SPN v3.0(a.k.a IoTSec) - LoRa Security Project

(Part IV)

08.21.2019 ~

Doc. Revision: 1.5

We Secure the Internet of Things with 2STON SPN.

Michael Chunghan Yi(michael@2ipco.com)
2ip inc.,
Hannam dong, Yongsan Gu, 04418
Seoul

Table of Contents

Part I.

1. LoRa 개요
2. RAKWireless RAK831 LoRaWAN Kit
3. Dragino IoT Kit v2
4. Dragino LG308 LoRa Gateway
5. RAKWireless RAK7258 LoRa Gateway
6. RAKWireless RAK7249 Outdoor LoRa Gateway
7. MatchX MX1702 LoRa Gateway(LBT 지원 모델)

Part II.

8. MultiTech MultiConnect Conduit IP67 LoRa Gateway
9. LoRa Gateway에 SPN S/W Porting하기
10. LoRaServer Project 1 - 설치 & 운용
11. LoRaServer Project 2 - External Interface
12. ThingsBoard Integration - 대박 :)
13. Our LoRa Viewer: OLoRa(= ThingsBoard)

Part III.

14. Outdoor LoRa Node - Libelium
15. URSALink LoRa Products
16. LoRaWAN Stack

Part IV.

17. **Mesh SPN WebUI 작업하기 - 사전 준비 작업**
18. **UrsaLink LoRa Products 소개**
19. **OpenVPN 설정하기**
20. **MX1702 OpenVPN Client 사용하기**
21. **UrsaLink LoRa Gateway ⇔ LoRa Server 연동하기(OpenVPN 기반)**

17. Mesh SPN WebUI 작업 - 사전 준비 작업

이 장에서는 Mesh SPN WebUI 작업을 시작하기 전에 필요한 내용을 정리하고자 한다.

1) Mesh SPN의 정의

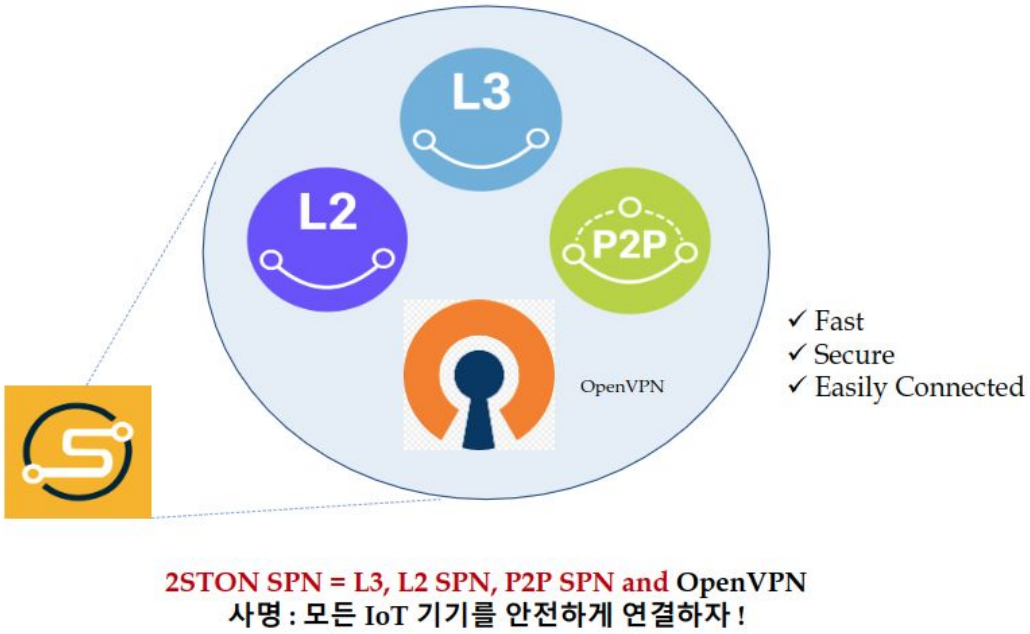
한가지 할 얘기가 있는데, Mesh SPN은 tinc를 기반으로 구현할 예정이었으나, **tinc가 동작 상에 심각한 문제**가 있는 바(왜 지난 5월에는 발견을 못했을까 ?), **wireguard-go version을 Mesh SPN으로 둔갑**(?) 시켜 사용하고자 한다. Wireguard-go의 경우도 tun/tap 기반이니 L3 spn(kernel 기반)과는 구분할 필요가 있을 것 같고, 이미 Mesh SPN이라는 용어를 (외부에) 남발(?)한 상태이니, 어쩔 수가 없을 듯 하다. :) 사실 wireguard도 mesh 적인 특징은 약간 가지고 있어, wireguard-go를 Mesh SPN이라고 하여도 크게 죄의식(?)은 없다. ㅎ ㅎ

Mesh SPN을 필요로 하는 SPNBox는 다음과 같다.

[그림 17.1] Mesh SPN을 필요로 하는 SPNBox(LoRa Gateway)

<현재까지 정리된 사항>

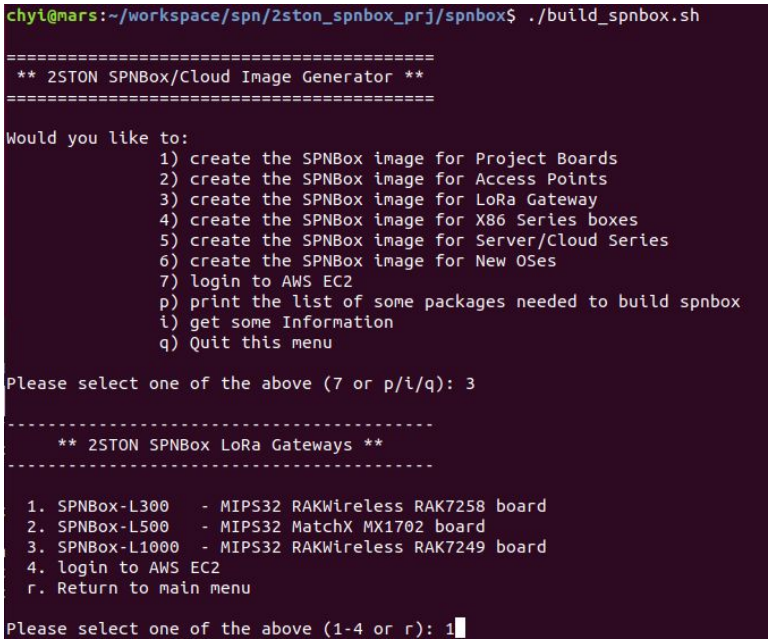
1. SPNBox-L500(MX1702) : **위의 그림과는 달리 p2p SPN으로 간다.** 단, **OpenVPN이 기본적으로 탑재되어 있는 관계로 이의 가능성도 타진 예정**.
2. SPNBox-L1000/L1500(RAK7258/RAK7249) : wireguard-go(=> mesh SPN)으로 간다.
3. UrsaLink : 위의 그림에는 없으나 OpenVPN으로 간다.

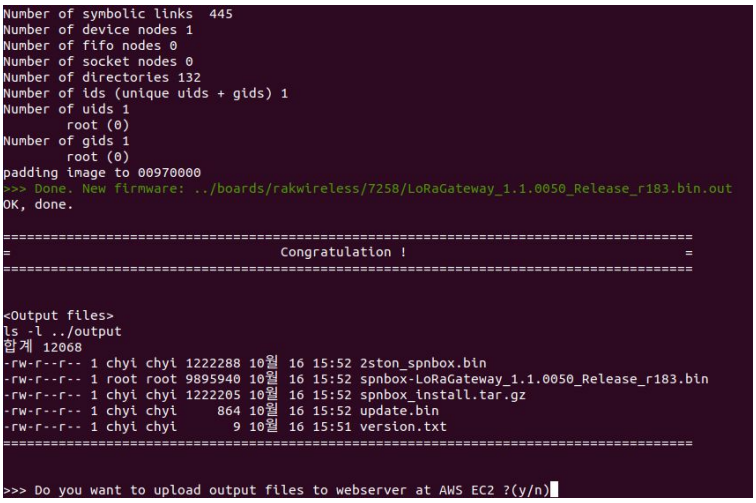


[그림 17.2] SPN에 OpenVPN을 추가시켜야 하지 않을까 ?

2) SPNBox image build 하기

SPNBox(LoRa Gateway 용) 설치 이미지를 생성하는 절차는 다음과 같다. 이미 잘 알고 있는 내용이라 부연 설명은 생략 :) 현재 아래 3가지 model(RAK7258, RAK7249, MX1702)에 대한 image 생성이 가능하다~.





[그림 17.3] LoRa Gateway 용 SPNBox image 생성하기

참고: 위의 build는 Ubuntu 18.04 환경에서만 진행할 수 있다.

<Build 결과물>

spnbox-LoRaGateway_1.1.0050_Release_r183.bin

- SPNBox-L300(RAK7258)용 firmware 이미지, WebUI 상에서 인스톨 가능 이미지
- (주의) **문제 없을 것으로 보이나, 아직 설치 테스트를 해 보지 않았으니, 설치하지 마시기 바람. 추후 다시 공지 하겠음.**

spnbox_install.tar.gz

- **수동 설치 파일, 이 파일을 scp로 target board에 밀어 넣은 후, 설치하시기 바람.**

2ston_spnbox.bin/update.bin/version.txt

- Auto install용 이미지(CLI에서 remote upgrade시 사용). 아직 시험하지 않음.

3) Target board에 SPNBox image 설치하기

이 절에서는 우선 **수동 설치(spnbox_install.tar.gz) 설치 방법 만을 소개**하기로 한다. 나머지 방법은 추후 다시 소개하기로 하겠다.

```
root@SPNBox-L1000:/mnt/mmcblk0p1# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    7552      5700      1852  75% /
/dev/root                 7424      7424         0 100% /rom
tmpfs                    63168       208     62960   0% /tmp
/dev/mtdblock6            7552      5700      1852  75% /overlay
overlayfs:/overlay        7552      5700      1852  75% /
tmpfs                      512         0       512   0% /dev
/dev/mmcblk0p1         15549952      3680  15546272   0% /mnt/mmcblk0p1
```

[그림 17.4] RAK7249에서의 df 명령 실행 모습

$ **scp ./spnbox_install.tar.gz root@172.30.1.21:~/workspace**

➔ 단, target board에서 ~/workspace folder를 미리 만들어 두었다고 가정.
➔ 172.30.1.21은 RAK7258의 WAN(ethernet) port에 해당함.

$ ssh **root@172.30.1.21**

➔ default password: root
➔ spnbox 설치 후에도 아직은 root로 유지됨. 이 부분도 spnbox!로 추후 변경 예정임.

<target board>

cd ~/workspace

mkdir -p /mnt/mmcblk0p1/workspace

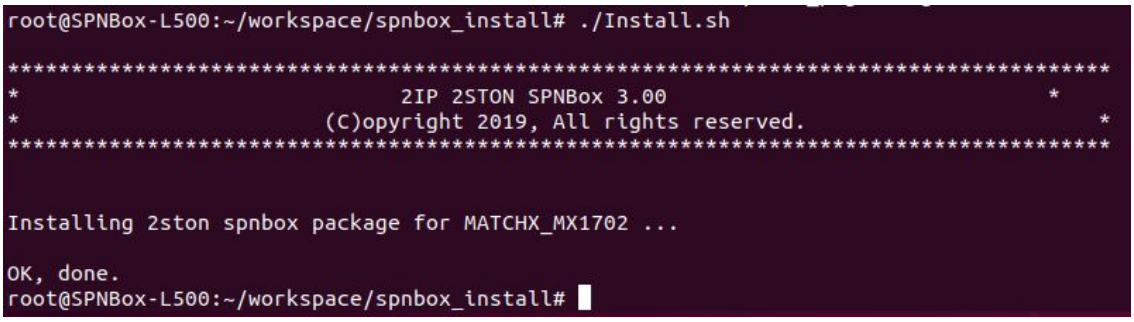
mv ./spnbox_install.tar.gz /mnt/mmcblk0p1/workspace # flash 공간이 부족하니 여기를 활용하자.

 # cd /mnt/mmcblk0p1/workspace

 # tar xvzf spnbox_install.tar.gz

 # cd spnbox_install

 # ./Install.sh



[그림 17.5] SPNBox image 설치하기

참고: 위의 그림은 SPNBox-L500(MX1702)에서 설치한 내용을 나중에 capture하여 여기에 정리한 것임.

 # sync; sync

 # reboot

부팅 후, (정상적으로 설치가 되었다면) 아래와 같이 SPNBox CLI가 출력될 것이다.

```
chyi@mars:~$ ssh root@172.30.1.21
root@172.30.1.21's password:


BusyBox v1.23.2 (2019-08-15 18:31:30 CST) built-in shell (ash)


 -----------------------------------------------------
 RAK Wireless LoraWAN Gateway (1.1.0048_Release r180 20190815)
 -----------------------------------------------------
Build On Oct 16 2019 13:02:19
spnbox-l300> en
spnbox-l300# configure  terminal
spnbox-l300(config)#
  bridge      add/modify the bridge information
  date        Set the date
  enable      Modify enable password parameters
  exit        Exit current mode and down to previous mode
  factory     Go back to the factory default state
  hostname    Set system's network name
  ip          IP information set
  meshvpn     Configure meshvpn tunnel
  nameserver  Config the dns server
  no          Negate a command or set its defaults
  p2p         Configure p2p tunnel
  password    Modify password parameters
  ping        Send echo messages
  se          Configure SoftEther VPN
  sfirewall   Configure spn firewall rules
  show        Show running system information
  spn         Configure SPN rules
  ssh         Open a ssh connection
  swupgrade   spnbox software upgrade
  write       Write running configuration to memory, network, or terminal
spnbox-l300(config)#
```

[그림 17.6] LoRa Gateway 상에서 동작하는 SPNBox CLI(vtysh)

4) Mesh SPN CLI 설정하기

이 절에서는 CLI 상에서 Mesh SPN을 설정하는 방법을 소개하고자 한다. **사실 기존 L3 SPN과 설정 방식은 동일하다.** 차이가 있다면, auto connection 기능이 없다는 것 정도 …

```
spnbox-l300(config)# show running-config
#Writed on Wed Oct 16 07:05:14 2019
hostname spnbox-l300
ip address spn0 10.1.2.50 255.255.255.0
password 8 spYzDw1OqDeMQ
spn link-up
spn listenport 59760
spn peer hgyR8p+gjjrQjLHBrrvGEgxA1ztnyonIF0MdRJHfUzw= allowed-ips 10.1.2.0/24 endpoint 13.124.231.2
9:51820 persistent-keepalive 25
!
spnbox-l300(config)#
```

[그림 17.7] Mesh SPN CLI 설정 모습

<Mesh SPN 설정 절차 요약>

➔ 모두 다 잘 알고 있는 내용이라, firewall(or 공유기)에서 port forwarding 설정을 해 줘야 하는 부분은 별도로 정리하지 않았다.

1. spn ip 설정 변경

spnbox-l300(config)# **no ip address spn0**

spnbox-l300(config)# **ip address spn0 10.1.2.50 255.255.255.0**

2. spn listen port 변경(필요 시)

spnbox-l300(config)# **spn listenport 59760**

3. peer 등록/추가

spnbox-l300(config)# **spn peer hgyR8p+gjjrQjLHBrrvGEgxA1ztnyonIF0MdRJHfUzw= allowed-ips 10.1.2.0/24 endpoint 13.124.231.29:51820 persistent-keepalive 25**

4. 설정 저장

spnbox-l300(config)# **wr**

5. ping test

spnbox-l300(config)# **ping 10.1.2.254**

➔ 참고로 10.1.2.254는 AES EC2(LoRa Server: 13.124.231.29)이며, L3 SPN이 설치되어 있다.

PING 10.1.2.254 (10.1.2.254): 56 data bytes

64 bytes from 10.1.2.254: seq=0 ttl=64 time=4.057 ms

64 bytes from 10.1.2.254: seq=1 ttl=64 time=4.026 ms

64 bytes from 10.1.2.254: seq=2 ttl=64 time=3.995 ms

64 bytes from 10.1.2.254: seq=3 ttl=64 time=3.958 ms

64 bytes from 10.1.2.254: seq=4 ttl=64 time=4.692 ms

^C

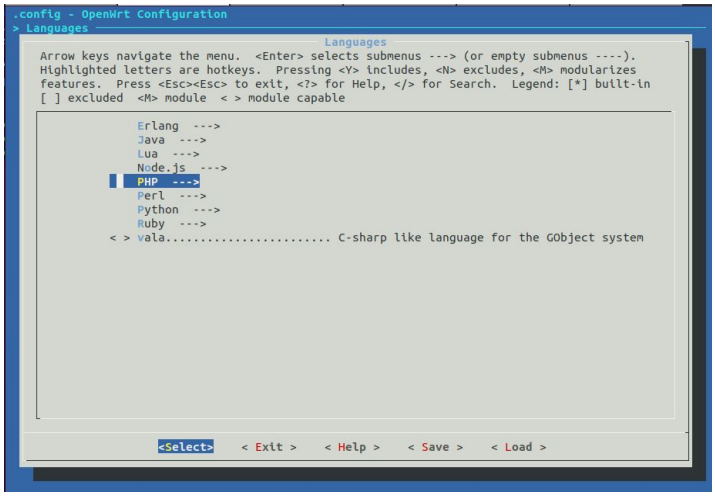
--- 10.1.2.254 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 3.958/4.145/4.692 ms

5) PHP Porting하기

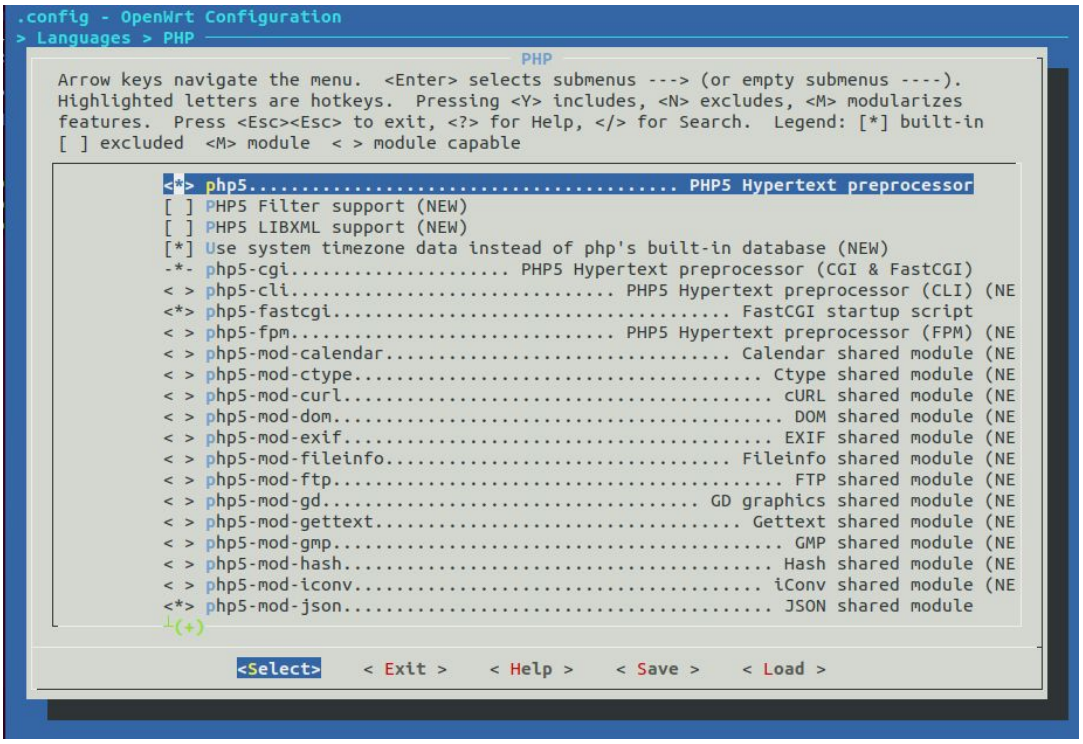
PHP를 MIPS32 용으로 cross-compile해야 한다. **(맨날 porting만 하려니)매우 짜증이 난다 :(**
따라서, 이 절에서는 보다 간편한 방법을 하나 소개하기로 하겠다.



[그림 17.8] OpenWrt PHP 선택 모습

GI.iNet 등에서 PHP 7.x 버젼 사용시 아래의 feature가 필요하였다. 따라서, OpenWrt PHP 설정 시 이를 반영해 주어야 한다.

php7 php7-cgi php7-mod-json php7-mod-session php7-mod-sockets



[그림 17.9] OpenWrt PHP config 조정

참고: RAK831-LoRaGateway-OpenWRT-MT7628 github에 포함되어 있는 있는 openwrt는 15.05 버젼이고, PHP는 5.6.17 버젼이다. PHP version이 좀 낮으나, WebUI 작업에는 문제가 안된다.

이렇게 build하여 얻은 php 관련 파일을 추려 보면 다음과 같다.

```
/etc/php.ini
/etc/php5/json.ini
/etc/php5/session.ini
/etc/php5/sockets.ini

/usr/bin/php-cgi, php-fcgi

/usr/lib/php/json.so
/usr/lib/php/session.so
/usr/lib/php/sockets.so
```

이 상의 내용은 RAK7258/7249, MX1702 용 SPNBox build 시 자동으로 포함되도록 해 두었다. :)

```
root@spnbox-l500:~# php-cgi --version
PHP 5.6.17 (cgi-fcgi) (built: Oct 21 2019 11:02:35)
Copyright (c) 1997-2015 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies
root@spnbox-l500:~#
```

[그림 17.10] Target board에서 php-cgi 실행 모습

주의: 필요시 /etc/php.ini 파일 등을 수정해야 할 수도 있다.

6) WebUI 작업에 들어가기 전에 ..

a) RAK7258/RAK7249는 기존 L3 SPN 화면과 동일(단, 이름은 Mesh SPN으로 표기)하게 만들자. 단, auto connection 기능은 빼자.

b) MX1702는 (현재까지의 생각은) P2P SPN 화면을 추가하는 것으로 하자. 따라서 L3 SPN 화면은 불필요하다.

c) 공통 사항(1): Sfirewall 정도는 살리자.

d) 공통 사항(2): Network 상태 page, 기타 설정 page 등은 그대로 유지하자.

e) **주의: flash memory 공간이 많지 않으니, (GI.iNet의 경우처럼) 최대한 light하게 만들자.**

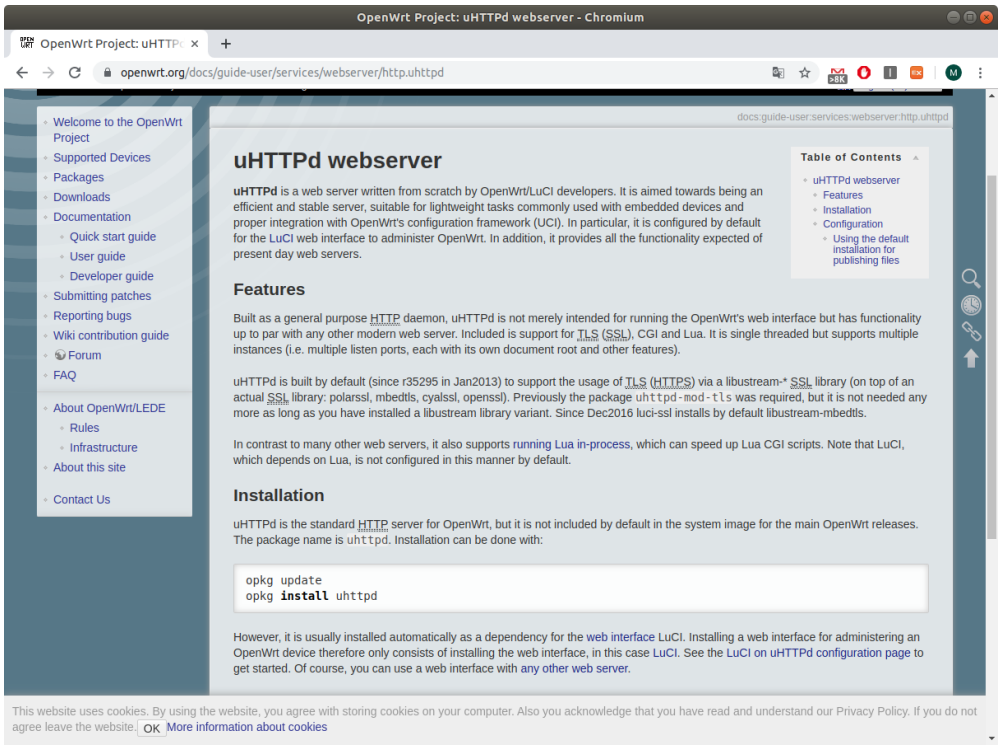
f) webserver는 lighttpd가 아니라, uhttpd 이다.

```
1392 root      1500 S    /usr/sbin/crond -f -c /etc/crontabs -l 5
1406 root      1148 S    /usr/sbin/dropbear -F -P /var/run/dropbear.1.pid -p 22 -K 300
1443 root      2196 S    /usr/sbin/uhttpd -f -h /www -r SPNBox-L300 -x /cgi-bin -u /ubus -t 60 -T 3
1458 root       776 S    /usr/sbin/rteswplugd -r /etc/eswplug.action
```

[그림 17.11] RAK7258/7249, MX1702에 사용된 webserver - **uhttpd**

<uhttpd + PHP 관련 참고 site>

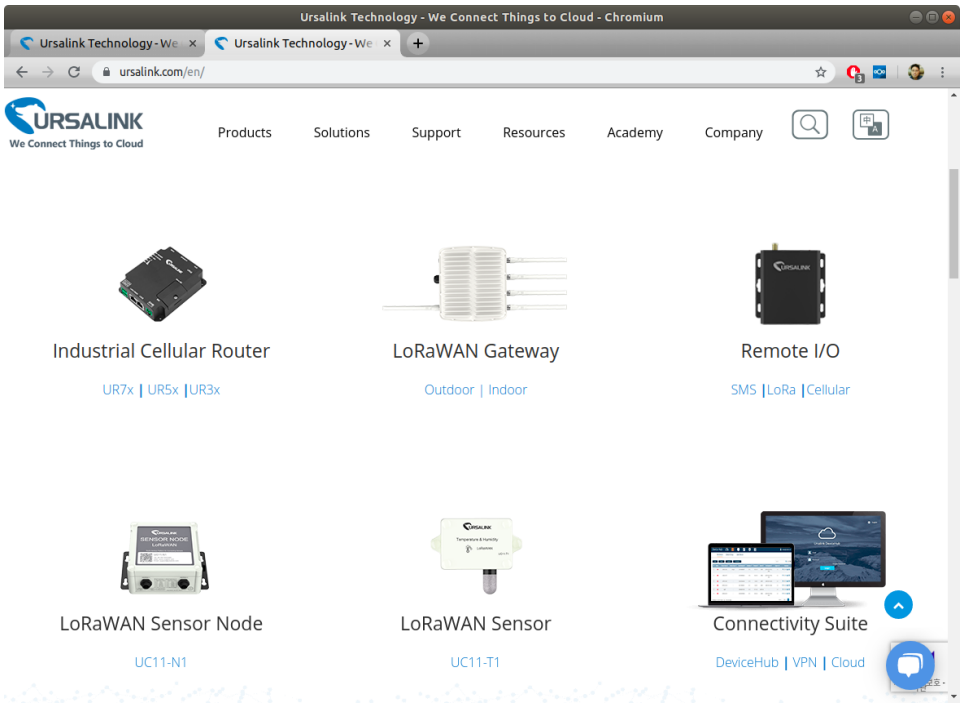
https://stackoverflow.com/questions/19534734/how-to-install-and-configure-php-in-openwrt



[그림 17.12] uHTTPd webserver home page

18. UrsaLink LoRa Products 소개

이번 장에서는 "We Connect Things to Cloud"라는 slogan을 전면에 내세운 **UrsaLink LoRa 제품(industrial router, LoRa 제품 보유)**을 소개해 보고자 한다. **참으로 배울 것이 많은 회사 같다. 우리 회사의 방향성을 이곳에서 찾을 수 있지 않을까 생각해 본다(우리에게 제 2의 GI.iNet이 될 지 검토해 보기로 하자).**



[그림 18.1] URSALINK Home Page

[그림 18.2] URSALINK LoRa Gateways

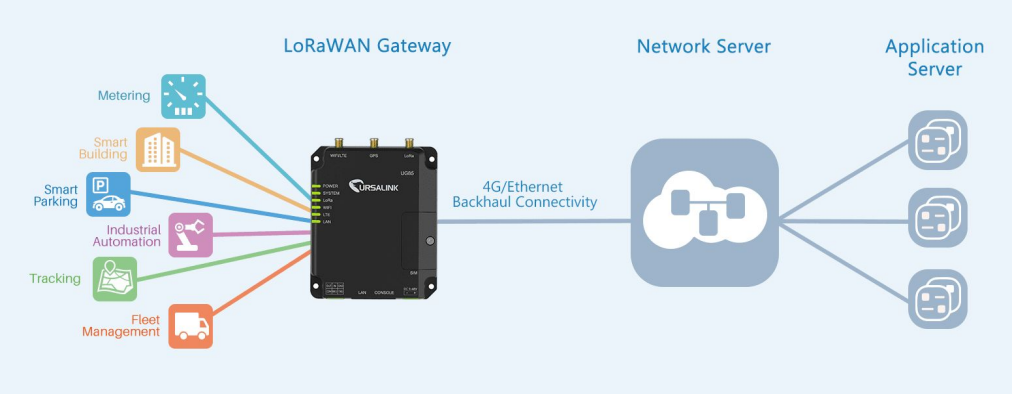
UrsaLink 제품은 두가지 측면에서 (우리에게) merit가 있어 보인다.

1. 산업용(industrial) IoT gateway 제품이라 활용 영역을 실내(Home, 사무실 등)에서 산업 현장으로 확장시키는데 도움을 줄 수 있다.
2. RS232/485, ModBus 등 serial 연결 기능이 제공되고 있어, 다양한 산업 현장의 장비를 연결하는데 유리하다. 그 동안은 ethernet, wi-fi를 통해서만 장치와 통신할 수 있었다(예: IP camera).

1) UG85 Indoor LoRa Gateway

우선 아래 제품을 하나 구매하여 내부를 분석해 보아야 겠다.

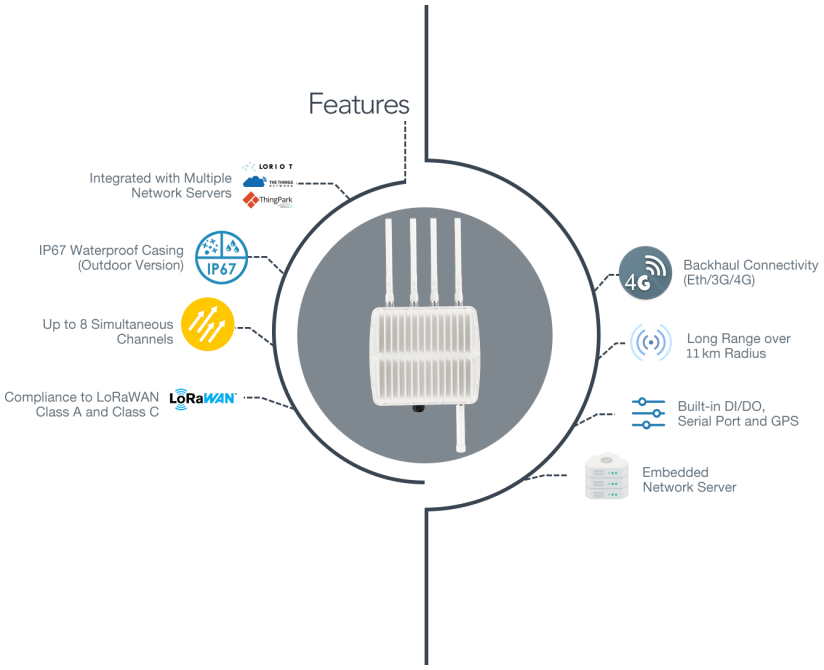
[그림 18.3] URSALINK UG85 Indoor LoRa Gateway(1)



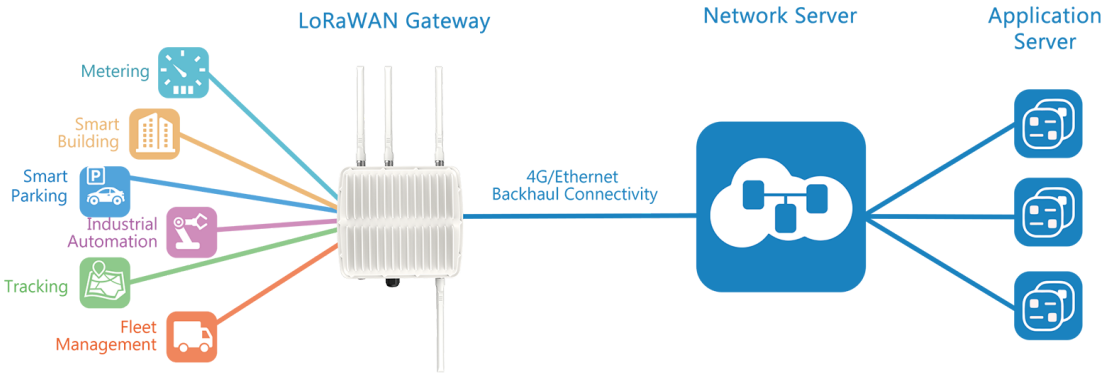
[그림 18.4] URSALINK UG85 Indoor LoRa Gateway(2)

[그림 18.5] URSALINK UG85 Indoor LoRa Gateway(3) - 실제 설치된 모습

2) UG87 Ourdoor LoRa Gateway

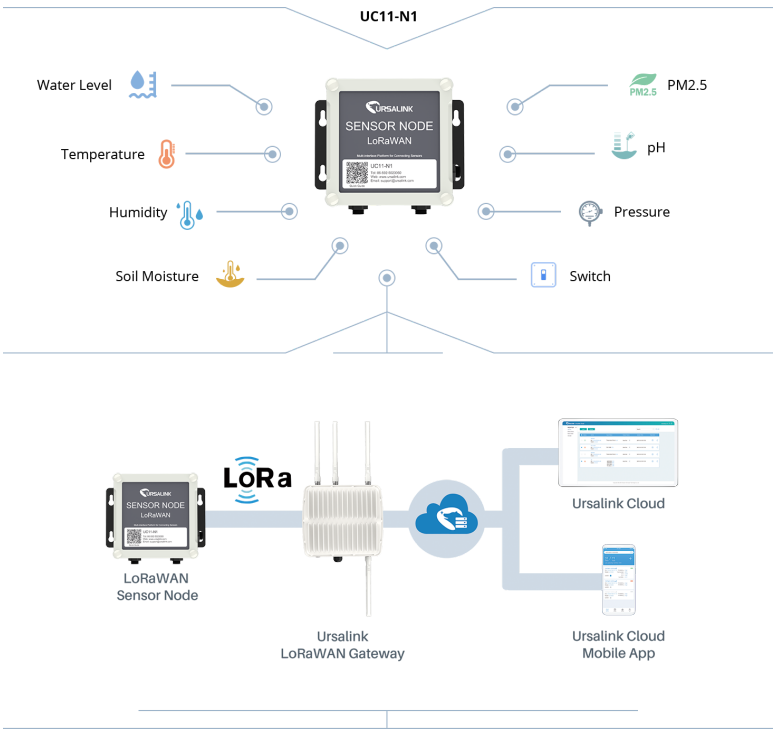


[그림 18.6] URSALINK UG87 Outdoor LoRa Gateway(1)



[그림 18.7] URSALINK UG87 Outdoor LoRa Gateway(2)

3) LoRaWAN Sensors



[그림 18.8] URSALINK UC11-N1 LoRaWan Sensor

UC11-T1 온습도 센서가 도착(11/06/2019)했다. 연결해 보도록 하자.

[그림 18.9] URSALINK UC11-T1 LoRaWan Sensor(온습도 센서)

이를 위해 아래 두개 문서를 참조하도록 하자.

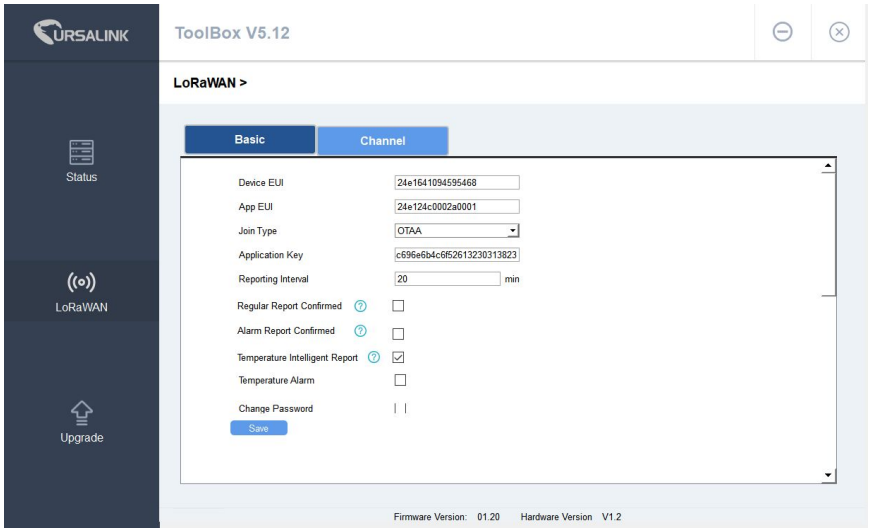
UC11-T1 Temp_Humidity LoRaWAN Sensor User Guide.pdf

UC11-T1 Payload Structure.pdf

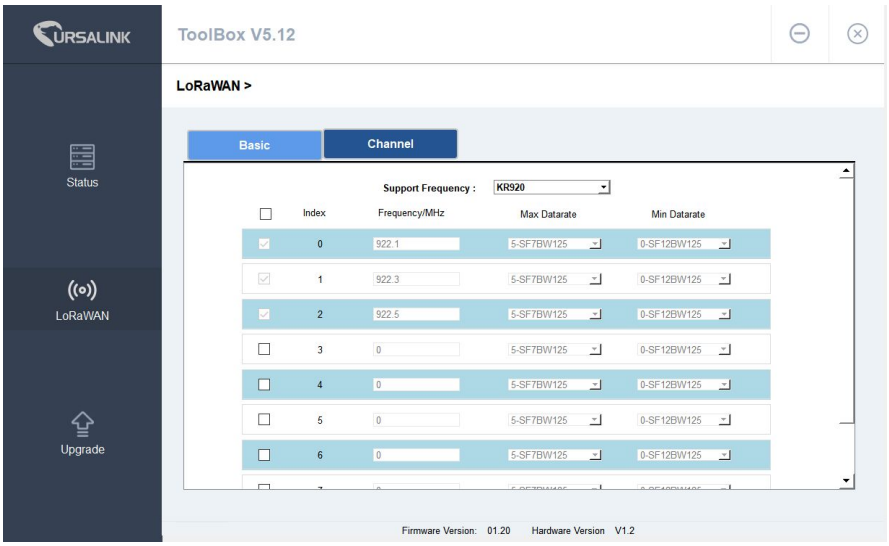
<간략 설정>

1) USB 연결(나사를 풀어 뚜껑을 열어야 함)
2) Power-on(자석을 가져다 댐 - 2초간)
3) 이후 Toolbox 실행
   ➔ COM5, 57600으로 연결

[그림 18.10] URSALINK UC11-T1 LoRaWan Sensor - ToolBox 연결 모습(1)

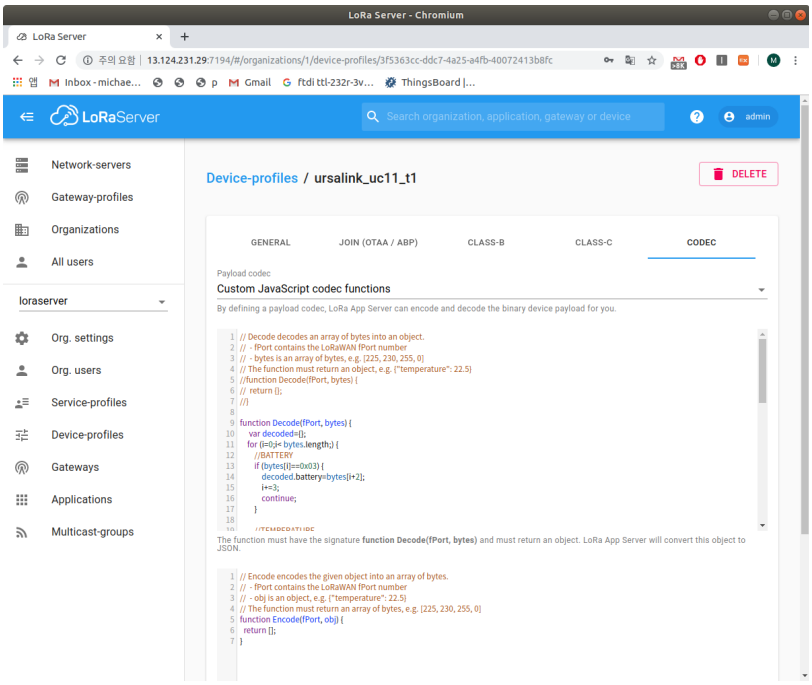


[그림 18.11] URSALINK UC11-T1 LoRaWan Sensor - ToolBox 연결 모습(2)



[그림 18.12] URSALINK UC11-T1 LoRaWan Sensor - ToolBox 연결 모습(3)

LoRaServer와 연동하는 부분은 많이 해 본 내용이라 자세한 설명은 생략하기로 하자. 다만, UC11-T1이 Cayenne LPP format과 정확히 일치하지 않는 관계로, 아래와 같이 Decoding javascript code를 추가해 주어야만 했다.



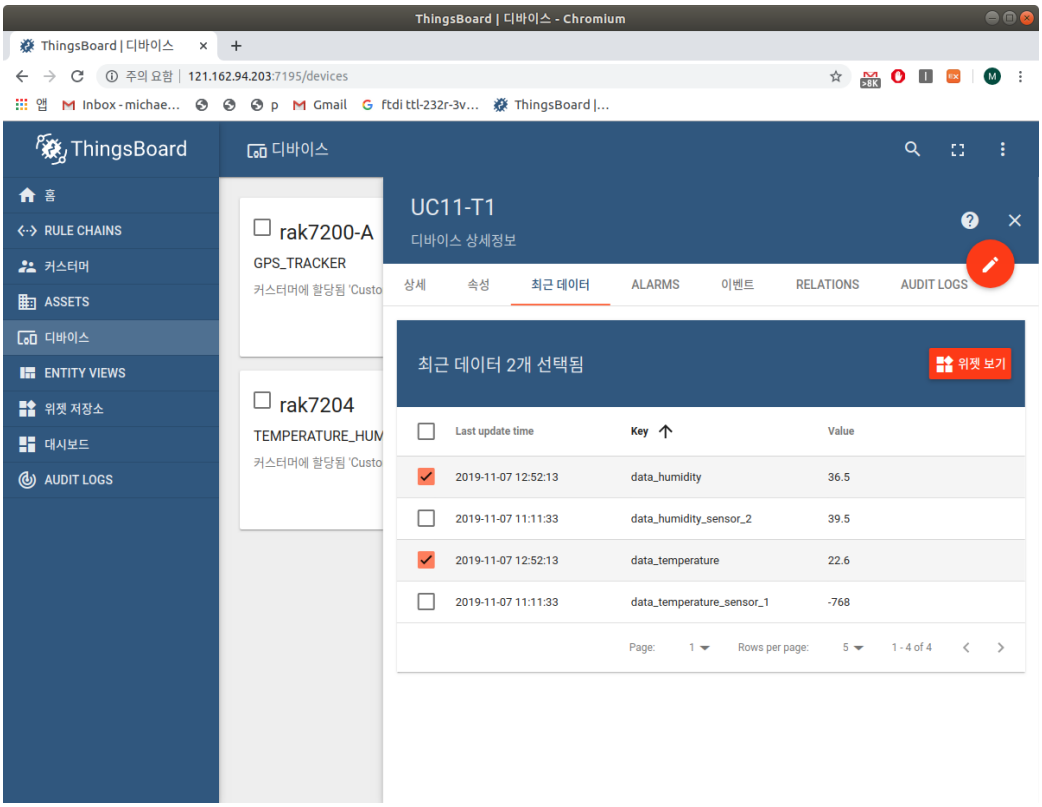
[그림 18.13] URSALINK UC11-T1 LoRaWan Sensor - LoRaServer Codec code 추가

<Decoding Javascript>

```
// Decode decodes an array of bytes into an object.
//  - fPort contains the LoRaWAN fPort number
//  - bytes is an array of bytes, e.g. [225, 230, 255, 0]
// The function must return an object, e.g. {"temperature": 22.5}
//function Decode(fPort, bytes) {
//  return {};
//}

function Decode(fPort, bytes) {
    var decoded={};
   for (i=0;i< bytes.length;) {
      //BATTERY
      if (bytes[i]==0x03) {
         decoded.battery=bytes[i+2];
         i+=3;
         continue;
      }

      //TEMPERATURE
      if (bytes[i]==0x01) {
         decoded.temperature=(readInt16LE(bytes.slice(i+2, i+4)))/10;
         i+=4;
         continue;
      }

      //HUMIDITY
      if (bytes[i]==0x02) {
        decoded.humidity=readUInt8LE(bytes[i+2]) / 2;
         i+=3;
         continue;
      }
   }
  return decoded;
}

function readUInt8LE(bytes) {
   return (bytes & 0xFF);
}

function readInt8LE(bytes) {
   var ref = readUInt8LE(bytes);
   return (ref > 0x7F) ? ref - 0x100 : ref;
}
```

```
function readUInt16LE(bytes) {
    var value = (bytes[1] << 8) + bytes[0];
    return (value & 0xFFFF);
}

function readInt16LE(bytes) {
    var ref = readUInt16LE(bytes);
    return (ref > 0x7FFF) ? ref - 0x10000 : ref;
}
```

ThingsBoard Device에서 확인해 보니, 온도, 습도 값이 정상적으로 올라온다.



[그림 18.14] URSALINK UC11-T1 LoRaWan Sensor - ThingsBoard Device 최신 데이터

[그림 18.15] URSALINK Cellular Remote I/O(1)

[그림 18.16] URSALINK Cellular Remote I/O(2)

4) UG75 Industrial Cellular Router

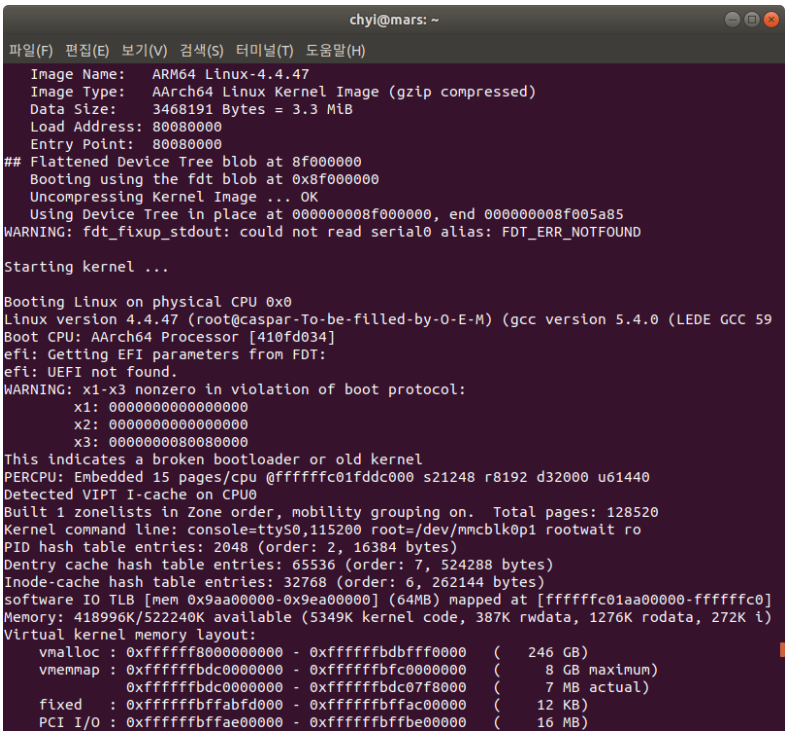
얘는 LoRa Gateway는 아니지만, SPNBox의 모델 후보로 가능할 듯 보여 여기서 잠시 소개하고자 한다.

[그림 18.17] URSALINK UR7X Cellular Router

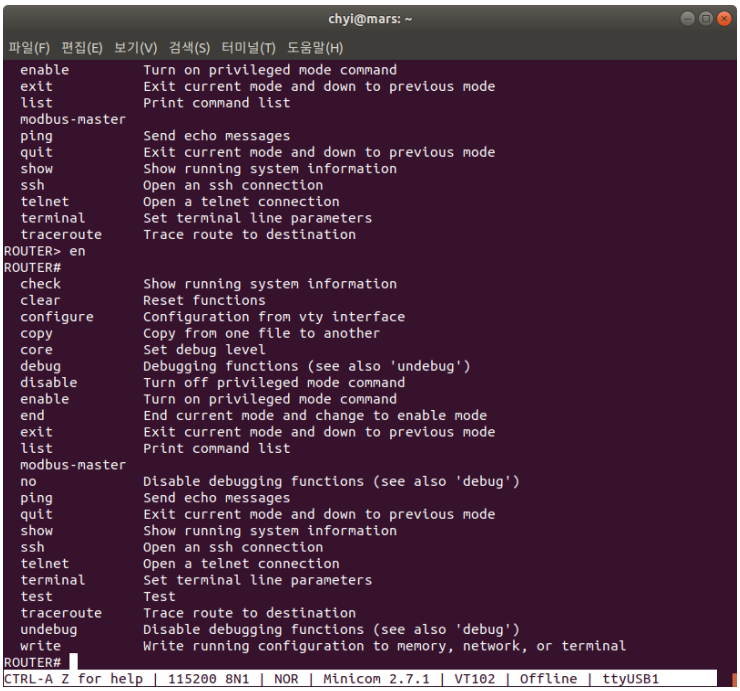
[그림 18.18] URSALINK UR7X Cellular Router 네트워크 구성

5) UrsaLink WebUI & CLI

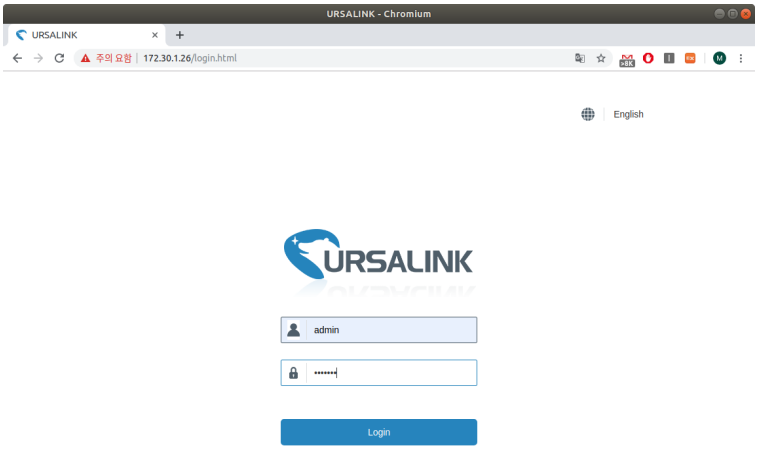
구매한 UG85 장비를 간략히 소개하면 다음과 같다.



[그림 18.19] Console(115200, 8N1)로 확인한 kernel booting 모습



[그림 18.20] CLI 실행 모습



[그림 18.21] WebUI login 화면 - admin/password

참고: default id/pass는 admin/password이며, 현재 테스트 중인 장비의 password는 spnbox!로 변경해 두었음.

6) UrsaLink VPN

UrsaLink 제품은 IPsec, OpenVPN, PPTP, L2TP DMVPN, GRE 등의 다양한 VPN(or Tunnel) 기능을 제공한다.

19. OpenVPN 설정하기

1) OpenVPN 개요

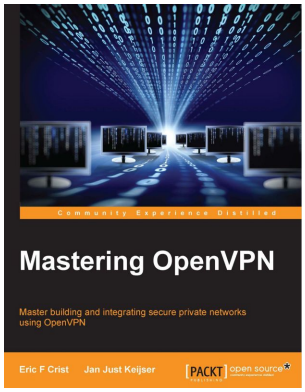
Wireguard(L3 SPN의 근간)는 IPsec이나 OpenVPN에 비해 빠른 속도, 간편한 설정, end-to-end 환경에 적합한 특징 등 많은 장점을 가지고 있다. 때문에 그 동안 IPsec이나 OpenVPN 등은 우리의 관심 대상이 될 수 없었다. 하지만, 앞서 1장에서 언급한 UrsaLink 제품이 불행하게도 내부 시스템(Linux)에 접근할 수 있는 방법을 제공하지 않고 있기 때문에, 어쩔 수 없이 OpenVPN or IPsec 등의 방법을 사용하여 UrsaLink Gateway와 연결할 수 밖에 없는 상황이 되었다.

참고: UrsaLink 제품은 ssh 연결은 가능하나 SPNBox 처럼 CLI가 가로 막고 있어, 내부로 진입(shell을 획득할 수 없음)할 수 없는 문제(?)가 있다.

<UrsaLink Gateway의 활용 가능성/방향>

UrsaLink Gateway(OpenVPN client) ⇒ (internet) ⇒ 2ip LoRaServer(OpenVPN Server)

OpenVPN에 관한 자세한 사항은 아래 책(Mastering OpenVPN) 및 공식 home page를 참조하기 바라며, 여기서는 구체적인 사항(TLS 기반, UDP/TCP tunnel 제공, Tun/Tap 기반 등)은 언급하지 않도록 하겠다.

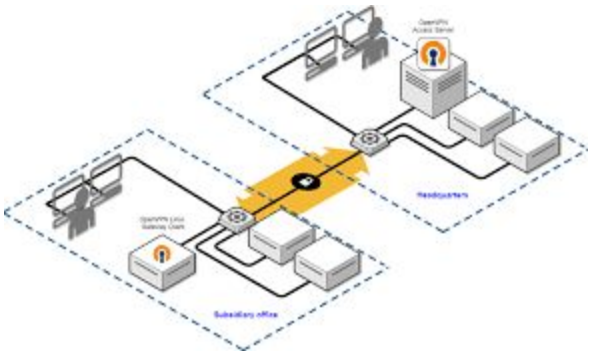


[그림 19.1] OpenVPN 책(**mastering-openvpn-2015-pdf)**

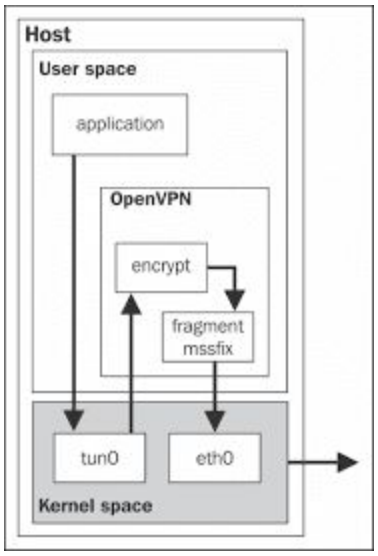
<OpenVPN 공식 site>

https://openvpn.net/

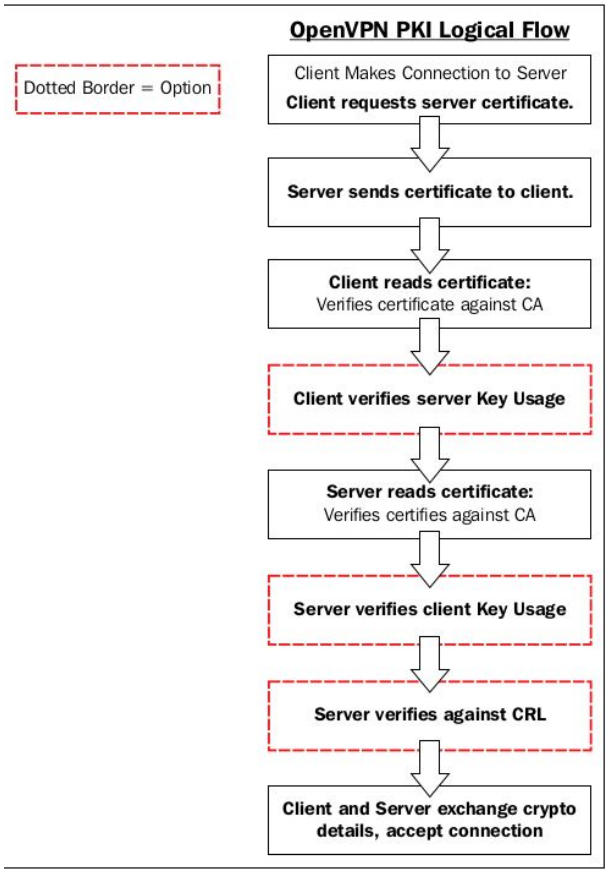
[그림 19.2] OpenVPN logo



[그림 19.3] OpenVPN network



[그림 19.4] OpenVPN 아키텍쳐 - tun/tap 기반



[그림 19.5] OpenVPN PKI flow

2) OpenVPN Server 설치하기

아래 site에는 Ubuntu 18.04에 OpenVPN Server & Client를 설치하는 과정이 상세히 정리되어 있다.

<영문으로 정리>

https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-ubuntu-18-04

<한글로 정리>

https://dejavuqa.tistory.com/243?category=299614

근데, 설치 script를 만들어 OpenVPN Server & Client 설치 & 설정을 하고 싶은데, 위의 내용은 생각보다 절차가 너무 복잡하다. 보다 간편한 방법이 없을까 ? 그 답은 아래 site에 있다.

https://github.com/angristan/openvpn-install

지금 부터는 위의 내용을 토대로 Ubuntu 18.04 환경에 OpenVPN server를 간단하게 설치하는 과정을 소개해 보기로 하겠다.

<Ubuntu 18.04 Server>

➔ AWS EC2(loraserver가 설치된 서버)

$ **curl -O https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh**

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 37787  100 37787    0     0  16336      0  0:00:02  0:00:02 --:--:-- 16336
```

$ **chmod 755 openvpn-install.sh**

$ **sudo ./openvpn-install.sh**

➔ 이 script를 실행하면 몇가지 내용을 확인하는 절차가 있기는 하지만, 한번에 OpenVPN server를 설치할 수 있다.

```
Welcome to the OpenVPN installer!
The git repository is available at: https://github.com/angristan/openvpn-install

I need to ask you a few questions before starting the setup.
You can leave the default options and just press enter if you are ok with them.

I need to know the IPv4 address of the network interface you want OpenVPN listening to.
Unless your server is behind NAT, it should be your public IPv4 address.
IP address: 13.124.231.29

Checking for IPv6 connectivity...

Your host does not appear to have IPv6 connectivity.
```

Do you want to enable IPv6 support (NAT)? [y/n]: **n**

What port do you want OpenVPN to listen to?

1) Default: 1194

2) Custom

3) Random [49152-65535]

Port choice [1-3]: **1**

What protocol do you want OpenVPN to use?

UDP is faster. Unless it is not available, you shouldn't use TCP.

1) UDP

2) TCP

Protocol [1-2]: **1**

What DNS resolvers do you want to use with the VPN?

1) Current system resolvers (from /etc/resolv.conf)

2) Self-hosted DNS Resolver (Unbound)

3) Cloudflare (Anycast: worldwide)

4) Quad9 (Anycast: worldwide)

5) Quad9 uncensored (Anycast: worldwide)

6) FDN (France)

7) DNS.WATCH (Germany)

8) OpenDNS (Anycast: worldwide)

9) Google (Anycast: worldwide)

10) Yandex Basic (Russia)

11) AdGuard DNS (Russia)

12) Custom

DNS [1-12]: **9**

Do you want to use compression? It is not recommended since the VORACLE attack make use of it.

Enable compression? [y/n]: **n**

Do you want to customize encryption settings?

Unless you know what you're doing, you should stick with the default parameters provided by the script.
Note that whatever you choose, all the choices presented in the script are safe. (Unlike OpenVPN's defaults)
See https://github.com/angristan/openvpn-install#security-and-encryption to learn more.

Customize encryption settings? [y/n]: **y**

Choose which cipher you want to use for the data channel:
  1) AES-128-GCM (recommended)
  2) AES-192-GCM
  3) AES-256-GCM
  4) AES-128-CBC
  5) AES-192-CBC
  6) AES-256-CBC
Cipher [1-6]: **1**

Choose what kind of certificate you want to use:
  1) ECDSA (recommended)
  2) RSA
Certificate key type [1-2]: **1**

Choose which curve you want to use for the certificate's key:
  1) prime256v1 (recommended)
  2) secp384r1
  3) secp521r1
Curve [1-3]: **1**

Choose which cipher you want to use for the control channel:
  1) ECDHE-ECDSA-AES-128-GCM-SHA256 (recommended)
  2) ECDHE-ECDSA-AES-256-GCM-SHA384
Control channel cipher [1-2]: **1**

Choose what kind of Diffie-Hellman key you want to use:
  1) ECDH (recommended)
  2) DH

DH key type [1-2]: **1**

Choose which curve you want to use for the ECDH key:

1) prime256v1 (recommended)

2) secp384r1

3) secp521r1

Curve [1-3]: **1**

The digest algorithm authenticates tls-auth packets from the control channel.

Which digest algorithm do you want to use for HMAC?

1) SHA-256 (recommended)

2) SHA-384

3) SHA-512

Digest algorithm [1-3]: **1**

You can add an additional layer of security to the control channel with tls-auth and tls-crypt

tls-auth authenticates the packets, while tls-crypt authenticate and encrypt them.

1) tls-crypt (recommended)

2) tls-auth

Control channel additional security mechanism [1-2]: **1**

Okay, that was all I needed. We are ready to setup your OpenVPN server now.

You will be able to generate a client at the end of the installation.

Press any key to continue...

Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease

Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]

Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]

Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [753 kB]

Hit:5 http://ppa.launchpad.net/wireguard/wireguard/ubuntu bionic InRelease

Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]

Hit:7 https://artifacts.loraserver.io/packages/3.x/deb stable InRelease

Fetched 1005 kB in 2s (466 kB/s)

Reading package lists... Done

Reading package lists... Done

```
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.4-1ubuntu1.2).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
iptables is already the newest version (1.6.1-2ubuntu2).
iptables set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
openssl is already the newest version (1.1.1-1ubuntu2.1~18.04.4).
openssl set to manually installed.
wget is already the newest version (1.19.4-1ubuntu2.2).
wget set to manually installed.
The following additional packages will be installed:
  libpkcs11-helper1
Suggested packages:
  easy-rsa resolvconf
The following NEW packages will be installed:
  libpkcs11-helper1 openvpn
0 upgraded, 2 newly installed, 0 to remove and 14 not upgraded.
Need to get 514 kB of archives.
After this operation, 1274 kB of additional disk space will be used.
Get:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libpkcs11-helper1 amd64
1.22-4 [43.5 kB]
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openvpn amd64
2.4.4-2ubuntu1.3 [470 kB]
Fetched 514 kB in 1s (498 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libpkcs11-helper1:amd64.
```

(Reading database ... 72963 files and directories currently installed.)

Preparing to unpack .../libpkcs11-helper1_1.22-4_amd64.deb ...

Unpacking libpkcs11-helper1:amd64 (1.22-4) ...

Selecting previously unselected package openvpn.

Preparing to unpack .../openvpn_2.4.4-2ubuntu1.3_amd64.deb ...

Unpacking openvpn (2.4.4-2ubuntu1.3) ...

Setting up libpkcs11-helper1:amd64 (1.22-4) ...

Setting up openvpn (2.4.4-2ubuntu1.3) ...

* Restarting virtual private network daemon. [OK]

Created symlink /etc/systemd/system/multi-user.target.wants/openvpn.service → /lib/systemd/system/openvpn.service.

Processing triggers for systemd (237-3ubuntu10.29) ...

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

Processing triggers for ureadahead (0.100.0-21) ...

Processing triggers for libc-bin (2.27-3ubuntu1) ...

--2019-10-19 07:31:21-- https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz

Resolving github.com (github.com)... 52.78.231.108

Connecting to github.com (github.com)|52.78.231.108|:443... connected.

HTTP request sent, awaiting response... 302 Found

Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/4519663/8d46db80-266e-11e9-85e3-7de4dbee40d9?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20191019%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191019T073122Z&X-Amz-Expires=300&X-Amz-Signature=9605538897bee9a22479578293abf7e69a3c236745976abb708d1368c9511595&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DEasyRSA-unix-v3.0.6.tgz&response-content-type=application%2Foctet-stream [following]

--2019-10-19 07:31:22-- https://github-production-release-asset-2e65be.s3.amazonaws.com/4519663/8d46db80-266e-11e9-85e3-7de4dbee40d9?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20191019%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191019T073122Z&X-Amz-Expires=300&X-Amz-Signature=9605538897bee9a22479578293abf7e69a3c236745976abb708d1368c9511595&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DEasyRSA-unix-v3.0.6.tgz&response-content-type=application%2Foctet-stream

Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.216.137.108

Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216.137.108|:443... connected.

HTTP request sent, awaiting response... 200 OK

```
Length: 40840 (40K) [application/octet-stream]
Saving to: '/home/ubuntu/EasyRSA-unix-v3.0.6.tgz'

/home/ubuntu/EasyRSA-uni 100%[=================================>]  39.88K   215KB/s    in 0.2s

2019-10-19 07:31:23 (215 KB/s) - '/home/ubuntu/EasyRSA-unix-v3.0.6.tgz' saved [40840/40840]

sed: can't read pki/openssl-easyrsa.cnf: No such file or directory

Note: using Easy-RSA configuration from: ./vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/pki

read EC key
writing EC key
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
140373738652096:error:2406F079:random number generator:RAND_load_file:Cannot open
file:../crypto/rand/randfile.c:88:Filename=/etc/openvpn/easy-rsa/pki/.rnd

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
140150280020416:error:2406F079:random number generator:RAND_load_file:Cannot open
file:../crypto/rand/randfile.c:88:Filename=/etc/openvpn/easy-rsa/pki/.rnd
        <---------- 여기서 에러가 발생했는데 ....
Generating an EC private key
writing new private key to '/etc/openvpn/easy-rsa/pki/private/server_7naUyNU2YK8l8hg0.key.6vMFHwWww2'
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
140516378264000:error:2406F079:random number generator:RAND_load_file:Cannot open
file:../crypto/rand/randfile.c:88:Filename=/etc/openvpn/easy-rsa/pki/.rnd
```

```
Can't open /etc/openvpn/easy-rsa/pki/index.txt.attr for reading, No such file or directory
140516378264000:error:02001002:system library:fopen:No such file or directory:../crypto/bio/bss_file.c:72:fopen('/etc/openvpn/easy-rsa/pki/index.txt.attr','r')
140516378264000:error:2006D080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:79:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server_7naUyNU2YK8l8hg0'
Certificate is to be certified until Oct  3 07:31:23 2022 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
140394271482304:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/etc/openvpn/easy-rsa/pki/.rnd
            <---------- 여기서 에러가 발생했는데 ....

An updated CRL has been created.
CRL file: /etc/openvpn/easy-rsa/pki/crl.pem

* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
* Applying /etc/sysctl.d/10-link-restrictions.conf ...
fs.protected_hardlinks = 1
```

fs.protected_symlinks = 1

* Applying /etc/sysctl.d/10-lxd-inotify.conf ...

fs.inotify.max_user_instances = 1024

* Applying /etc/sysctl.d/10-magic-sysrq.conf ...

kernel.sysrq = 176

* Applying /etc/sysctl.d/10-network-security.conf ...

net.ipv4.conf.default.rp_filter = 1

net.ipv4.conf.all.rp_filter = 1

net.ipv4.tcp_syncookies = 1

* Applying /etc/sysctl.d/10-ptrace.conf ...

kernel.yama.ptrace_scope = 1

* Applying /etc/sysctl.d/10-zeropage.conf ...

vm.mmap_min_addr = 65536

* Applying /etc/sysctl.d/20-openvpn.conf ...

net.ipv4.ip_forward = 1

* Applying /etc/sysctl.d/30-postgresql-shm.conf ...

* Applying /usr/lib/sysctl.d/50-default.conf ...

net.ipv4.conf.all.promote_secondaries = 1

net.core.default_qdisc = fq_codel

* Applying /etc/sysctl.d/99-cloudimg-ipv6.conf ...

net.ipv6.conf.all.use_tempaddr = 0

net.ipv6.conf.default.use_tempaddr = 0

* Applying /etc/sysctl.d/99-sysctl.conf ...

* Applying /etc/sysctl.conf ...

Created symlink /etc/systemd/system/multi-user.target.wants/openvpn@server.service → /etc/systemd/system/openvpn@.service.

Created symlink /etc/systemd/system/multi-user.target.wants/iptables-openvpn.service → /etc/systemd/system/iptables-openvpn.service.

Tell me a name for the client.

Use one word only, no special characters.

Client name: **michael**

Do you want to protect the configuration file with a password?

```
(e.g. encrypt the private key with a password)
   1) Add a passwordless client
   2) Use a password for the client
Select an option [1-2]: 1

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
139846297821632:error:2406F079:random number generator:RAND_load_file:Cannot open
file:../crypto/rand/randfile.c:88:Filename=/etc/openvpn/easy-rsa/pki/.rnd
            <---------- 여기서 에러가 발생했는데 ....
Generating an EC private key
writing new private key to '/etc/openvpn/easy-rsa/pki/private/michael.key.6Rdc1OAclZ'
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Can't load /etc/openvpn/easy-rsa/pki/.rnd into RNG
139794386883008:error:2406F079:random number generator:RAND_load_file:Cannot open
file:../crypto/rand/randfile.c:88:Filename=/etc/openvpn/easy-rsa/pki/.rnd
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'michael'
Certificate is to be certified until Oct  3 07:32:33 2022 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Client michael added, the configuration file is available at /home/ubuntu/michael.ovpn.
Download the .ovpn file and import it in your OpenVPN client.
```

$ **ifconfig -a**

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
```

```
        inet 172.31.17.27  netmask 255.255.240.0  broadcast 172.31.31.255
        inet6 fe80::868:2bff:fe00:47a8  prefixlen 64  scopeid 0x20<link>
        ether 0a:68:2b:00:47:a8  txqueuelen 1000  (Ethernet)
        RX packets 682213  bytes 142573612 (142.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 563540  bytes 157978869 (157.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12322822  bytes 2833636508 (2.8 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12322822  bytes 2833636508 (2.8 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

n2n0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1400
        inet 172.16.1.100  netmask 255.255.255.0  broadcast 172.16.1.255
        inet6 fe80::3ea0:12ff:fe34:5678  prefixlen 64  scopeid 0x20<link>
        ether 3c:a0:12:34:56:78  txqueuelen 1000  (Ethernet)
        RX packets 7740  bytes 735728 (735.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8325  bytes 763558 (763.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

spn0: flags=144<POINTOPOINT,NOARP>  mtu 1420
        inet 10.1.2.254  netmask 255.255.255.0  destination 10.1.2.254
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 160  bytes 7228 (7.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
        TX packets 44  bytes 4508 (4.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

spn1: flags=144<POINTOPOINT,NOARP>  mtu 1420
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.0  destination 10.8.0.1
        inet6 fe80::6a59:46af:113f:1911  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 384 (384.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

$ **netstat -nr**

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         172.31.16.1     0.0.0.0         UG        0 0          0 eth0
10.8.0.0        0.0.0.0         255.255.255.0   U         0 0          0 tun0
172.16.1.0      0.0.0.0         255.255.255.0   U         0 0          0 n2n0
172.31.16.0     0.0.0.0         255.255.240.0   U         0 0          0 eth0
172.31.16.1     0.0.0.0         255.255.255.255 UH        0 0          0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U         0 0          0 tun1
```

참고: tun1은 tinc(mesh vpn)를 테스트하는 과정에서 생성된 것으로 openvpn과는 무관하다.

$ **ps aux|grep openvpn**

```
nobody   16307  0.0  0.6  44308  7012 ?        Ss   07:31   0:00 /usr/sbin/openvpn
--daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn
--script-security 2 --config /etc/openvpn/server.conf --writepid
/run/openvpn/server.pid
ubuntu   17567  0.0  0.1  14856  1112 pts/3    S+   07:46   0:00 grep --color=auto openvpn
```

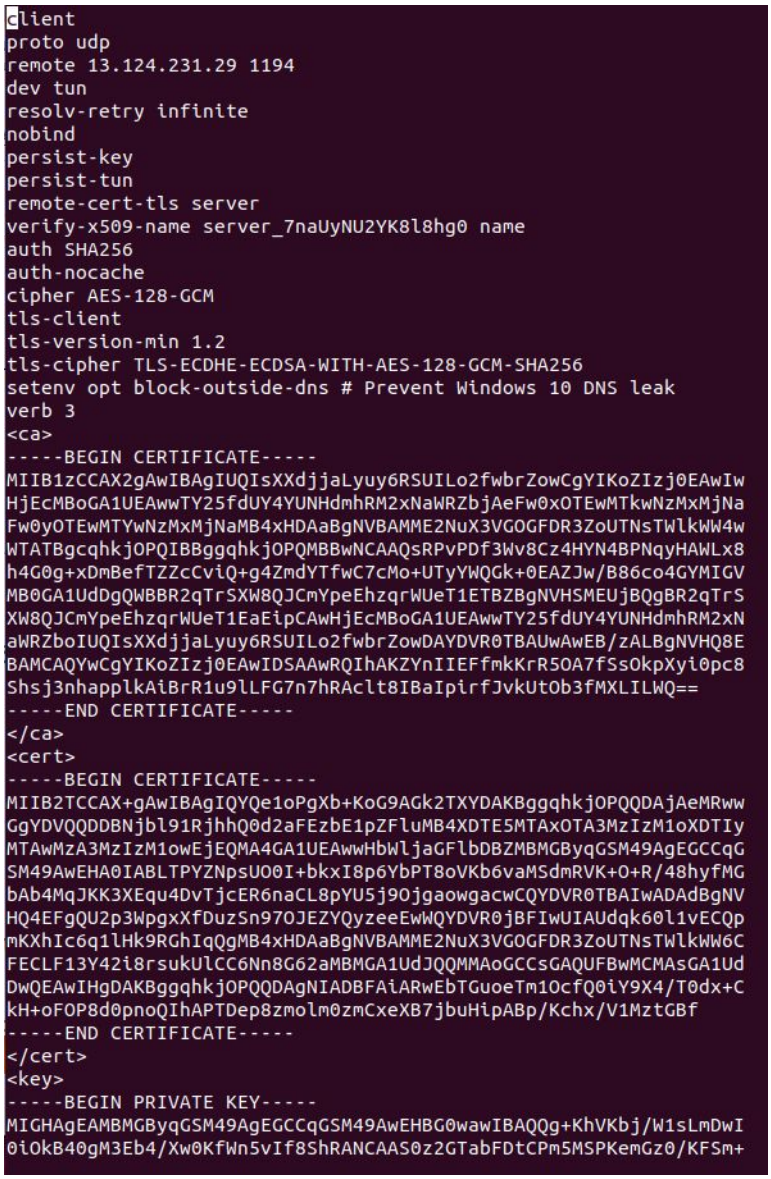
일단, 에러가 하나 보이기는 하지만, openvpn server는 정상적으로 구동된 느낌이다.

$ **sudo ./openvpn-install.sh**

➔ 이 shell script를 다시 실행하게 되면, vpn client를 추가 혹은 삭제(revoke)할 수 있으며, OpenVPN server 자체를 통째로 제거할 수도 있다.

```
Welcome to OpenVPN-install!
The git repository is available at: https://github.com/angristan/openvpn-install

It looks like OpenVPN is already installed.

What do you want to do?
   1) Add a new user
   2) Revoke existing user
   3) Remove OpenVPN
   4) Exit
Select an option [1-4]:
```

OpenVPN 설치 과정 끝 부분에서 추가한 VPN 사용자(예: michael)의 .ovpn 파일(OpenVPN client에서 서버에 접속할 때 사용하는 정보) 내용을 살펴 보면 다음과 같다(내용이 길어서 앞 부분만 capture하였음).



```
client
proto udp
remote 13.124.231.29 1194
dev tun
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
verify-x509-name server_7naUyNU2YK8l8hg0 name
auth SHA256
auth-nocache
cipher AES-128-GCM
tls-client
tls-version-min 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256
setenv opt block-outside-dns # Prevent Windows 10 DNS leak
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIIB1zCCAX2gAwIBAgIUQIsXXdjjaLyuy6RSUILo2fwbrZowCgYIKoZIzj0EAwIw
HjEcMBoGA1UEAwwTY25fdUY4YUNHdmhRM2xNaWRZbjAeFw0xOTEwMTkwNzMxMjNa
Fw0yOTEwMTYwNzMxMjNaMB4xHDAaBgNVBAMME2NuX3VGOGFDR3ZoUTNsTWlkWW4w
WTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAAQsRPvPDf3Wv8Cz4HYN4BPNqyHAWLx8
h4G0g+xDmBefTZZcCviQ+g4ZmdYTfwC7cMo+UTyYWQGk+0EAZJw/B86co4GYMIGV
MB0GA1UdDgQWBBR2qTrSXW8QJCmYpeEhzqrWUeT1ETBZBgNVHSMEUjBQgBR2qTrS
XW8QJCmYpeEhzqrWUeT1EaEipCAwHjEcMBoGA1UEAwwTY25fdUY4YUNHdmhRM2xN
aWRZboIUQIsXXdjjaLyuy6RSUILo2fwbrZowDAYDVR0TBAUwAwEB/zALBgNVHQ8E
BAMCAQYwCgYIKoZIzj0EAwIDSAAwRQIhAKZYnIIEFfmkKrR5OA7fSsOkpXyi0pc8
Shsj3nhapplkAiBrR1u9lLFG7n7hRAclt8IBaIpirfJvkUtOb3fMXLILWQ==
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
MIIB2TCCAX+gAwIBAgIQYQe1oPgXb+KoG9AGk2TXYDAKBggqhkjOPQQDAjAeMRww
GgYDVQQDDBNjbl91RjhhQ0d2aFEzbE1pZFluMB4XDTE5MTAxOTA3MzIzM1oXDTIy
MTAwMzA3MzIzM1owEjEQMA4GA1UEAwwHbWljaGFlbDBZMBMGByqGSM49AgEGCCqG
SM49AwEHA0IABLTPYZNpsUO0I+bkxI8p6YbPT8oVKb6vaMSdmRVK+O+R/48hyfMG
bAb4MqJKK3XEqu4DvTjcER6naCL8pYU5j9OjgaowgacwCQYDVR0TBAIwADAdBgNV
HQ4EFgQU2p3WpgxXfDuzSn97OJEZYQyzeeEwWQYDVR0jBFIwUIAUdqk60l1vECQp
mKXhIc6q1lHk9RGhIqQgMB4xHDAaBgNVBAMME2NuX3VGOGFDR3ZoUTNsTWlkWW6C
FECLF13Y42i8rsukUlCC6Nn8G62aMBMGA1UdJQQMMAoGCCsGAQUFBwMCMAsGA1Ud
DwQEAwIHgDAKBggqhkjOPQQDAgNIADBFAiARwEbTGuoeTm1OcfQ0iY9X4/T0dx+C
kH+oFOP8d0pnoQIhAPTDep8zmolm0zmCxeXB7jbuHipABp/Kchx/V1MztGBf
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQg+KhVKbj/W1sLmDwI
0iOkB40gM3Eb4/Xw0KfWn5vIf8ShRANCAAS0z2GTabFDtCPm5MSPKemGz0/KFSm+
```

[그림 19.6] michael.ovpn 파일 내용 확인

<참고 사항 - ovpn 내용 중 알아야 할 부분>

<ca> ~ </ca> : CA(공인 인증 기관)가 발생한 공인 인증서
<cert> ~ </cert> : 이 경우는 client의 public key를 포함한 공인 인증서
<key> ~ </key> : client의 private key
<tls-auth> ~ </tls-auth> : TLS-Auth 용으로 사용되는 preshared-key

주의: OpenVPN은 TLS 1.x를 기반으로 하는데, TLS 1.x에서는 서버와 클라이언트는 자신의 공개키(정확히는 인증서)를 network을 통해 교환하는 방식을 따른다.

이 파일은 OpenVPN client에서 server와의 TLS 연결을 위해 사용된다. Wireguard의 경우를 생각해 보라. Peer의 public key말고 필요한게 없지 않았는가 ?. 그에 비하면 OpenVPN은 매우 많은 설정 정보를 요구함을 알 수 있다.

3) OpenVPN Client 설치하기

이 절에서는 OpenVPN Client를 설치하고 서버와 연동하는 방법을 소개해 보기로 하겠다.

a) Linux(Ubuntu) OpenVPN Client 설치하기

$ **sudo apt update**

$ **sudo apt install openvpn**

이 시점에서 openvpn server로 부터 michael.ovpn 파일을 download 받는다.

$ **sudo mv michael.ovpn /etc/openvpn/**

➔ michael.ovpn 파일을 /etc/openvpn으로 복사한다(꼭 이래야 하는 것은 아님).

$ **sudo openvpn --config /etc/openvpn/michael.ovpn**

➔ OpenVPN client를 실행한다.

```
Sat Oct 19 16:57:39 2019 Unrecognized option or missing or extra parameter(s) in /etc/openvpn/michael.ovpn:17: block-outside-dns (2.4.4)

Sat Oct 19 16:57:39 2019 OpenVPN 2.4.4 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on May 14 2019

Sat Oct 19 16:57:39 2019 library versions: OpenSSL 1.1.1  11 Sep 2018, LZO 2.08

Sat Oct 19 16:57:39 2019 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key

Sat Oct 19 16:57:39 2019 Outgoing Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication

Sat Oct 19 16:57:39 2019 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key

Sat Oct 19 16:57:39 2019 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication

Sat Oct 19 16:57:39 2019 TCP/UDP: Preserving recently used remote address: [AF_INET]13.124.231.29:1194

Sat Oct 19 16:57:39 2019 Socket Buffers: R=[212992->212992] S=[212992->212992]

Sat Oct 19 16:57:39 2019 UDP link local: (not bound)
```

Sat Oct 19 16:57:39 2019 UDP link remote: [AF_INET]13.124.231.29:1194

Sat Oct 19 16:57:39 2019 TLS: Initial packet from [AF_INET]13.124.231.29:1194, sid=792a021c 5b2d3809

Sat Oct 19 16:57:39 2019 VERIFY OK: depth=1, CN=cn_uF8aCGvhQ3lMidYn

Sat Oct 19 16:57:39 2019 VERIFY KU OK

Sat Oct 19 16:57:39 2019 Validating certificate extended key usage

Sat Oct 19 16:57:39 2019 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication

Sat Oct 19 16:57:39 2019 VERIFY EKU OK

Sat Oct 19 16:57:39 2019 VERIFY X509NAME OK: CN=server_7naUyNU2YK8l8hg0

Sat Oct 19 16:57:39 2019 VERIFY OK: depth=0, CN=server_7naUyNU2YK8l8hg0

Sat Oct 19 16:57:39 2019 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 256 bit EC, curve: prime256v1

Sat Oct 19 16:57:39 2019 [server_7naUyNU2YK8l8hg0] Peer Connection Initiated with [AF_INET]13.124.231.29:1194

Sat Oct 19 16:57:40 2019 SENT CONTROL [server_7naUyNU2YK8l8hg0]: 'PUSH_REQUEST' (status=1)

Sat Oct 19 16:57:40 2019 PUSH: Received control message: 'PUSH_REPLY,dhcp-option DNS 8.8.8.8,dhcp-option DNS 8.8.4.4,redirect-gateway def1 bypass-dhcp,route-gateway 10.8.0.1,topology subnet,ping 10,ping-restart 120,ifconfig 10.8.0.2 255.255.255.0,peer-id 0,cipher AES-128-GCM'

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: timers and/or timeouts modified

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: --ifconfig/up options modified

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: route options modified

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: route-related options modified

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: peer-id set

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: adjusting link_mtu to 1624

Sat Oct 19 16:57:40 2019 OPTIONS IMPORT: data channel crypto options modified

Sat Oct 19 16:57:40 2019 Outgoing Data Channel: Cipher 'AES-128-GCM' initialized with 128 bit key

Sat Oct 19 16:57:40 2019 Incoming Data Channel: Cipher 'AES-128-GCM' initialized with 128 bit key

Sat Oct 19 16:57:40 2019 ROUTE_GATEWAY 172.30.1.254/255.255.255.0 IFACE=wlp1s0 HWADDR=f8:a2:d6:80:91:39

Sat Oct 19 16:57:40 2019 TUN/TAP device tun0 opened

Sat Oct 19 16:57:40 2019 TUN/TAP TX queue length set to 100

Sat Oct 19 16:57:40 2019 do_ifconfig, tt->did_ifconfig_ipv6_setup=0

Sat Oct 19 16:57:40 2019 /sbin/ip link set dev tun0 up mtu 1500

Sat Oct 19 16:57:40 2019 /sbin/ip addr add dev tun0 10.8.0.2/24 broadcast 10.8.0.255

```
Sat Oct 19 16:57:40 2019 /sbin/ip route add 13.124.231.29/32 via 172.30.1.254
Sat Oct 19 16:57:40 2019 /sbin/ip route add 0.0.0.0/1 via 10.8.0.1
Sat Oct 19 16:57:40 2019 /sbin/ip route add 128.0.0.0/1 via 10.8.0.1
Sat Oct 19 16:57:40 2019 Initialization Sequence Completed
```

VPN client가 서버에 정상적으로 붙은 것 같다.

b) 서버를 통해 인터넷 사용하기 or 서버에 직접 연결하기

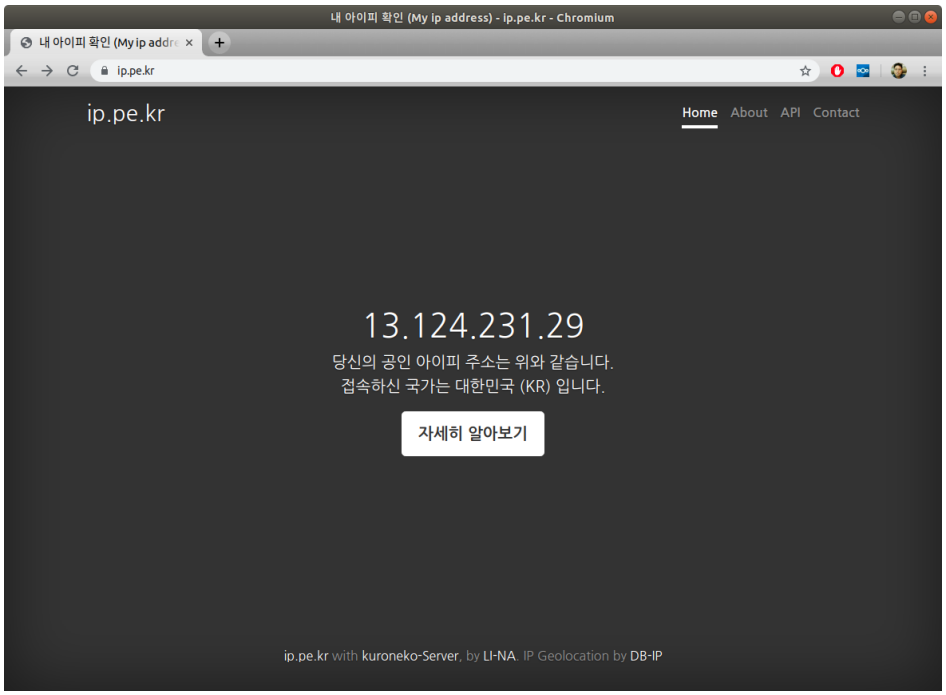
$ **ifconfig -a**

```
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 98:83:89:9a:9b:c6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 127449  bytes 19418429 (19.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 127449  bytes 19418429 (19.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.2  netmask 255.255.255.0  destination 10.8.0.2
        inet6 fe80::8e46:e7bf:700:dc54  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 303  bytes 66520 (66.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
        TX packets 375  bytes 47155 (47.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.1.14  netmask 255.255.255.0  broadcast 172.30.1.255
        inet6 fe80::e921:c0c7:3b45:2231  prefixlen 64  scopeid 0x20<link>
        ether f8:a2:d6:80:91:39  txqueuelen 1000  (Ethernet)
        RX packets 1627094  bytes 2360521704 (2.3 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 222616  bytes 32912262 (32.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

$ **netstat -nr**

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.8.0.1        128.0.0.0       UG        0 0          0 tun0
0.0.0.0         172.30.1.254    0.0.0.0         UG        0 0          0 wlp1s0
10.8.0.0        0.0.0.0         255.255.255.0   U         0 0          0 tun0
13.124.231.29   172.30.1.254    255.255.255.255 UGH       0 0          0 wlp1s0
128.0.0.0       10.8.0.1        128.0.0.0       UG        0 0          0 tun0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 wlp1s0
172.30.1.0      0.0.0.0         255.255.255.0   U         0 0          0 wlp1s0
```



[그림 19.7] OpenVPN이 설치된 client의 외부에서 바라본 ip

$ ping www.google.co.kr

```
PING www.google.co.kr (172.217.31.131) 56(84) bytes of data.
64 bytes from nrt20s08-in-f3.1e100.net (172.217.31.131): icmp_seq=1 ttl=44 time=40.4 ms
64 bytes from nrt20s08-in-f3.1e100.net (172.217.31.131): icmp_seq=2 ttl=44 time=45.0 ms
64 bytes from nrt20s08-in-f3.1e100.net (172.217.31.131): icmp_seq=3 ttl=44 time=48.6 ms
64 bytes from nrt20s08-in-f3.1e100.net (172.217.31.131): icmp_seq=4 ttl=44 time=41.4 ms
^C
--- www.google.co.kr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 40.407/43.883/48.633/3.253 ms
```

위의 ping www.google.co.kr은 예상했겠지만, 13.124.231.29 서버(OpenVPN Server)를 통해서 www.google.co.kr과 통신하는 것이다(VPN 우회 서비스 개념).

$ ping 10.8.0.1

```
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=10.4 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=9.19 ms
```

```
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=8.96 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=13.3 ms
64 bytes from 10.8.0.1: icmp_seq=5 ttl=64 time=12.6 ms
^C
--- 10.8.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 8.961/10.930/13.398/1.812 ms
```

$ **ssh -v -i 2ip-loraserver.pem ubuntu@10.8.0.1**

➔ OK

c) 다른 OS에서 OpenVPN Client 사용하기

<TBD> Windows, macOS, Android, iOS

20. MX1702 OpenVPN Client 사용하기

이 절에서는 MatchX MX1702와 LoRaServer를 **OpenVPN을 통해 연동**하는 방법을 소개하고자 한다.

<전체 작업 개요>

a) krx.matchx.io에서 gateway config 조정 : LoRa Server 변경

```
"gateway_conf": {
        /* change with default server address/ports, or overwrite in local_conf.json */
        "server_address": "10.8.0.1",    /* "krx.matchx.io", */
                                    =>  OpenVPN server ip로 교체해야 한다.
        "serv_port_up": 1700,
        "serv_port_down": 1700,
        /* adjust the following parameters for your network */
        "keepalive_interval": 10,
        "stat_interval": 30,
        "push_timeout_ms": 100,
        /* forward only valid packets */
        "forward_crc_valid": true,
        "forward_crc_error": false,
        "forward_crc_disabled": false,
```

```
                /* GPS configuration */
                "gps_tty_path": "/dev/ttyS1",
                /* GPS reference coordinates */
                "ref_latitude": 0.0,
                "ref_longitude": 0.0,
                "ref_altitude": 0
        }
```

b) /etc/openvpn/spnbox/openvpn-matchx_client.conf
- ➔ 여기에 loraserver와의 연동 client conf 유지(이 파일을 직접 사용함)
- ➔ 이를 위해 /etc/init.d/openvpn, /etc/config/openvpn 수정함.

c) cli "openvpn start client" 는 기존에 동작하던 openvpn kill하고, 새로 구동 !
- ➔ /etc/init.d/openvpn start|stop 이용

d) 이후, LoRaServer에서 lora packet 확인 ~

1) OpenVPN Client

<구동 중인 MX1702 openvpn client>

```
/usr/sbin/openvpn --syslog openvpn(matchx_client) --status
/var/run/openvpn.matchx_client.status --cd /var/etc --config
openvpn-matchx_client.conf
```

```
client
nobind
persist-key
persist-tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/client.crt
comp-lzo yes
dev tun
key /etc/openvpn/client.key
proto udp
remote krx.matchx.io 1194
resolv-retry infinite
user nobody
verb 3
~
```

[그림 20.1] openvpn client config 파일 - /var/etc/openvpn-matchx_client.conf

</var/etc/openvpn-matchx_client.conf 파일 수정하기>

```
client
proto udp
remote 13.124.231.29 1194
dev tun
nobind
ca /etc/openvpn/michael/ca.crt
cert /etc/openvpn/michael/client.crt
key /etc/openvpn/michael/client.key
```

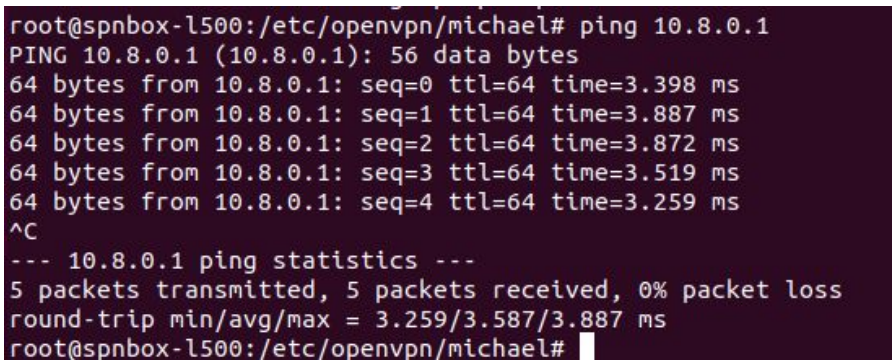
참고: /etc/openvpn/michael folder에 OpenVPN server에서 생성해 준, ca.crt, client.crt, client.key 파일을 미리 복사해 두어야 한다.

<openvpn client 재 구동하기>

 # killall -9 openvpn

 # /usr/sbin/openvpn --syslog "openvpn(matchx_client)" --status /var/run/openvpn.matchx_client.status --cd /var/etc --config openvpn-matchx_client.conf

 # ping 10.8.0.1



[그림 20.2] OpenVPN Server로의 ping OK

2) /etc/init.d/openvpn 수정하기

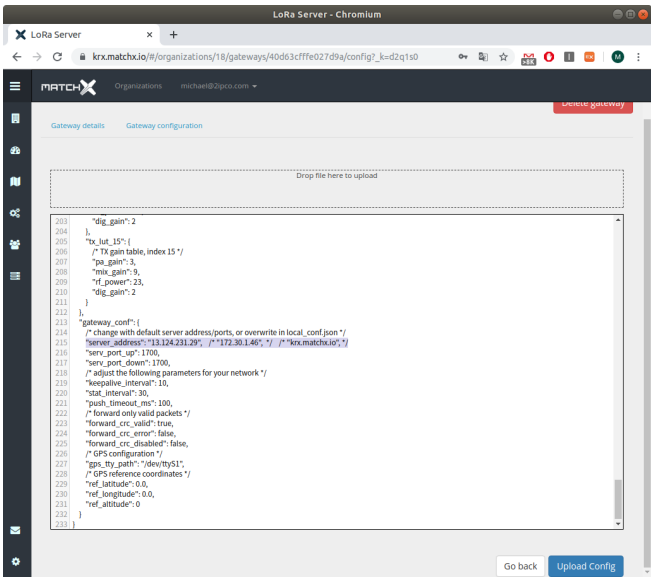
현재 mx1702는 부팅 시, /etc/init.d/openvpn 파일이 실행되면서, /var/etc/openvpn-matchx_client.conf 파일이 원래 것으로 복원되는 구조로 되어 있다. 따라서 spnboot에서는 이를 LoRaServer OpenVPN server와 연동하도록 변경시켜 줄 필요가 있다.

</usr/bin/spnboot>

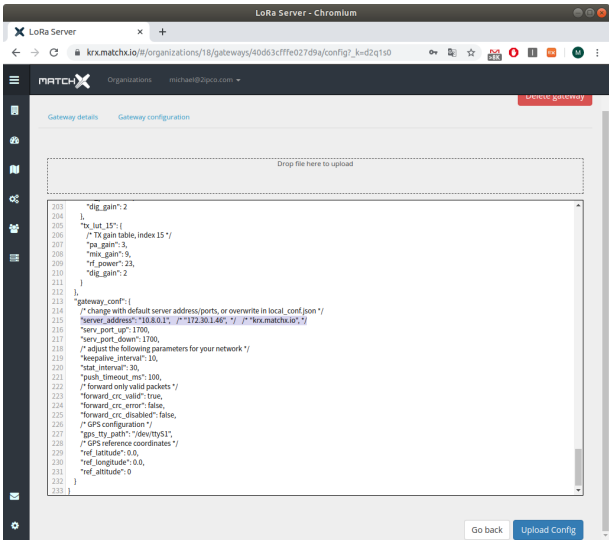
```
cp /etc/openvpn/spnbox/openvpn-matchx_client.conf /var/etc
```

3) LoRaServer와 연동하기

https://krx.matchx.io/에서 LoRa Gateway config 내용 중, server address를 OpenVPN server ip address로 미리 변경시켜 두어야 한다.



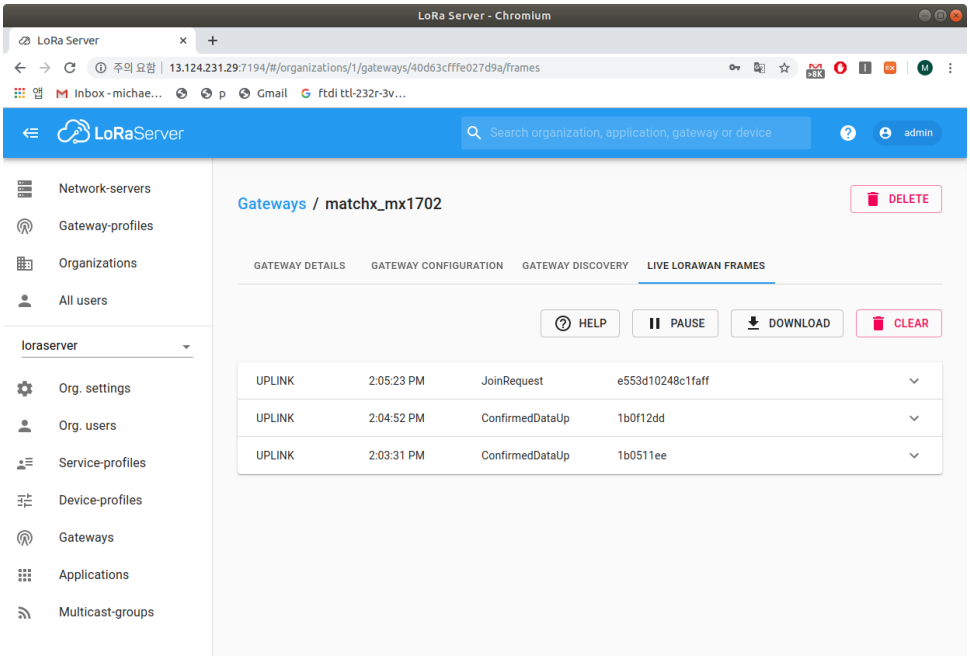
[그림 20.3] krx.matchx.io - Gateway configuration 수정 - LoRaServer real ip



[그림 20.4] krx.matchx.io - Gateway configuration 수정 - LoRaServer OpenVPN IP

위의 gateway configuration 용 실제 파일은 /root/lora_pkt_fwd/global_conf.json 파일이다. 근데, global_conf.json.kr 파일도 수정해 주어야 하나 ? 맞다. 이 파일을 바꿔 주어야만 정상 동작한다.

OK, 정상 동작한다.



[그림 20.5] MX1702 ⇔ OpenVPN ⇔ LoRaServer 정상 연동 모습

21. UrsaLink LoRa Gateway ⇔ LoRaServer 연동하기(OpenVPN 기반)

이 절에서는 UrsaLink LoRa Gateway(UG85)와 LoRaServer를 연동하는 방법을 소개하고자 한다.

1) OpenVPN Point-to-Point 연결 설정하기

(우리에게는 익숙한) 1 대 1 VPN 연결을 하고자 할 때 OpenVPN을 어찌 설정하는 지를 설명하고자 한다.

<OpenVPN Server 설정하기>

$ **openvpn --genkey --secret secret.key**

➔ Preshared-key로 활용할 2048 bit key를 생성한다.

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
c77888dae32dc8acd5556345015a8dd6
a1aab017bb083d929959cf867ccdf4e5
16935e36100a700f92b628b9c31ce9c3
bc4a5fe9a9222961723c6f700099586a
27c5b944434d1a4bebe385a224996898
5b0cc3b4889fc3fdf539939273ed3e60
1477e9744847d65521c168db37c6d74c
3ad344a743d4daa8923044c5a2520786
f26df818cf2d7cc6a69c0ef2348ad2c0
62b2264c0d898a522489e47542b277ee
416ad3cd91bdddd8f2c871b6dfc3f01b
206e9f790e6b597c159e2bf5dc0046ef
f698d22ec3dc7adbddfb48cb52c61d87
b6651e6f4a2e0aa8b2663e8c281934a2
acbaa10059ed479831c43af63a7198dd
dc302b1c7a1e2793473cf41fa6816aa7
-----END OpenVPN Static key V1-----
~
```

[그림 21.1] OpenVPN secret key 생성 예

$ **sudo openvpn --ifconfig 10.8.0.1 10.8.0.100 --dev tun --secret secret.key --verb 7**

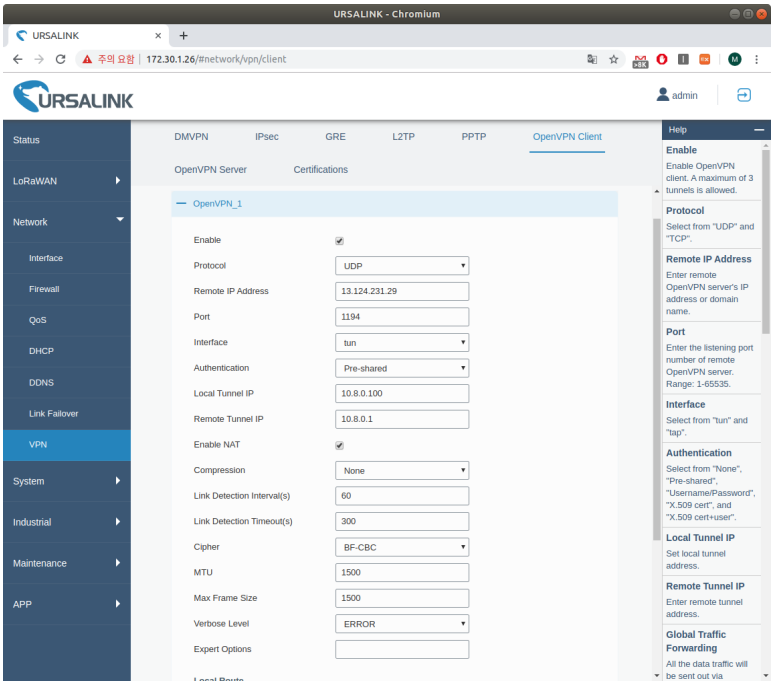
참고: 이렇게 설명할 경우, OpenVPN client는 Cipher 알고리즘으로 BF-CBC(blowfish CBC)를 선택해 주어야 한다.

<AES-256-CBC 예>

$ **sudo openvpn --ifconfig 10.8.0.1 10.8.0.100 --dev tun --secret secret.key --cipher AES256 --verb 7**

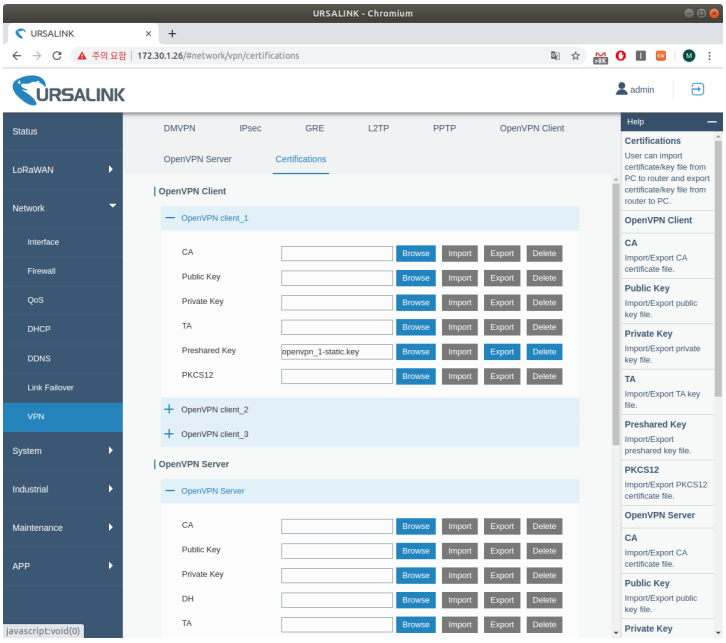
<OpenVPN Client 설정하기>

UG85 LoRa Gateway의 OpenVPN client 설정을 해 보도록 하겠다.

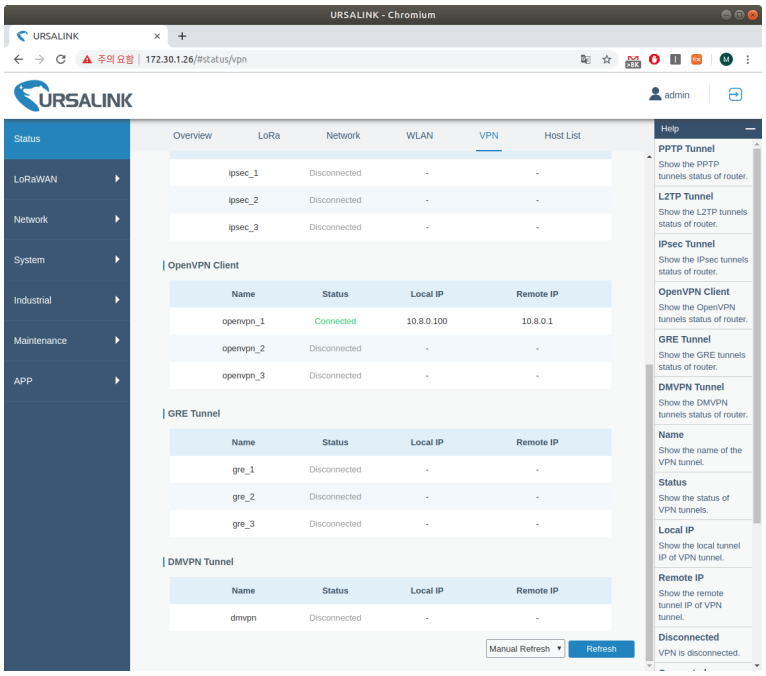


[그림 21.2] UrsaLink UG85 WebUI OpenVPN Client 설정(1)

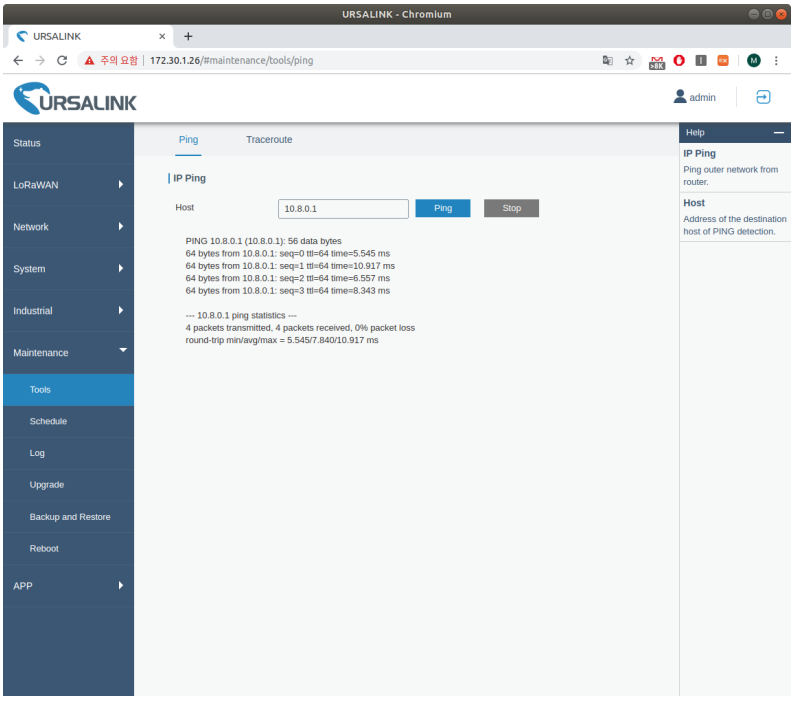
참고: UrsaLink WebUI id/pass ⇒ admin/spnbox! 이다.



[그림 21.3] UrsaLink UG85 WebUI OpenVPN Client 설정(2) - secret.key 파일 Preshared key로 등록



[그림 21.4] UrsaLink UG85 WebUI OpenVPN Status



[그림 21.5] UrsaLink UG85 WebUI - ping test

2) OpenVPN Client/Server 연결 설정하기

1개의 서버와 여러개의 Client를 VPN으로 연결하는 방식을 소개해 보도록 하겠다. 이번에는 앞서의 경우와는 달리 인증서를 사용하도록 한다.

<OpenVPN Server 설정하기>

$ **sudo ./openvpn-install.sh**

➔ 기존 OpenVPN server를 제거한다.

```
Welcome to OpenVPN-install!
The git repository is available at: https://github.com/angristan/openvpn-install

It looks like OpenVPN is already installed.

What do you want to do?
   1) Add a new user
   2) Revoke existing user
   3) Remove OpenVPN
   4) Exit
Select an option [1-4]: 3
```

$ **sudo ./openvpn-install.sh**

➔ 최대한 UrsaLink UG85에 맞게 재 설정하자.

```
Welcome to the OpenVPN installer!
The git repository is available at: https://github.com/angristan/openvpn-install

I need to ask you a few questions before starting the setup.
You can leave the default options and just press enter if you are ok with them.

I need to know the IPv4 address of the network interface you want OpenVPN listening to.
Unless your server is behind NAT, it should be your public IPv4 address.
IP address: 13.124.231.29

Checking for IPv6 connectivity...
```

Your host does not appear to have IPv6 connectivity.

Do you want to enable IPv6 support (NAT)? [y/n]: n

What port do you want OpenVPN to listen to?

1) Default: 1194

2) Custom

3) Random [49152-65535]

Port choice [1-3]: 1

What protocol do you want OpenVPN to use?

UDP is faster. Unless it is not available, you shouldn't use TCP.

1) UDP

2) TCP

Protocol [1-2]: 1

What DNS resolvers do you want to use with the VPN?

1) Current system resolvers (from /etc/resolv.conf)

2) Self-hosted DNS Resolver (Unbound)

3) Cloudflare (Anycast: worldwide)

4) Quad9 (Anycast: worldwide)

5) Quad9 uncensored (Anycast: worldwide)

6) FDN (France)

7) DNS.WATCH (Germany)

8) OpenDNS (Anycast: worldwide)

9) Google (Anycast: worldwide)

10) Yandex Basic (Russia)

11) AdGuard DNS (Russia)

12) Custom

DNS [1-12]: 39

DNS [1-12]: 9

Do you want to use compression? It is not recommended since the VORACLE attack make use of it.

Enable compression? [y/n]: n

Do you want to customize encryption settings?

Unless you know what you're doing, you should stick with the default parameters provided by the script.

Note that whatever you choose, all the choices presented in the script are safe. (Unlike OpenVPN's defaults)

See https://github.com/angristan/openvpn-install#security-and-encryption to learn more.

Customize encryption settings? [y/n]: y

Choose which cipher you want to use for the data channel:

1) AES-128-GCM (recommended)

2) AES-192-GCM

3) AES-256-GCM

4) AES-128-CBC

5) AES-192-CBC

6) AES-256-CBC

Cipher [1-6]: 6

Choose what kind of certificate you want to use:

1) ECDSA (recommended)

2) RSA

Certificate key type [1-2]: 1

Choose which curve you want to use for the certificate's key:

1) prime256v1 (recommended)

2) secp384r1

3) secp521r1

Curve [1-3]: 1

Choose which cipher you want to use for the control channel:

1) ECDHE-ECDSA-AES-128-GCM-SHA256 (recommended)

2) ECDHE-ECDSA-AES-256-GCM-SHA384

Control channel cipher [1-2]: 1

Choose what kind of Diffie-Hellman key you want to use:

1) ECDH (recommended)

2) DH

DH key type [1-2]: 1

Choose which curve you want to use for the ECDH key:

1) prime256v1 (recommended)

2) secp384r1

3) secp521r1

Curve [1-3]: 1

The digest algorithm authenticates data channel packets and tls-auth packets from the control channel.

Which digest algorithm do you want to use for HMAC?

1) SHA-256 (recommended)

2) SHA-384

3) SHA-512

Digest algorithm [1-3]: 1

You can add an additional layer of security to the control channel with tls-auth and tls-crypt

tls-auth authenticates the packets, while tls-crypt authenticate and encrypt them.

1) tls-crypt (recommended)

2) tls-auth

Control channel additional security mechanism [1-2]: 2

$ **sudo vi /etc/openvpn/test.conf**

➔ 이 파일을 하나 만들어, 최대한 기본적인 내용만 추가하자.

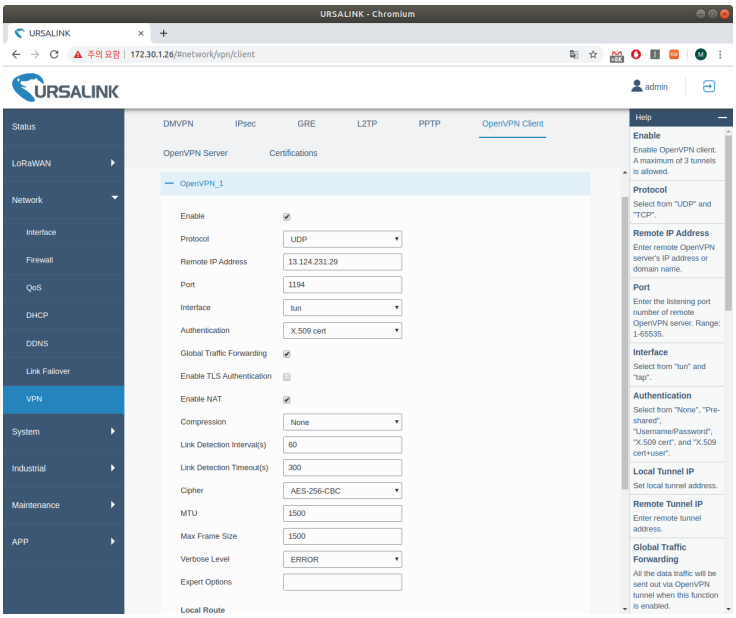
```
port 1194
proto udp
dev tun
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
push "redirect-gateway def1 bypass-dhcp"
dh none
ca ca.crt
cert server_rNPD6Y6ykVOt72EM.crt
key server_rNPD6Y6ykVOt72EM.key
status /var/log/openvpn/status.log
verb 3
```

[그림 21.6] OpenVPN server config 파일 - /etc/openvpn/test.conf

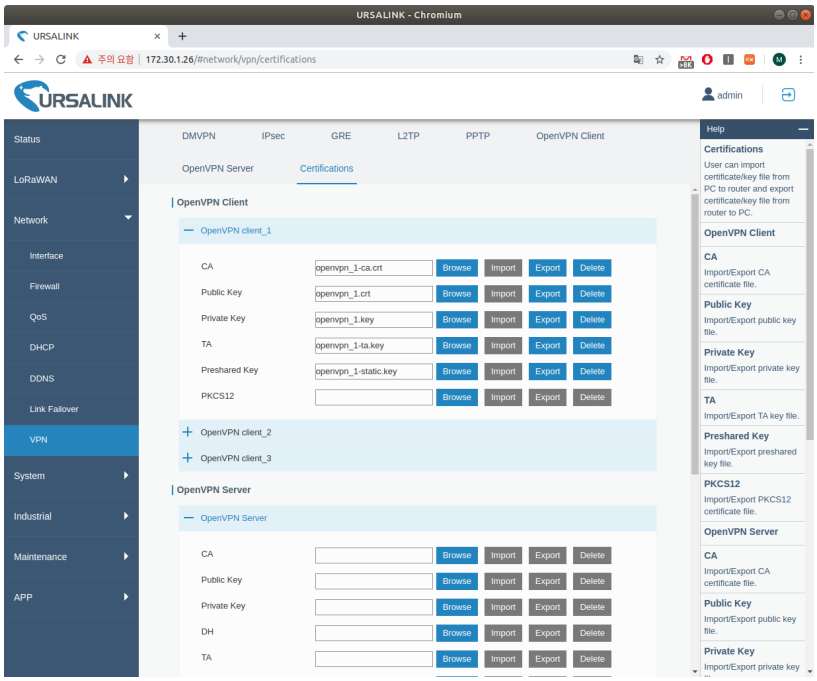
$ sudo openvpn --cd /etc/openvpn --script-security 2 --config /etc/openvpn/test.conf --verb 7

<OpenVPN Client 설정하기>

UG85 LoRa Gateway의 OpenVPN client 설정을 해 보도록 하겠다.



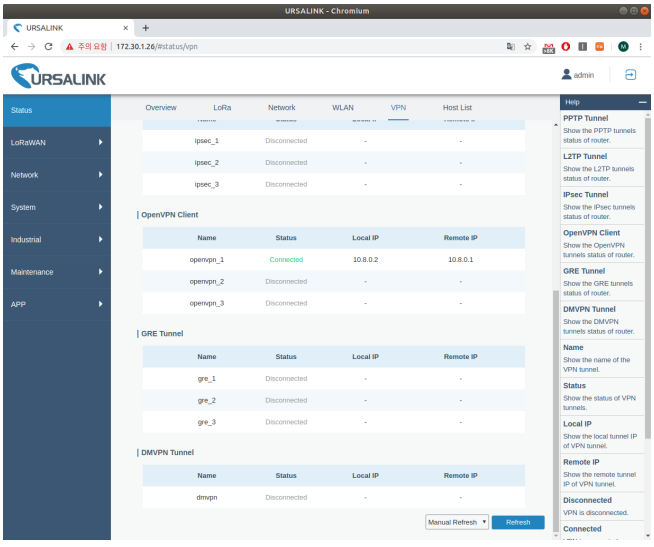
[그림 21.7] UrsaLink UG85 WebUI OpenVPN Client 설정 - 인증서 기반(1)



[그림 21.8] UrsaLink UG85 WebUI OpenVPN Client 설정 - 인증서 기반(2)

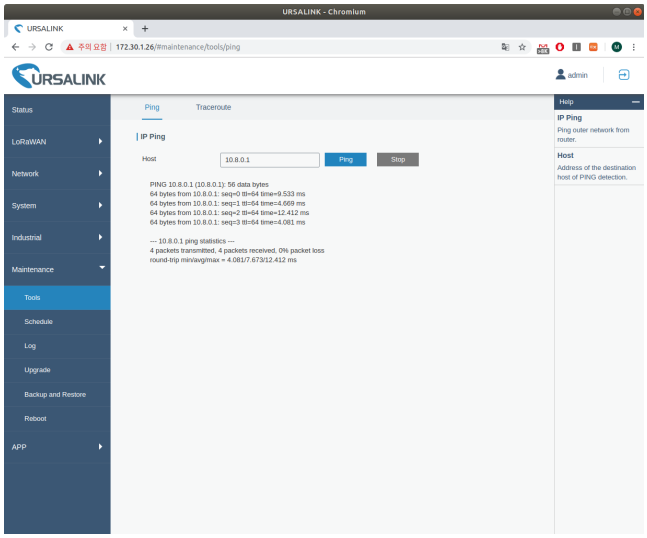
주의1: 위의 그림 처럼 파일을 올리기 위해서는 확장자를 정확히 해 주어야 한다. 즉, 인증서 파일은 crt, key file은 key라고 해 주어야 한다.

주의2: ca.crt, public.crt, private.key, static.key 등은 USERNAME.ovpn 파일에서 추출하거나, 서버의 /etc/openvpn/* 아래의 파일을 통해 만들 수 있다.



[그림 21.9] UrsaLink UG85 WebUI OpenVPN 상태 보기

참고: 이번에는 서버에서 정해준 ip 즉, 10.8.0.2를 할당 받았다.



[그림 21.10] UrsaLink UG85 WebUI OpenVPN ping 테스트

3) TLS-Auth 설정하기

<TBD> See 109 page

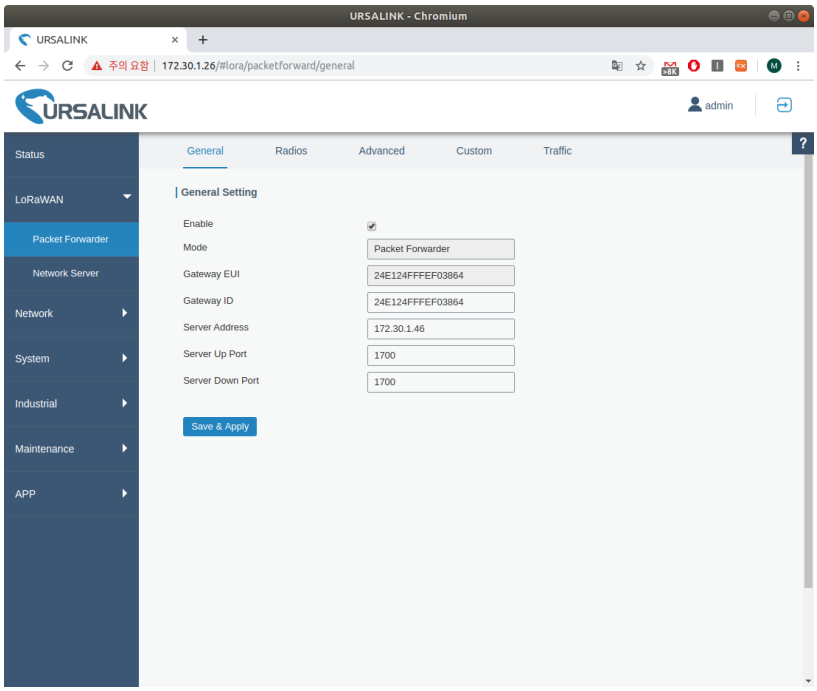
<UrsaLink OpenVPN Client의 문제점>

1. Version이 좀 낮은 것 같다.
2. Authentication Algorithm을 선택할 방법이 없다. 무조건 SHA1인 것 같다.
3. 암호 알고리즘의 선택이 폭이 좁다.
4. TLS encryption 방법을 지원 안한다.

…

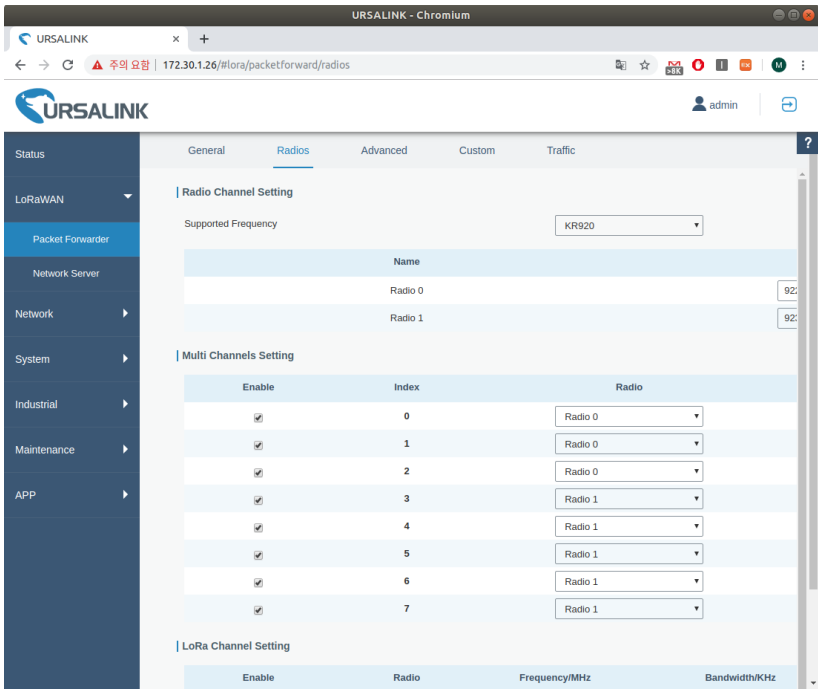
4) LoRaWAN 설정하기

자, 그럼 이제 부터는 OpenVPN 작업을 하느라 그동안 미뤄왔던 LoRaWAN 설정으로 넘어가도록 하겠다. WebUI 좌측의 Packet Forwarder 메뉴를 선택하고, Server Address를 LoRa Server 주소(여기서는 172.30.1.46)로 입력하고, Save & Apply 버튼을 선택한다.

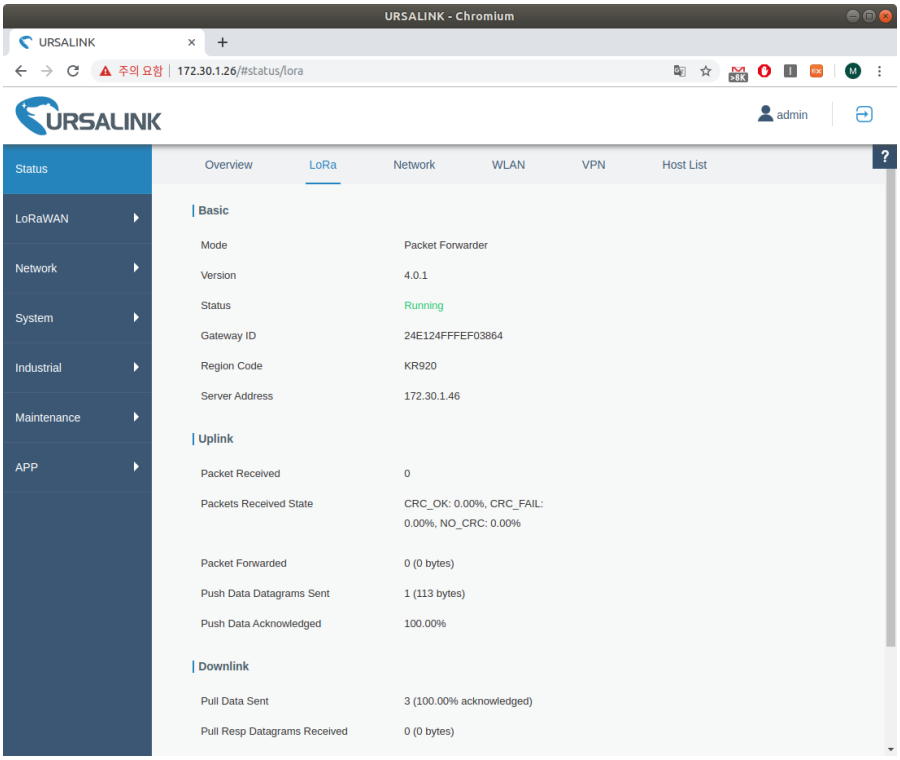


[그림 21.11] UrsaLink UG85 LoRaWAN - Packet Forwarder 설정

다음으로 KR920 주파수를 선택한다(이미 default로 KR920이 선택되어 있음).

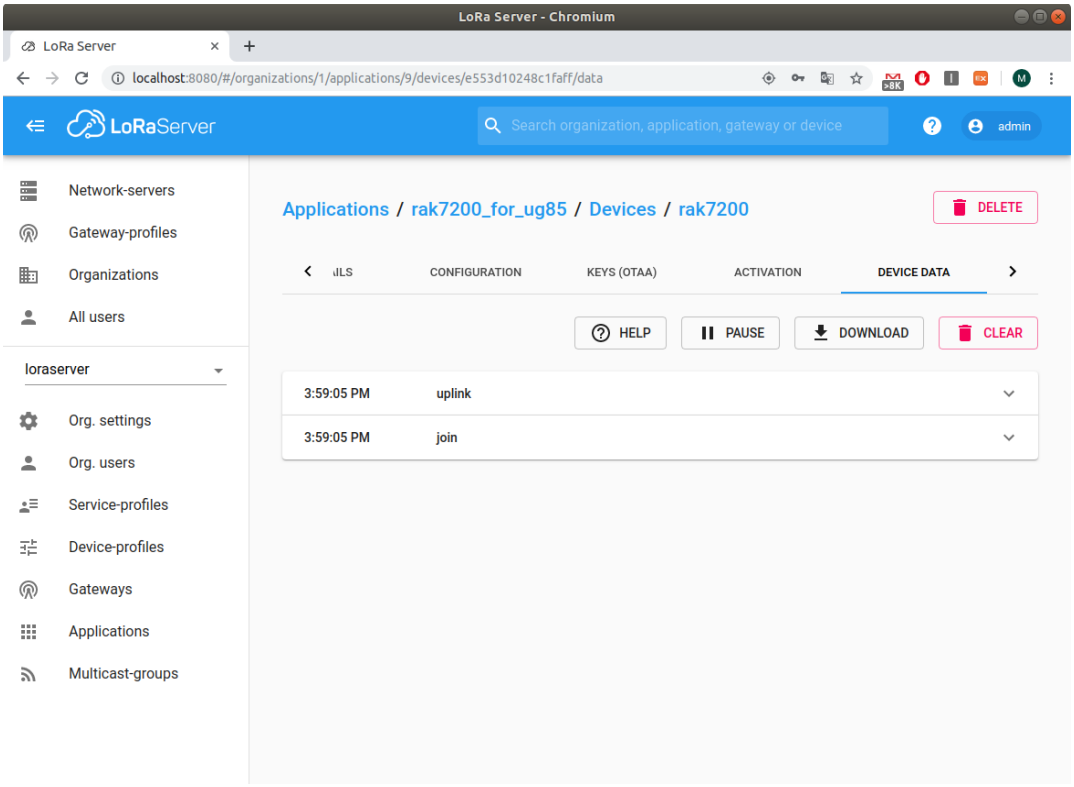


[그림 21.12] UrsaLink UG85 LoRaWAN - KR920 설정



[그림 21.13] UrsaLink UG85 LoRa Status

LoRa Server 서버 설정과 관련해서는 앞서 이미 여러 차례 소개한 바 있으므로, 여기서는 관련 내용을 모두 생략하기로 한다. 아래 그림에 보이는 것과 같이 RAK7200 센서로 부터 정보가 제대로 들어옴을 알 수 있다.



[그림 21.14] LoRa Server LoRa Packet 수신