



Google Developer Student Clubs

http, https 그리고 Nginx

Server/Cloud 강승준



Contents

http란?

https란?

https의 동작 과정

Nginx?



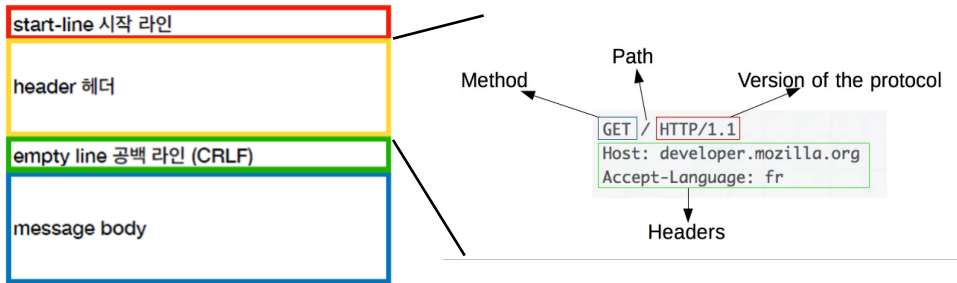
HTTP (HyperText Transfer Protocol)

인터넷을 작동시키는 역할을 하며, 웹 서버 및 웹 브라우저 상호 간의 데이터 전송을
위한 응용 계층 프로토콜

→ 서버/클라이언트 모델을 따라 데이터를 주고 받기 위한 프로토콜



HTTP (HyperText Transfer Protocol)





HTTP (HyperText Transfer Protocol)

1. Stateless, 무상태성

- 서버는 클라이언트의 상태를 보존하지 않는다.

2. Connectionless, 비연결성

- 클라이언트가 서버에 요청을 하고 응답을 받으면 TCP/IP 연결을 끊어 연결을 유지 하지 않는다.



HTTP (HyperText Transfer Protocol)

암호화가 되지 않은 **평문 데이터**를 전송!

→ 비밀번호나 주민등록번호 등을 주고 받게 되면 제3자가
정보를 조회 할 수 있게 된다!



HTTPS (HyperText Transtfer Protocol **Secure**)

HTTP에 **데이터 암호화**가 추가된 프로토콜 네트워크 상에서 중간에

제3자 정보를 볼 수 없도록 **암호화**를 지원한다.



HTTPS (HyperText Transtfer Protocol **Secure**)

1. 대칭키 암호화

- 클라이언트와 서버가 동일한 키를 사용해 **암호화/복호화**를 진행
- 키가 노출되면 매우 위험, But 연산속도가 빠름

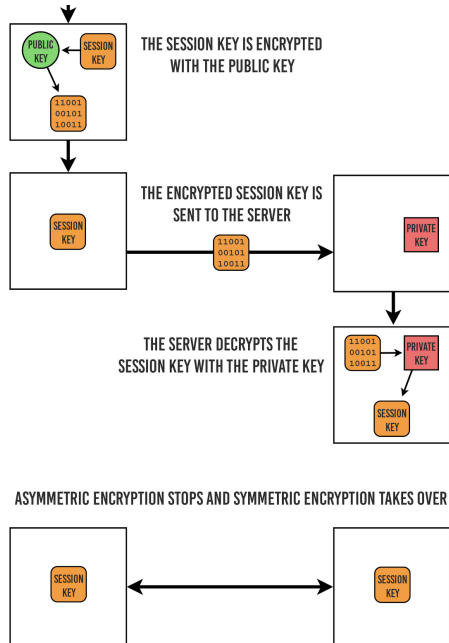
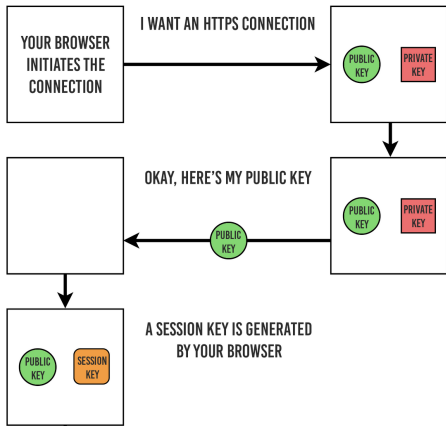
2. 비대칭키 암호화

- 1개의 쌍으로 구성된 공개키와 개인키를 **암호화/복호화** 하는데 사용함
- 키가 노출되어도 비교적 안전, But 연산속도가 느림

HOW HTTPS ENCRYPTION WORKS

YOUR COMPUTER

WEB SERVER





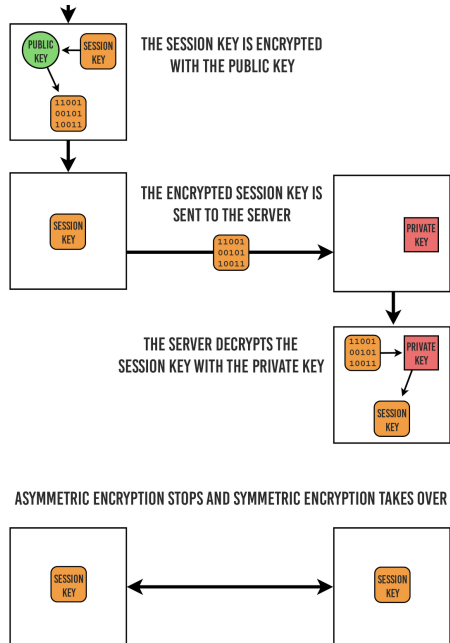
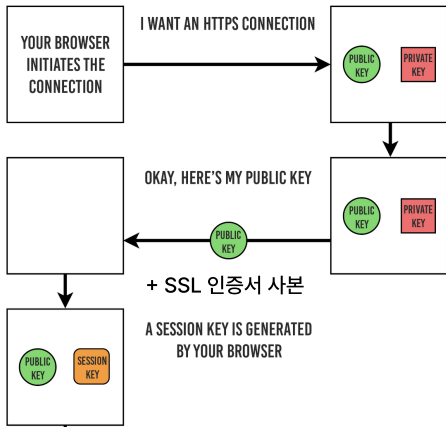
SSL(Secure Socket Layer) 인증서

- **웹 브라우저와 서버간 암호화 통신을 위한 프로토콜**
(웹서버의 공개키, 소유자에 대한 정보 등이 포함)
- **제 3의 인증기관으로부터 발급받음**
(Comodo, Digicert, GlobalSign,,,등등)

HOW HTTPS ENCRYPTION WORKS

YOUR COMPUTER

WEB SERVER





Certbot?

**Let's encrypt를 이용해서 SSL 인증서를
발급, 갱신할 수 있게 도와주는 무료, 오픈소스 툴!**

Nginx



높은 성능과 안정성 그리고 현재 가장 많이 사용되는 웹 서버

Nginx

1. 높은 성능과 적은 메모리 사용
2. 리버스 프록시 사용 가능
3. 로드 밸런싱 가능
4. CORS 깔끔하게 잡기 가능
5. SSL 적용하기

Nginx

1. 높은 성능과 적은 메모리 사용
2. 리버스 프록시 사용 가능
3. 로드 밸런싱 가능
4. CORS 깔끔하게 잡기 가능
5. SSL 적용하기!!!





Nginx

```
skyung7272@instance-1:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-01-17 12:58:41 UTC; 19min ago
     Docs: man:nginx(8)
    Main PID: 1825 (nginx)
      Tasks: 3 (limit: 4680)
     Memory: 5.7M
    CGroup: /system.slice/nginx.service
            └─1825 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─1826 nginx: worker process
                  └─1827 nginx: worker process

Jan 17 12:58:41 instance-1 systemd[1]: Starting A high performance web server and a reverse proxy server...
Jan 17 12:58:41 instance-1 systemd[1]: Started A high performance web server and a reverse proxy server.
skyung7272@instance-1:~$
```




Nginx



Welcome to nginx!

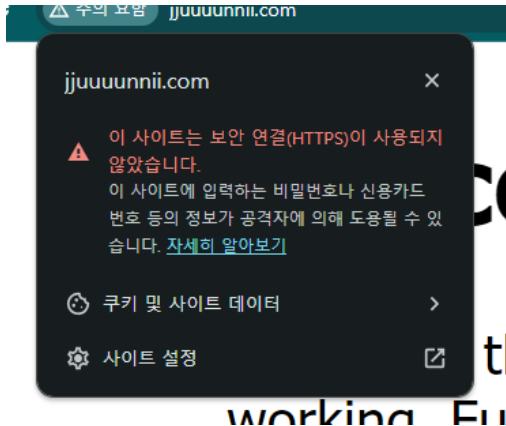
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Nginx





Nginx

Certbot 설치

```
skyung7272@instance-1:~$ sudo apt-get install python3-certbot-nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libatasmart4 libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-
  utils2 libblockdev2 libmbim-glib4 libmbim-proxy libmm-glib0 libparted-fs-resize0
  libqmi-glib5 libqmi-proxy libudisks2-0 libxmlb2 usb-modeswitch
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  certbot python3-acme python3-certbot python3-configargparse python3-future
```



Nginx

설치된 certbot을 이용하여 도메인에 대한 SSL 인증서 발급

```
skyung7272@instance-1:~$ sudo certbot certonly --nginx -d jjuuuunnii.com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): kangseung1110@gmail.com
```

```
-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
```

```
(A)gree/(C)ancel: a
-----
```



Nginx

/etc/letsencrypt/live/(자신이 설정한 도메인) 에

- 1개의 README
- 4개의 pem파일

이 생겼는지 확인

```
root@instance-1:/etc/letsencrypt/live/jjuuunnii.com# ls -la
total 12
drwxr-xr-x 2 root root 4096 Jan 17 15:51 .
drwx----- 3 root root 4096 Jan 17 15:51 ..
-rw-r--r-- 1 root root 692 Jan 17 15:51 README
lrwxrwxrwx 1 root root 38 Jan 17 15:51 cert.pem -> ../../archive/jjuuunnii.com/cert1.pem
lrwxrwxrwx 1 root root 39 Jan 17 15:51 chain.pem -> ../../archive/jjuuunnii.com/chain1.pem
lrwxrwxrwx 1 root root 43 Jan 17 15:51 fullchain.pem -> ../../archive/jjuuunnii.com/fullchain1.pem
lrwxrwxrwx 1 root root 41 Jan 17 15:51 privkey.pem -> ../../archive/jjuuunnii.com/privkey1.pem
root@instance-1:/etc/letsencrypt/live/jjuuunnii.com#
```



Nginx

/etc/nginx/sites-available의 default 파일

```
root@instance-1:/etc/nginx/sites-available# ls  
default
```



Nginx

```
server {  
    listen 80;  
    server_name jjuuuunnii.com;  
    return 302 https://jjuuuunnii.com$request_uri;  
}  
  
server {  
    listen 443 ssl http2;  
    server_name jjuuuunnii.com;  
  
    #ssl 인증서, 비밀키 설정  
    ssl_certificate /etc/letsencrypt/live/jjuuuunnii.com/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/jjuuuunnii.com/privkey.pem;  
  
    # 정적 파일을 제공하는 루트 디렉토리를 지정  
    root /usr/share/nginx/html;  
  
    # 기본 페이지를 설정  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```



Nginx

```
root@instance-1:/etc/nginx/sites-available# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@instance-1:/etc/nginx/sites-available# systemctl restart nginx
root@instance-1:/etc/nginx/sites-available#
root@instance-1:/etc/nginx/sites-available#
```




Google Developer Student Clubs

Nginx

<https://jjuuunnii.com>