录持久化到本地。
- notelib: 官方的 Supernote 笔记转换程序, 基于 Ubuntu 24.04 构建。
- `supernote-service`: Supernote 私有云程序, 基于 Ubuntu 24.04 构建

创建安装目录

示例:

```
mkdir -pv /data/supernote
```

下载初始化数据库文件

示例:

```
cd /data/supernote && curl -O https://private-  
data.supernote.com.cn/cloud/supernotedb.sql
```

创建数据持久化相关目录

示例:

```
mkdir -pv {sndata/cert,supernote_data}
```

使用 Docker Compose 部署 Supernote 私有云

docker-compose 配置清单的下载链接: [docker-compose.yml](#)

- 下载后请勿直接运行此文件。您可以继续参考本文档,了解如何使用 docker-compose 成功部署私有云所需的设置。

创建变量文件

您需要在安装目录下创建一个变量文件,用于设置私有云数据库配置、SSL 配置以及其他相关信息。

变量文件配置说明:

- `DB_HOSTNAME="Mariadb 容器名称"`: 配置 Supernote 私有云应用程序连接数据库服务时使用的容器名称。此名称应为 MariaDB 容器名称,以便容器能够正确解析数据。私有云应用程序使用此名称连接到数据库服务。
- `MYSQL_ROOT_PASSWORD="Mariadb ROOT 密码"`: 设置您的数据库管理员密码并妥善保管。
- `MYSQL_DATABASE="supernotedb"`: 请保留默认值 (supernotedb)。
- `MYSQL_USER="用户名"`: 配置 Supernote 私有云应用程序连接数据库服务时使用的用户名。私有云应用程序将使用此用户名连接到数据库服务。
- `MYSQL_PASSWORD=""`: 这是您在安装数据库服务期间为 Supernote 私有云应用程序设置的数据库用户密码。私有云使用此密码连接到数据库服务。
- `REDIS_PASSWORD="Redis 密码"`: 这是您在安装 Redis 服务期间为 Supernote 私有云应用程序设置的 Redis 密码。私有云应用程序使用此密码连接到 Redis 服务。
- `REDIS_HOST="Redis 容器名称"`: 配置 Supernote 私有云应用程序用于连接 Redis 服务的容器名称。此名称应与 Redis 容器名称匹配,以确保正确解析容器名称。私有云应用程序将使用此名称连接到 Redis 服务。
- `REDIS_PORT="6379"`: 请配置 Redis 容器服务的端口。私有云应用程序将使用此端口连接到容器内的 Redis。此端口无需对外暴露。
- `HTTP_PORT="local http port"`: 请配置私有云容器 Docker 主机的外部连接服务端口。详情请参阅“主机端口映射 19072”。

如何在自定义域名上启用 HTTPS

如果您不需要使用自定义域名启用 HTTPS,请移除以下部署示例中“启用 HTTPS 设置”选项中的相关变量设置;此外,您还需要手动移除 `docker-compose.yml` 配置文件中与 `supernote-service` 容器配置相关的配置项 `./sndata/cert:/etc/nginx/cert`。否则,私有云可能无法启动。

1. 启用 HTTPS 前,请确保 Docker 主机上的 443 端口未被占用。HTTPS 默认使用此端口;使用其他端口会导致浏览器显示不受信任的证书警告。
2. 请确保您的 SSL 证书有效且与您的域名匹配;否则,浏览器将显示不受信任的证书警告。
3. 确认 DNS 可以正确解析您的域名。

4. 如果需要在私有云启用https功能，请在.env文件中增加如下设置

- `DOMAIN_NAME="test-private-cloud.supernote.com":`
- `HTTPS_PORT="本地 HTTPS 端口":` 请配置 Docker 主机需要使用的 HTTPS 端口。
- `SSL_CERT_NAME="您的域名 SSL 证书名称":` 请设置 HTTPS 证书的公钥名称（通常以 `.crt`、`.pem` 等结尾）。如果未设置此配置，则将使用默认名称 `server.crt`。
`SSL_KEY_NAME="您的域名 SSL 密钥名称":` 请设置 HTTPS 证书私钥名称（通常以 `.key` 结尾）。如果未设置此配置，则将使用默认名称 `server.key`。
- 将 `./sndata/cert:/etc/nginx/cert` 设置添加到 `docker-compose.yml` 文件中 `supernote-service` 容器配置部分，如果您尚未修改 `docker-compose.yml` 文件，则此配置已存在于其中。

```
# 示例
touch $(install_dir)/.env

# 请在 .env 文件中设置变量配置信息，如下例所示：
cat .env
# Databases settings
DB_HOSTNAME="supernote-mariadb"
MYSQL_ROOT_PASSWORD="mysqlrootPassword"
MYSQL_DATABASE="supernotedb"
MYSQL_USER="supernote"
MYSQL_PASSWORD="supernotedbpassword"

# Redis settings
REDIS_HOST="supernote-redis"
REDIS_PASSWORD="supernoteredispassword"
REDIS_PORT="6379"

# supernote-service SSL settings
# 如果不需要启用 HTTPS，则无需进行下面的配置。
# 有关 HTTPS 配置的详细信息，请参阅部署文档。
DOMAIN_NAME="supernote.xxxx.com"
HTTPS_PORT="443"
HTTP_PORT="19072"
SSL_CERT_NAME="xxxx.com.crt"
SSL_KEY_NAME="xxxx.com.key"
```

修改docker-compose.yml

1. 默认情况下，下载的 `docker-compose` 文件中除数据库和 Redis 镜像外的所有镜像均由 Supernote 提供。
2. 文件当中所有用到的镜像均从阿里云下载。
3. 完成上述自定义配置后，现在可以使用 `docker-compose` 部署私有云应用程序了。

```
# Check if the configuration is correct.
docker-compose -f docker-compose.yml config -q

# Deploy Supernote Private Cloud
docker-compose up -d
```

如何升级 Supernote 私有云组件

1. 更新前，请确保已根据上述设置配置相关容器的持久目录。

2. 为确保数据完整性，建议备份数据相关目录，包括数据库服务的本地持久目录、Supernote 数据文件的本地持久目录（默认：/{install_dir}/supernote_data）以及 Redis 服务的本地持久目录。
- 有关如何更新数据库配置的说明，请滚动至文档末尾查看“[如何升级 MariaDB 或更新服务](#)”部分。
 - 要升级您的私有云应用程序，您可以直接使用 `docker-compose down` 终止容器，然后手动删除主机上的 `supernote-service` 镜像，再重新执行 `docker-compose up -d` 即可完成更新。

使用 docker run 部署 Supernote 私有云

创建自定义网络

示例:

- 创建一个名为 `supernote-net` 的自定义网络，并确保所有 Supernote 私有云服务都在此网络内运行。

```
docker network create supernote-net
```

安装数据库服务

`--name`: 请设置MariaDB 数据库容器的名称。

`--network`: 请设置运行私有云应用程序的容器配置自定义网络名称，请参考“创建自定义网络”中创建的自定义网络。

`MYSQL_ROOT_PASSWORD`: 请设置数据库管理员密码

`MYSQL_DATABASE`: 请设置supernote私有云使用的数据库 (supernotedb).

`MYSQL_USER`: 请配置私有云应用程序的用户以连接到数据库

`MYSQL_PASSWORD`: 请设置私有云应用程序连接数据库的用户密码。

`-v "local store:/var/lib/mysql"`: 设置本地持久化目录。请注意，为防止数据丢失，我们强烈建议为数据库服务设置本地持久化路径。

`-v $(pwd)/supernotedb.sql:/docker-entrypoint-initdb.d/supernotedb.sql:ro`: 请将 Supernote 私有云所需的数据库初始化文件映射到数据库容器，以启用数据库初始化。

- 参考示例

```
docker run -d \  
--name supernote-mariadb \ # 请设置mariadb容器名称。  
--network supernote-net \ # 请设置运行私有云应用程序的容器配置自定义网络名称，请参考“创建自定义网络”中创建的自定义网络。  
-e MYSQL_ROOT_PASSWORD="Set database administrator password" \ # 请设置数据库的管理员密码  
-e MYSQL_DATABASE=supernotedb \  
-e MYSQL_USER="Set up the database user for the Supernote Private Cloud program" \  
 \ # 请配置私有云应用程序的用户以连接到数据库  
-e MYSQL_PASSWORD="Set the database user password for the Supernote Private Cloud program" \ # 请设置私有云应用程序连接数据库的用户密码。  
-v "Local persistence directory for database data":/var/lib/mysql \  
-v $(pwd)/supernotedb.sql:/docker-entrypoint-initdb.d/supernotedb.sql:ro \  
--restart=unless-stopped \  
mariadb:10.6.24
```

安装 Redis 服务

`--name`: 请设置 Redis 容器名称。

`--network`: 请设置运行私有云应用程序的容器配置自定义网络名称, 请参考"创建自定义网络"中创建的自定义网络。

`--requirepass ''`: 请为 Supernote 私有云应用程序连接 Redis 服务设置密码。

`-v`: 设置本地持久化目录

- 参考示例

```
docker run -d \  
--name "Please set your Redis container name." \ # 请设置redis容器名称  
--network supernote-net \ # 请设置运行私有云应用程序的容器配置自定义网络名称, 请参考"创建自定义网络"中创建的自定义网络。  
-v "Local Persistence Directory for Redis Data":/data \  
--restart=unless-stopped \  
redis:7.4.7 \  
redis-server --requirepass 'Set Redis Password' --dir /data --dbfilename  
dump.rdb
```

安装 Notelib 服务

- 运行以下命令, 即可快速构建 Supernote 私有云笔记转换程序

```
docker run -d --name notelib --network supernote-net crpi-ynh0vytd4hxq64c7.cn-shanghai.personal.cr.aliyuncs.com/sn-private-cloud/notelib:latest
```

安装 Supernote 私有云程序

`--name`: 请为运行私有云应用程序的容器设置名称。

`--network`: 请设置运行私有云应用程序的容器配置自定义网络名称, 请参考"创建自定义网络"中创建的自定义网络。

`-p 18072:18072`: 如果您需要使用自动同步功能, 请打开此端口。

`-p "local port:8080"`: 请为私有云容器的 Docker 主机配置外部连接服务端口。详情请参阅"主机端口映射 19072"。

`-p local https port:443`: 请设置你docker主机需要使用的https端口

`-v $(pwd)/sndata/cert:/etc/nginx/cert`: 设置您的SSL证书路径, 请将SSL证书重命名为 `server.crt` 及 `server.key`, 如果未设置此项将默认使用Supernote自签名证书

`-v $(pwd)/sndata/recycle:/home/supernote/recycle`: 设置回收站的本地持久路径

`-v $(pwd)/supernote_data:/home/supernote/data`: 设置笔记和同步文件的本地持久路径。

`-v $(pwd)/sndata/logs/cloud`: 设置前端程序日志的本地持久化路径

`-v $(pwd)/sndata/logs/app:/home/supernote/logs`: 设置私有云服务日志的本地持久化路径

`-v $(pwd)/sndata/logs/web:/var/log/nginx`: 设置 Web 服务日志的本地持久化路径

`-v $(pwd)/sndata/convert:/home/supernote/convert`: 设置转换后笔记的本地持久路径

`-v $(pwd)/sndata/cert:/etc/nginx/cert`: 配置您的 SSL 证书路径。请将 SSL 证书重命名为 `server.crt` 和 `server.key`。如果未设置，则默认使用 Supernote 自签名证书。

`-e DB_HOSTNAME`: 请设置您的 MariaDB 容器名称。请参考数据库服务安装过程中 `--name` 字段中指定的名称。

`-e MYSQL_DATABASE`: 请设置 supernote 私有云使用的数据库 (supernotedb)。

`-e MYSQL_USER`: 这是在安装数据库服务期间为 Supernote 私有云程序设置的数据库用户。请参阅“安装数据库服务”部分中的 `MYSQL_USER` 变量。

`-e MYSQL_PASSWORD`: 这是在安装数据库服务期间为 Supernote 私有云程序设置的数据库用户密码。请参阅“安装数据库服务”部分中的 `MYSQL_PASSWORD`。

`-e REDIS_HOST`: 请设置 Redis 容器名称。请参考 Redis 服务安装过程中 `--name` 字段中指定的名称

`-e REDIS_PORT`: 请配置 Redis 容器服务的端口。私有云应用程序将使用此端口连接到容器内的 Redis 服务器。此端口无需对外暴露。

`-e REDIS_PASSWORD`: 这是在安装 Redis 服务期间为 Supernote 私有云程序设置的 Redis 密码。请参阅“安装 Redis 服务”部分中的 `--requirepass` 选项。

`-v /etc/localtime:/etc/localtime:ro`: 设置 Supernote 私有云程序的本地时区。

`-e DOMAIN_NAME="your.domain.com"`: 设置 HTTPS 所使用的域名，将 `your.domain.com` 替换为您的实际域名，域名需与 SSL 证书中的主题名称匹配

`-e SSL_CERT_NAME="your domain ssl cert name"`: 请设置 HTTPS 证书公钥名称（通常以 `.crt`、`.pem` 等结尾）。如果未设置此配置，则将使用默认名称 `server.crt`。

`-e SSL_KEY_NAME="your domain ssl key name"`: 请设置 HTTPS 证书私钥名称（通常以 `.key` 结尾）。如果未设置此配置，则将使用默认名称 `server.key`。

如何使用自定义域名开启HTTPS

1. 开启HTTPS之前请确认您的docker主机上的443端口是否已经被占用，HTTPS协议默认使用此端口，如果使用其它端口则浏览器会提示证书不可信
2. 请确保SSL证书在有效期内，且与您的域名匹配，否则浏览器会提示证书不可信
3. 确认您的域名是否能够被DNS正常解析
4. 有关开启https的设置如下

`-p local https port:443`: 请设置你docker主机需要使用的https端口，推荐使用 `443:443`（标准 HTTPS 端口）

`-v $(pwd)/sndata/cert:/etc/nginx/cert`: 设置您的SSL证书挂载路径，请将您的SSL证书上传至您的docker主机的 `$(pwd)/sndata/cert` 目录，暂时不支持自定义证书目录

`-e DOMAIN_NAME="your.domain.com"`: 设置HTTPS所使用的域名，将 `your.domain.com` 替换为您的实际域名，域名需与 SSL 证书中的主题名称匹配

`-e SSL_CERT_NAME="your domain ssl cert"`: 请设置HTTPS证书公钥名称(通常以 `.crt`、`.pem` 等结尾)，如果未设置该配置将使用默认名称 `server.crt`

`-e SSL_KEY_NAME="your domain ssl key"`: 请设置HTTPS证书私钥名称(通常以 `.key` 结尾)，如果未设置该配置将使用默认名称 `server.key`

使用以下命令可快速运行具备 HTTPS 功能的 Supernote 私有云实例。请注意，您需要根据自身配置修改上述持久化设置和变量。

```
docker run -d \
```

```

--name supernote-service \
--network supernote-net \
-p 18072:18072 \
-p "local port":8080 \
-p "local https port:443" \
-v $(pwd)/sndata/recycle:/home/supernote/recycle \
-v $(pwd)/supernote_data:/home/supernote/data \
-v $(pwd)/sndata/logs/cloud:/home/supernote/cloud/logs \
-v $(pwd)/sndata/logs/app:/home/supernote/logs \
-v $(pwd)/sndata/logs/web:/var/log/nginx \
-v $(pwd)/sndata/convert:/home/supernote/convert \
-v $(pwd)/sndata/cert:/etc/nginx/cert \
-v /etc/localtime:/etc/localtime:ro \
-e DB_HOSTNAME="Please set your MariaDB container name." \
-e MYSQL_DATABASE=supernotedb \
-e MYSQL_USER="Enter the private cloud database user you set when creating the
database container." \
-e MYSQL_PASSWORD="Enter the private cloud database user password you set when
creating the database container." \
-e REDIS_HOST="Please set your Redis container name." \
-e REDIS_PORT="Please configure the port for your Redis container." \
-e REDIS_PASSWORD="Enter the password you set when creating the Redis
container." \
-e DOMAIN_NAME="your.domain.com" \
-e SSL_CERT_NAME="your domain ssl cert name" \
-e SSL_KEY_NAME="your domain ssl key name" \
crpi-ynh0vytd4hxq64c7.cn-shanghai.personal.cr.aliyuncs.com/sn-private-
cloud/supernote-service:latest

```

如何升级 Supernote 私有云组件

- 1.更新前，请确保已按照上述设置为相关容器设置了持久目录。
- 2.为确保数据完整性，我们建议备份与数据相关的目录，包括数据库服务的本地持久目录、Supernote 数据文件的本地持久目录（默认：/{install_dir}/supernote_data）以及 Redis 服务的本地持久目录。

如何升级 Supernote 私有云使用的容器镜像

注意：此过程不包含 MariaDB 数据库服务的升级。

1. 请确保最新版本容器使用的持久化目录与初始安装时使用的路径一致。
2. 从 Docker 主机上移除需要升级的容器。
3. 我们的镜像托管在阿里云，可从 [阿里云镜像](#) 获取最新版本镜像，镜像标签会始终为"latest"，然后通过重新安装将容器升级到最新版本。默认情况下，我们仅提供 notelib 和 supernote-service 的更新。

如何升级 MariaDB 或更新服务

1. MariaDB 数据库服务有两种更新方式。
2. **升级 MariaDB 版本：**对于稳定系统，通常不建议直接升级 MariaDB 数据库版本，因为这很容易导致兼容性问题，并影响私有云的正常运行。我们目前正在测试私有云提供商支持的 MariaDB 版本。
3. **更新 supernotedb.sql 脚本：**通常情况下，更新 MariaDB 数据库时，我们只需要更新 supernotedb 使用的 SQL 脚本（有关 SQL 脚本的下载说明，请参阅“下载数据库文件”）。关于更新 SQL 脚本，请注意，您应该确保 MariaDB 容器使用的数据目录已持久化到 Docker 主机上的本地路径。

更新方法如下（建议使用 SQL 脚本进行更新）

1. **使用重建容器的方法：** 您可以将 SQL 脚本下载到 Docker 私有云主机的安装目录，替换上次安装时下载的 [supernotedb.sql](#) 脚本。为此，您需要先移除之前运行的 MariaDB 容器，然后使用“安装数据库服务”命令重新安装。
2. **使用 SQL 脚本进行更新：** 您可以将 SQL 脚本下载到 Docker 主机私有云的安装目录，替换上次安装时下载的 [supernotedb.sql](#) 脚本。替换完成后，您可以使用 `docker exec` 命令直接在容器中手动执行更新。具体步骤如下：

```
# To avoid file system caching issues, the MariaDB container needs to be
restarted.
docker restart "Mariadb Container Name"

# Enter the container using the following command.
docker exec -ti "Mariadb Container Name" /bin/bash

# Check if the latest SQL script has been successfully mounted.
ls /docker-entrypoint-initdb.d/supernotedb.sql

# Update SQL script:
Method 1: mysql -u "DB Username" -p supernotedb < /docker-entrypoint-
initdb.d/supernotedb.sql

#Enter data user password
Method 2:
mysql -u "DB Username" -p
sql: use supernotedb;
sql: source /docker-entrypoint-initdb.d/supernotedb.sql
```