

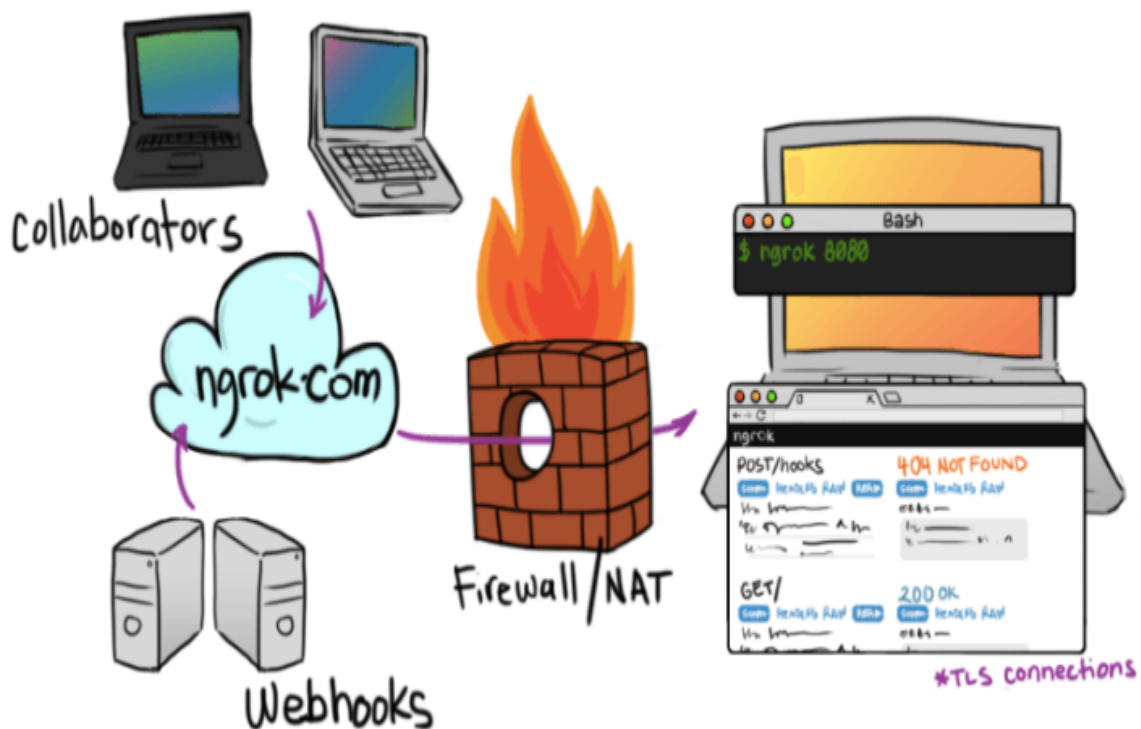
การเข้าถึง webserver ในตัว esp8266 จาก internet โดยใช้ ngrok แทน forward port และ dynamic dns

<http://cmmakerclub.com/2016/11/esp8266/access-webserver-on-esp8266-from-internet/>

การเข้าถึง webserver ในตัว esp8266 จาก internet โดยใช้ ngrok แทน forward port และ dynamic dns

HTML Content

ปกติแล้วการเข้าถึง esp8266 ที่ต่อ WiFi ในโหมด STA นั้น เราไม่มีทางเข้าถึงจากนอกวงแลนได้เลย ใหนจะติด firewall ใหนจะติด NAT
ถ้าเป็นเมื่อก่อนผมคงหาวิธีการ ทำ dynamic dns แล้วก็ forward port เข้าไปที่ esp8266 แล้วล่ะครับ
แต่ทุกวันนี้มีความรู้มากขึ้น ก็เลยเลือกใช้วิธีการ [sshtunnel](#) เอาครับ แต่ก็ยังไม่สะดวกสบายนัก
วันนี้เลยเอา [ngrok](#) (อ่านว่า en-grok) มานำเสนอครับ



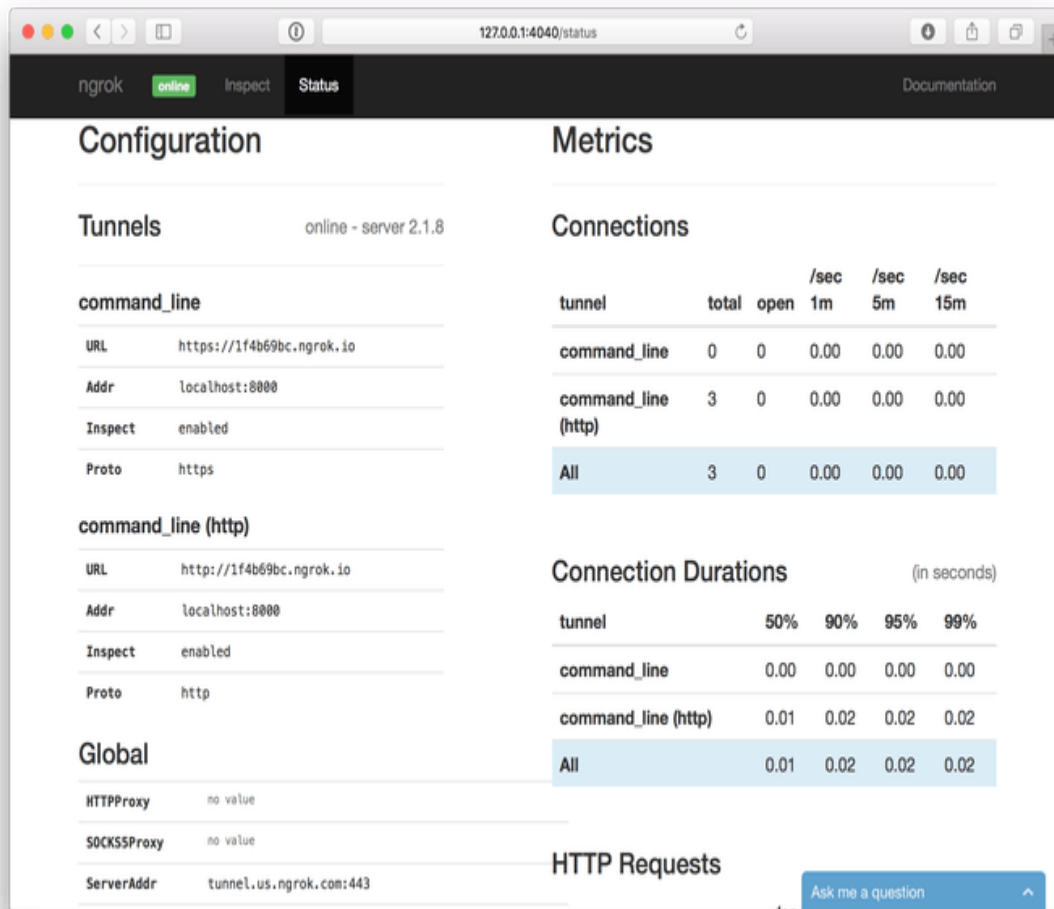
ngrok สามารถนำ localhost ในวงแลนของเราออกไปสู่การเชื่อมต่อผ่านอินเทอร์เน็ตได้ครับ ไม่ว่าจะเป็น http, tcp ได้ทำได้เลยครับ

แต่จุดเด็ด หรือคำถามที่จะต้องเกิดขึ้นมาคือแล้วมัน expose หรือเปิด service พวกนี้ไปที่ไหนล่ะ? เราไม่ต้องมี cloud service เอาไว้ทำ tunnel พวกนี้หรือ?

ถ้าเราใช้ ngrok ปกติก็ไม่ต้องมี cloud server เลยครับ เพราะตัว ngrok จะมี ngrok.io ให้เราใช้ทั้ง http และ https เลยสบาย สมายนะ แต่ package ตัวฟรีก็มีจำกัด connection นะครับ

เราสามารถสั่งแค่ ngrok http 8000 เท่านั้นก็นำ localhost เราขึ้นไปสู่ internet แล้วง่ายไหมล่ะครับ!!

และยังมี Web Interface ไว้ให้ monitor ดูด้วยนะ!! แจ่วเนอะ



วิธีการติดตั้งไม่ยากเลยครับ เพียงแค่ดาวน์โหลดมาแตกไฟล์ แล้วก็เปิดโปรแกรมได้เลยครับ!
ซึ่งรองรับหมดเลยไม่ว่าจะเป็น Mac OS X, Windows, Linux, Linux
ARM, FreeBSD ทั้ง 32-bit และ 64-bit เลยนะครับ

Download and Installation

ngrok is easy to install. Download a single binary with *zero run-time dependencies* for any major platform. Unzip it and then run it from the command line.

Step 1: Download ngrok

Mac OS X 64-Bit

Download

Windows 64-Bit

Download

Linux 64-Bit

Download

Linux ARM

Download

FreeBSD 64-Bit

Download

32-bit platforms

Step 2: Unzip it

On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip.

```
$ unzip /path/to/ngrok.zip
```

Step 3: Run it!

Read the [documentation](#) on how to use ngrok. Try it out by running it from the command line:

วิธีใช้งานก็ง่ายมาก สามารถดูได้จาก help ได้ตามคำสั่งนี้ครับ

```
./ngrok help
./ngrok http 8000
```

ทีนี้มาสร้าง webserver ฟัง esp8266 กันมั่งครับ ใช้ตัวอย่างที่ติดมาได้เลยครับ แบบนี้

```
#include <ESP8266WiFi.h>
```

```
#include <WiFiClient.h>
```

```
#include <ESP8266WebServer.h>
```

```
#include <ESP8266mDNS.h>
```

```
const char* ssid = "ESPERT-3020";
```

```
const char* password = "espertap";
```

```
ESP8266WebServer server(80);
```

```
const int led = 13;
```

```
void handleRoot() {
```

```
    digitalWrite(led, 1);
```

```
    server.send(200, "text/plain", "hello from esp8266!");
```

```
    digitalWrite(led, 0);
```

```
}
```

```
void handleNotFound(){

digitalWrite(led, 1);

String message = "File Not Found\n\n";

message += "URI: ";

message += server.uri();

message += "\nMethod: ";

message += (server.method() == HTTP_GET)?"GET":"POST";

message += "\nArguments: ";

message += server.args();

message += "\n";

for (uint8_t i=0; i<server.args(); i++){

message += " " + server.argName(i) + ": " + server.arg(i) + "\n";

}

server.send(404, "text/plain", message);

digitalWrite(led, 0);

}
```

```
void setup(void){

pinMode(led, OUTPUT);

digitalWrite(led, 0);

Serial.begin(115200);

WiFi.begin(ssid, password);

Serial.println("");

// Wait for connection

while (WiFi.status() != WL_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.print("Connected to ");

Serial.println(ssid);
```

```
Serial.print("IP address: ");
```

```
Serial.println(WiFi.localIP());
```

```
if (MDNS.begin("esp8266")) {
```

```
Serial.println("MDNS responder started");
```

```
}
```

```
server.on("/", handleRoot);
```

```
server.on("/inline", [](){
```

```
server.send(200, "text/plain", "this works as well");
```

```
});
```

```
server.onNotFound(handleNotFound);
```

```
server.begin();
```



```
Serial.println("HTTP server started");
```

```
}
```

```
void loop(void){
```

```
server.handleClient();
```

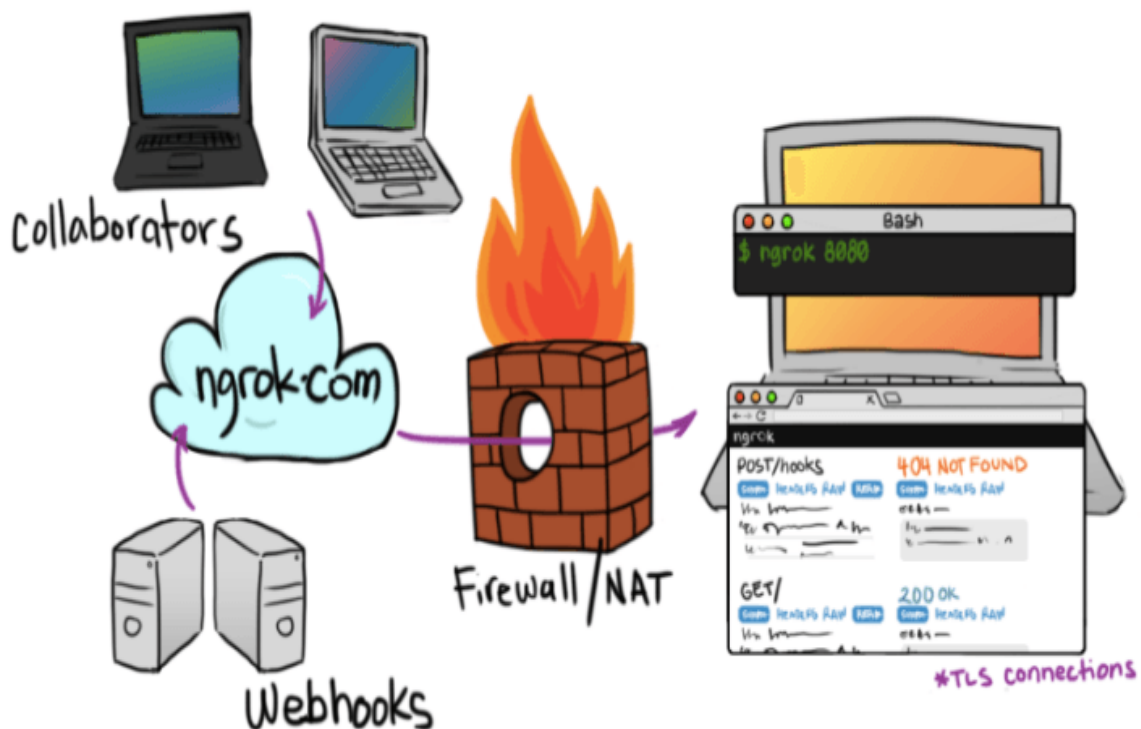
```
}
```

[esp8266-webserver.ino](https://github.com/esp8266/esp8266-webserver.ino) hosted with ❤ by [GitHub](https://github.com)

[view raw](#)

ต่อ WiFi แล้วก็เอา ip มาเปิดใน browser ครับ

ปกติแล้วการเข้าถึง esp8266 ที่ต่อ WiFi ในโหมด STA นั้น เราไม่มีทางเข้าถึงจากนอกวง
แลนได้เลย ไหนจะติด firewall ไหนจะติด NAT
ถ้าเป็นเมื่อก่อนผมคงหาวิธีการ ทำ dynamic dns แล้วก็ forward port เข้าไปที่ esp8266
แล้วล่ะครับ
แต่ทุกวันนี้มีความรู้มากขึ้น ก็เลยเลือกใช้วิธีการ [ssh tunnel](#) เอาครับ แต่ก็ยังไม่สะดวกสบาย
นัก
วันนี้เลยเอา [ngrok](#) (อ่านว่า en-grok) มานำเสนอครับ



ngrok สามารถนำ localhost ในวงแลนของเราออกไปสู่การเชื่อมต่อผ่านอินเทอร์เน็ตได้ครับ
ไม่ว่าจะเป็น http, tcp ได้ทำได้ครับ
แต่จุดเด่น หรือคำถามที่จะต้องเกิดขึ้นมาคือแล้วมัน expose หรือเปิด service พวกนี้ไป
ที่ไหนล่ะ? เราไม่ต้องมี cloud service เอาไว้ทำ tunnel พวกนี้หรือ?
ถ้าเราใช้ ngrok ปกติก็ไม่ต้องมี cloud server เลยครับ เพราะตัว ngrok จะมี ngrok.io ให้
เราใช้ทั้ง http และ https เลยสบาย สบายเนอะ แต่ package ตัวฟรีก็มีจำกัด connection
นะครับ
เราสามารถสั่งแค่ ngrok http 8000 เท่านั้นก็นำ localhost เราขึ้นไปสู่ internet แล้ว
ง่ายไหมล่ะครับ!!
และยังมี Web Interface ไว้ให้ monitor ดูด้วยนะ!! แจ่วเนอะ

