# NGROK

### Easy HTTPS

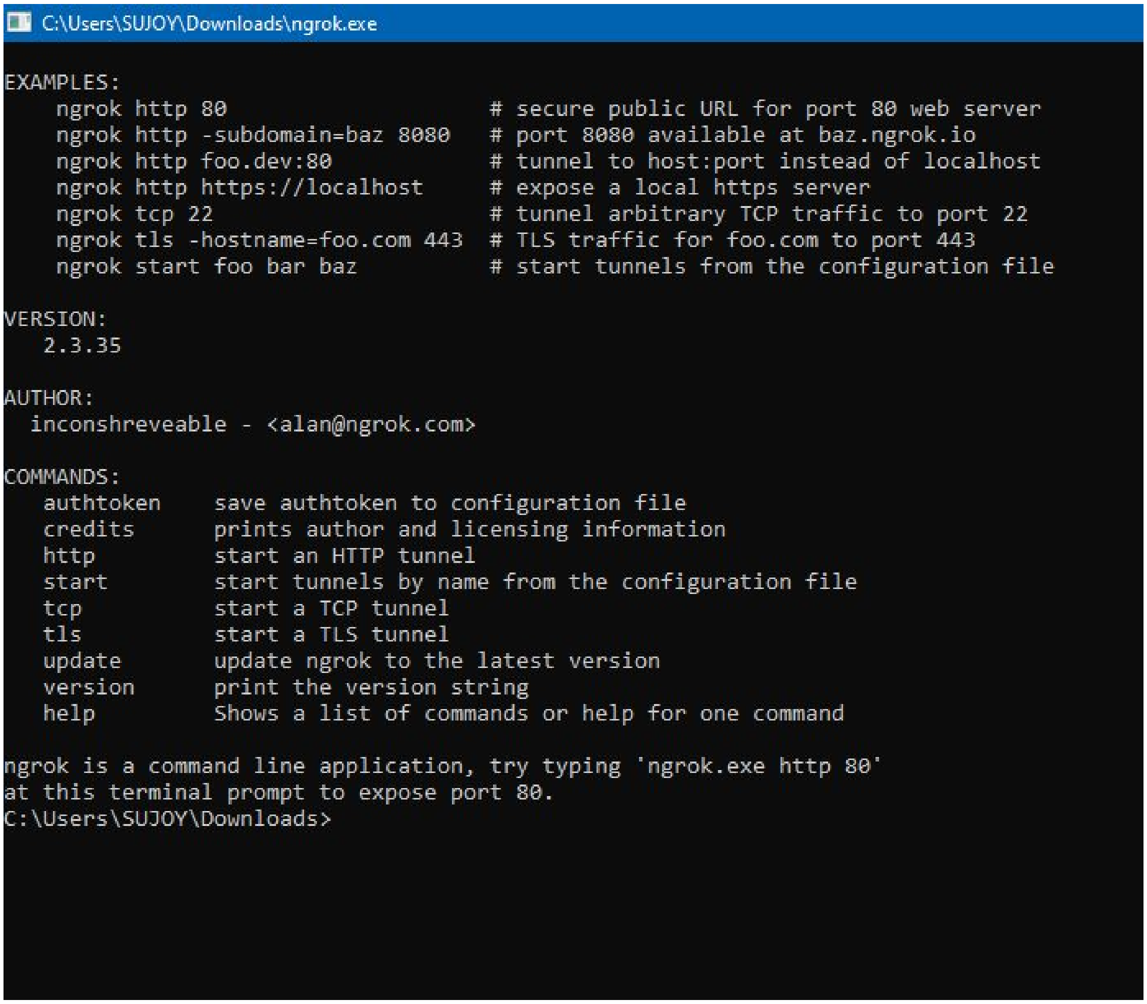
Instantly create a public HTTPS url for a website running locally on your development machine. ngrok offloads TLS so you don't have to worry about your configuration.

### Authenticated Access

Set HTTP auth credentials to protect access to your tunnel and those you share it with. ngrok also supports other authentication methods.  Add OAuth to your endpoints automatically; no code required.

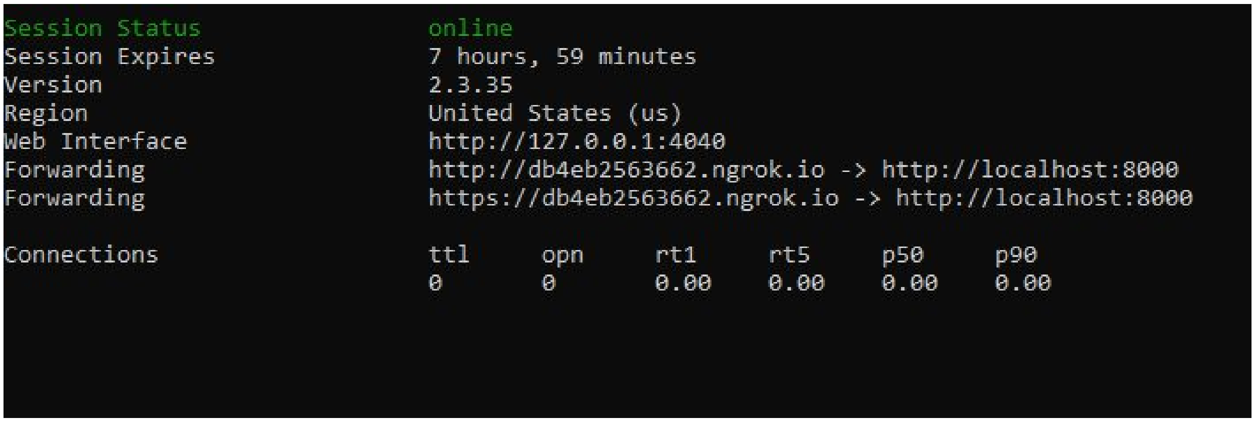
**Steps:**

1. Visit www.ngrok.com and download the application based on your OS.
2. Open the application file as follows:



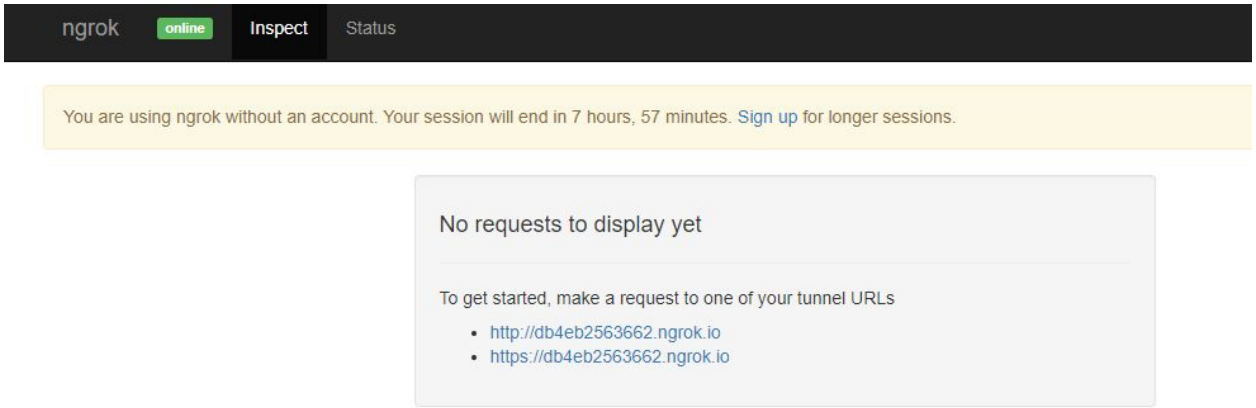
```
EXAMPLES:
    ngrok http 80                    # secure public URL for port 80 web server
    ngrok http -subdomain=baz 8080   # port 8080 available at baz.ngrok.io
    ngrok http foo.dev:80            # tunnel to host:port instead of localhost
    ngrok http https://localhost     # expose a local https server
    ngrok tcp 22                     # tunnel arbitrary TCP traffic to port 22
    ngrok tls -hostname=foo.com 443  # TLS traffic for foo.com to port 443
    ngrok start foo bar baz          # start tunnels from the configuration file

VERSION:
   2.3.35

AUTHOR:
  inconshreveable - <alan@ngrok.com>

COMMANDS:
   authtoken    save authtoken to configuration file
   credits      prints author and licensing information
   http         start an HTTP tunnel
   start        start tunnels by name from the configuration file
   tcp          start a TCP tunnel
   tls          start a TLS tunnel
   update       update ngrok to the latest version
   version      print the version string
   help         Shows a list of commands or help for one command

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\SUJOY\Downloads>
```

3. Enter the port command in terminal **./ngrok port 3000**



```
Session Status                online
Session Expires               7 hours, 59 minutes
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://db4eb2563662.ngrok.io -> http://localhost:8000
Forwarding                    https://db4eb2563662.ngrok.io -> http://localhost:8000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

The localhost port 8000 is set up and is tunneled to a public IP **127.0.0.1:4040.**



You are using ngrok without an account. Your session will end in 7 hours, 57 minutes. Sign up for longer sessions.

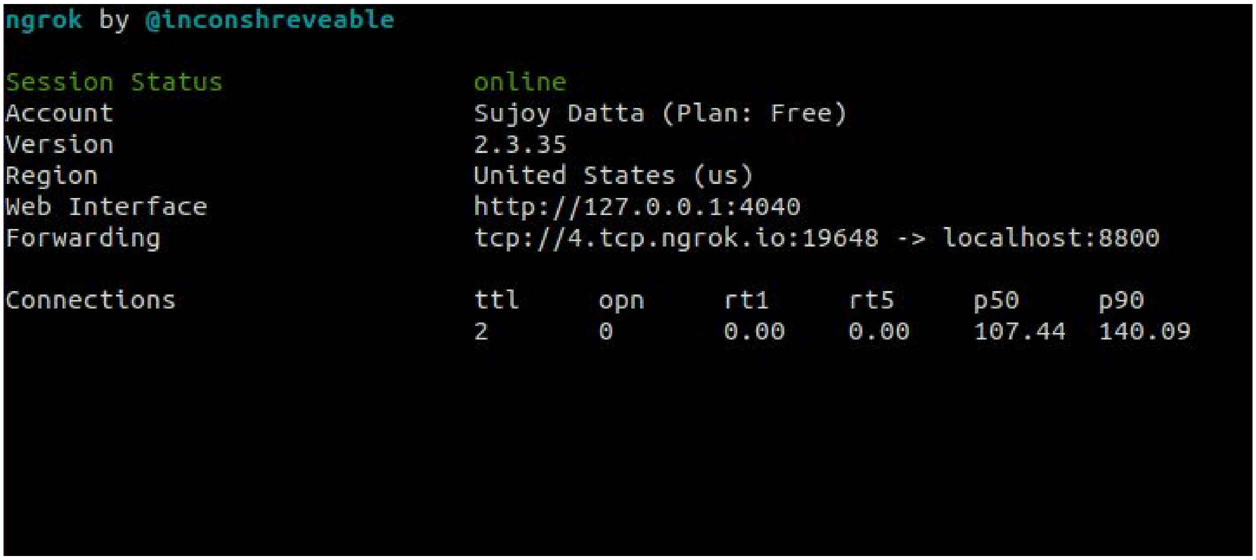
No requests to display yet

To get started, make a request to one of your tunnel URLs

- http://db4eb2563662.ngrok.io
- https://db4eb2563662.ngrok.io

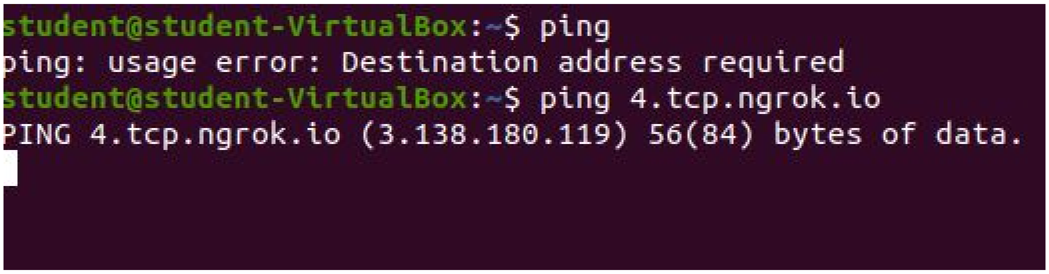
**Socket Connection:**

1.



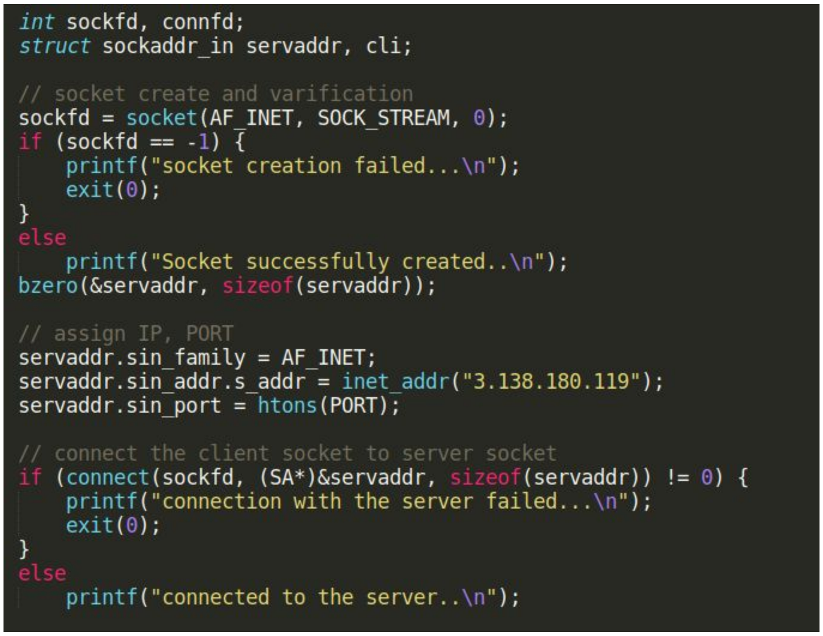
```
ngrok by @inconshreveable

Session Status                online
Account                       Sujoy Datta (Plan: Free)
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://4.tcp.ngrok.io:19648 -> localhost:8800

Connections                   ttl     opn     rt1     rt5     p50     p90
                              2       0       0.00    0.00    107.44  140.09
```

2. Gaining the public IP using ping.



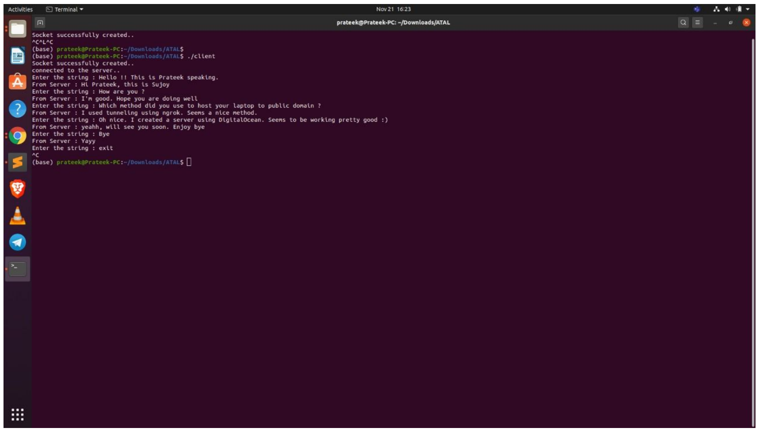
```
student@student-VirtualBox:~$ ping
ping: usage error: Destination address required
student@student-VirtualBox:~$ ping 4.tcp.ngrok.io
PING 4.tcp.ngrok.io (3.138.180.119) 56(84) bytes of data.
```

3. Changes in client end.



```c
int sockfd, connfd;
struct sockaddr_in servaddr, cli;

// socket create and varification
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd == -1) {
    printf("socket creation failed...\n");
    exit(0);
}
else
    printf("Socket successfully created..\n");
bzero(&servaddr, sizeof(servaddr));

// assign IP, PORT
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr("3.138.180.119");
servaddr.sin_port = htons(PORT);

// connect the client socket to server socket
if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr)) != 0) {
    printf("connection with the server failed...\n");
    exit(0);
}
else
    printf("connected to the server..\n");
```

4. Connect to the server and run the client on a different network, and you can chat.



```
Socket successfully created..
^C^L^C
(base) prateek@Prateek-PC:~/Downloads/ATAL$
(base) prateek@Prateek-PC:~/Downloads/ATAL$ ./client
Socket successfully created..
connected to the server..
Enter the string : Hello !! This is Prateek speaking.
From Server : Hi Prateek, this is Sujoy
Enter the string : How are you ?
From Server : I'm good. Hope you are doing well
Enter the string : Which method did you use to host your laptop to public domain ?
From Server : I used tunneling using ngrok. Seems a nice method.
Enter the string : Oh nice. I created a server using DigitalOcean. Seems to be working pretty good :)
From Server : yeahh, will see you soon. Enjoy bye
Enter the string : Bye
From Server : Yayy
Enter the string : exit
^C
(base) prateek@Prateek-PC:~/Downloads/ATAL$
```