my:D CONNECT

my:D Data Server 설치 안내서

SNP LAB

Security & Privacy Laboratory

버전: v1.0.0

마지막 수정일: 2023.07.03

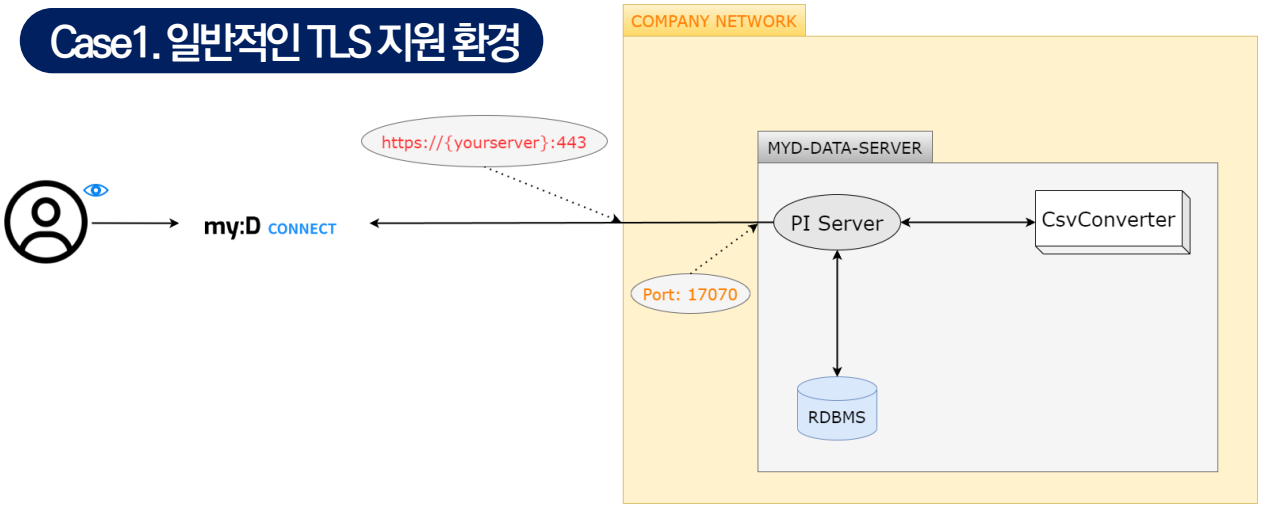
개정 이력

#	날짜	작성자	변경 내역
0	2023-07-03	에스앤피랩	최초 작성

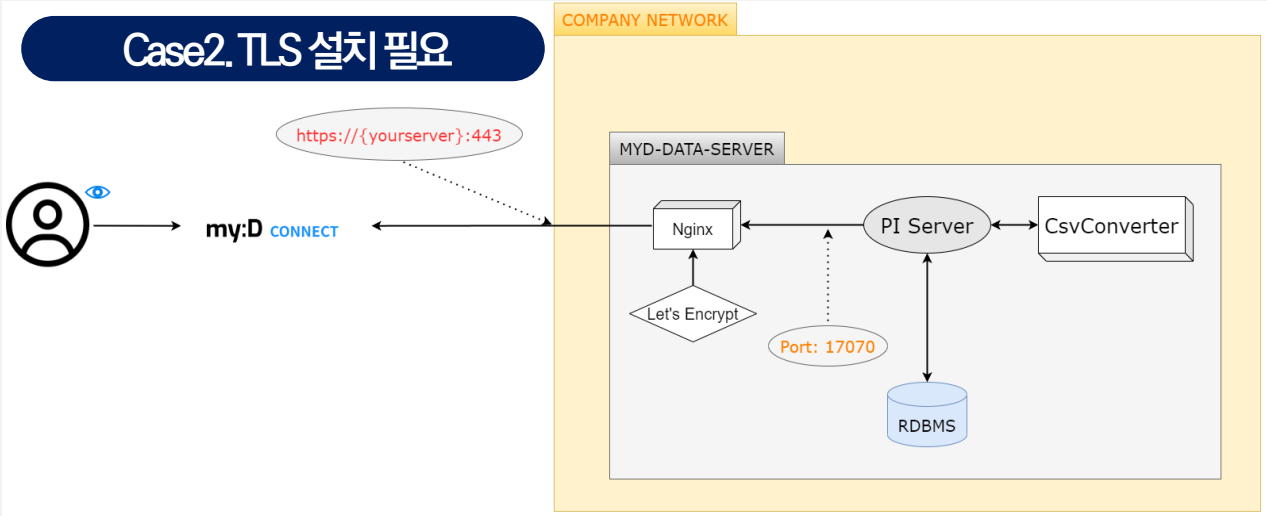
차례

1. 서비스 구조

Case1. 일반적인 TLS 지원 환경



Case2. TLS 설치 필요



구성 요소 (도커 컨테이너 이름)	설명
CsvConverter (Data-Analysis)	웹 응용프로그램 서버 • JSON 형식 데이터를 CSV 형식으로 변환하는 역할 • tiangolo/uwsgi-nginx-flask:python3.9 기반
DB (b2b.on-premise.db)	데이터베이스 • mariadb:10.1.34 기반
Server (b2b.on-premise)	웹 응용프로그램 서버 • 암호화/복호화 및 데이터 수신 검증 • 유일하게 외부에 노출되는 서버 (17070 포트) • openjdk:11.0.10-slim 기반

권장 사양

- Intel i7 4790K or later
- Memory 6GB or more
- Ubuntu 20.04.6 version x86_64

3. 설치 준비

my:D Data Server는 Linux 기반의 Docker 구동 가능한 환경에서 동작
Ubuntu 20.04 기준으로 테스트되었으며 Server, Desktop 관계 없이 사용 가능

1 Git repository 접근 권한

GitLab을 통해 my:D Data Server를 다운로드하여 설치할 수 있습니다.
SNPLab에 GitLab 계정을 전달하면 접근 권한을 부여합니다.

2 Git 설치

Git repository로부터 서버 구성 요소를 다운로드하기 위해 Git을 설치해야 합니다.

```
$ sudo apt install git
```

3 Docker 설치

Linux 상에서 Docker 컨테이너를 구동하기 위해 Docker 엔진을 설치해야 합니다.
Ubuntu 기준으로는 아래와 같이 진행할 수 있습니다.

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Ref: https://docs.docker.com/engine/install/ubuntu/

4 Docker-Compose 설치

Linux 내에 컨테이너를 배포하기 위해서는 docker-compose 설치가 필요합니다.

```
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
$ docker-compose -version
docker-compose version 1.29.2, build 5becea4c
```

Ref: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-20-04

4. 설치

Git을 통해 파일 받기

1 Repository 설치

git clone으로 repository 설치하기

```
$ git clone https://gitlab.com/snplab1/myd-data-server.git
$ cd myd-data-server
```

2 DB password 설정

데이터베이스는 사용자의 자산이므로 DB password 역시 사용자가 설정해야 합니다.
아래와 같이 .env.prod.hostmode 파일에서 PASSWORD 항목 2개를 찾아 변경합니다.

```
$ vi .env.prod.hostmode

MYSQL_ROOT_PASSWORD=inputyourrootpassword    // replace this
MYSQL_USER=cranberry
MYSQL_PASSWORD=inputyourpassword             // replace this
```

중요: 한번 설정된 password는 변경시 데이터베이스 내부의 데이터를 읽지 못하게 될 수 있으므로 신중히 기억하거나 안전하게 보관해야 합니다.

3 Docker container 시작

아래 명령어를 통해 서버를 시작할 수 있습니다.
최초 구동에는 python 패키지 설치에 시간이 걸리며 네트워크 상태에 따라 3분 가량 걸릴 수도 있습니다.

```
$ docker-compose up -d
Creating Data-Analysis     ... done
Creating b2b.on-premise    ... done
Creating b2b.on-premise.db ... done
```

5. 설치 검증

설치 후 동작 확인

1 웹 브라우저 접근

설치 후 아래 주소를 웹 브라우저로 접근하여 동작을 확인합니다.

```
http://{your-ip-address}:17070
```

2 확인

정상적으로 설치되었다면 아래와 같이 표시됩니다.

```
MyD data server status

Server version "1.0.0 (2023.07.01)"
CsvConverter "1.0.0 (2023.07.01)"

URL location http://{your-ip-address}:17070
```

6. 업그레이드

Docker 명령어를 사용하여 My:D Data Server를 최신 코드로 반영하기

1 코드 최신화

git pull 을 해주어야 최신 코드가 반영됩니다.

```
$ git pull
$ docker-compose down
$ docker-compose up -d
```

2 확인

정상적으로 설치되었다면 아래와 같이 표시됩니다.

```
MyD data server status

Server version “2.0.0 (2023.07.01)"
CsvConverter “2.0.0 (2023.07.01)"

URL location http://{your-ip-address}:17070
```

7. 외부 IP 및 HTTPS 설정 (자체 HTTP 제공 불가능한 경우)

외부 IP나 도메인은 있지만 HTTPS 제공이 불가한 경우 Nginx 와 Let's Encrypt를 통해서 HTTPS 제공 가능

1 Nginx 설정 전 준비

Nginx 설정 이전에 사용자 도메인의 IP가 현재 설치중인 서버인지, 그리고 현재 설치중인 서버의 80 포트로 포워딩되는지 확인해 주십시오.
(SNPLab은 도메인을 제공하지 않습니다.)

2 Nginx 설정

Nginx Install Guide

TLS 인증서가 발급되기 전에는 443 포트에 대한 기술 없이 **default.conf** 를 사용해야 합니다.
먼저 **nginx/conf/initial.conf** 를 열어 example.org를 사용자의 도메인명으로 변경합니다.

nginx/conf/initial.conf

```
server {
    listen 80;
    listen [::]:80;
    server_name example.org www.example.org;
    server_tokens off;

    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
      }
    location / {
        return 301 https://example.org$request_uri;
      }
    }
```

변경 이후에는 initial.conf 파일 내용을 default.conf 파일에 복사

```
$ cp nginx/conf/initial.conf nginx/conf/default.conf
$ docker-compose -f docker-compose.nginx.yml up -d webserver
```

7. 외부 IP 및 HTTPS 설정 (자체 HTTP 제공 불가능한 경우)

도메인 연결

3 Nginx구동 후 도메인 확인

Nginx가 정상적으로 구동되었다면 아래 명령어를 통해 도메인이 Nginx로 제대로 포워딩되는지 확인합니다.
정상적이라면 "The dry run was successful" 메세지가 출력되어야 합니다.

```
$ docker compose -f docker-compose.nginx.yml run --rm  certbot certonly --webroot --webroot-path /var/www/certbot/ --dry-run -d {yourserver}
```

4 인증서 발급

아래 명령어를 통해 실제 인증서를 발급받아 주십시오.

```
$ docker compose -f docker-compose.nginx.yml run --rm  certbot certonly --webroot --webroot-path /var/www/certbot/ -d {yourserver}
```

이제 인증서와 키가 모두 생성되었습니다.
(fullchain.pem, privkey.pem) 인증서가 준비된 후에는 /nginx/conf/full.conf 에서 example.org 로 표기된 부분을 모두 사용자의 도메인으로 변경해 주십시오.

7. 외부 IP 및 HTTPS 설정 (자체 HTTP 제공 불가능한 경우)

도메인 연결

/nginx/conf/full.conf

```
server {
  listen 80;
  listen [::]:80;

  server_name example.org www.example.org;
  server_tokens off;

  location /.well-known/acme-challenge/ {
    root /var/www/certbot;
  }
  location / {
     return 301 https://example.org$request_uri;
  }
}

server {
    listen 443 default_server ssl http2;
    listen [::]:443 ssl http2;

    server_name example.org;

    ssl_certificate /etc/nginx/ssl/live/example.org/fullchain.pem;
    ssl_certificate_key /etc/nginx/ssl/live/example.org/privkey.pem;

    location / {
      proxy_pass http://{your ip address}:17070/;
    }
}
```

7. 외부 IP 및 HTTPS 설정 (자체 HTTP 제공 불가능한 경우)

도메인 연결

5 Nginx 재시작

하기의 명령어를 통해 /nginx/conf/default.conf 파일 내용에 변경이 완료된 파일 복사를 진행해 주십시오.

```
$ cp nginx/conf/full.conf nginx/conf/default.conf
```

위의 명령어를 통해 nginx 웹서버를 재시작해 주십시오.

```
$ docker restart b2b.on-premise.web
```

7. 외부 IP 및 HTTPS 설정 (자체 HTTP 제공 불가능한 경우)

인증서 재발급 및 확인

6 인증서 재발급 및 확인

인증서 재발급 명령어는 하기와 같습니다.

```
$ docker compose -f docker-compose.nginx.yml run --rm certbot renew
```

인증서 확인 명령어는 하기와 같습니다.

```
$ docker compose -f docker-compose.nginx.yml run --rm certbot certificates
```

(참고) 자체 HTTPS 제공 가능한 경우

사내에 이미 도메인과 연결된 HTTPS 제공이 가능하면, my:D Data Server가 17070 포트를 열어 두었으므로
사용자의 도메인에 17070 포트를 reverse proxy로 연결하면 안전하게 사용 가능

8. 운영

백업 및 문제 해결

1 백업과 복원

주의사항 : opKey.jck 는 data encrypt key 입니다.
해당 파일이 없다면 암, 복호화가 불가하니 꼭 /store 폴더 내에 있는지 확인 부탁드립니다.

항목	내용	비고
백업	myd-data-server Repository 내에 data 폴더에 있는 /store/_ 와 /mysql/_ 파일들을 저장해야 합니다.	- 백업 관련 스크립트 참고 자료
복원	위에서 저장해 둔 파일들을 data 폴더 내에 붙여 넣습니다.	

2 문제 해결

Docker 로그를 로그 파일로 추출하기

```
$ docker logs b2b.on-premise > log1.txt
$ docker logs b2b.on-premise.db > log2.txt
$ docker logs Data-Analysis > log3.txt
$ ls -al
```

이렇게 진행하면 log1.txt, log2.txt, log3.txt 파일이 생성되는 것을 확인할 수 있습니다. 문제가 있는 경우엔 해당 파일을 SNPLAB로 보냅니다.

SNP LAB

Security & Privacy Laboratory

On-Device MyData Platform, Privacy Technology based on Blockchain & DID