

Setting up Ngrok for Tunneling

Step 1: Signup on Ngrok Website

- Go to the Ngrok website (<https://dashboard.ngrok.com/signup>) in your web browser.
- Sign up for a Ngrok account if you don't have one already.

Step 2: Installing Ngrok on Ubuntu

1. Open your Ubuntu terminal.
2. Download Ngrok:

...

```
wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
```

...

```
(venv) chinni@LAPTOP-UVNKR98P:~/W3$ wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
--2023-10-08 15:53:24-- https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
Resolving bin.equinox.io (bin.equinox.io)... 18.205.222.128, 52.202.168.65, 54.161.241.46, ...
Connecting to bin.equinox.io (bin.equinox.io)|18.205.222.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13921656 (13M) [application/octet-stream]
Saving to: 'ngrok-stable-linux-amd64.zip'

ngrok-stable-linux-amd64.zip  100%[=====>] 13.28M  13.7MB/s
.05
```

3. Install Unzip (if not already installed):

...

```
sudo apt install unzip
```

...

```
• (venv) chinni@LAPTOP-UVNKR98P:~/W3$ sudo apt install unzip
[sudo] password for chinni:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
```

4. Unzip the Ngrok package:

...

```
unzip ngrok-stable-linux-amd64.zip
```

...

```
• (venv) chinni@LAPTOP-UVNKR98P:~/W3$ unzip ngrok-stable-linux-amd64.zip
Archive:  ngrok-stable-linux-amd64.zip
  inflating: ngrok
```

5. Run Ngrok:

...

```
./ngrok
```

...

```
• (venv) chinni@LAPTOP-UVNKR98P:~/W3$ ./ngrok
NAME:
  ngrok - tunnel local ports to public URLs and inspect traffic

DESCRIPTION:
  ngrok exposes local networked services behinds NATs and firewalls to the
  public internet over a secure tunnel. Share local websites, build/test
  webhook consumers and self-host personal services.
  Detailed help for each command is available with 'ngrok help <command>'.
  Open http://localhost:4040 for ngrok's web interface to inspect traffic.
```

Step 3: Authenticate Ngrok

- Replace ``<your-authtoken-from-ngrok-website>`` with the actual authentication token you obtained from the Ngrok website.

- Run the following command to authenticate Ngrok:

...

```
./ngrok authtoken <your-authtoken-from-ngrok-website>
```

...

```
• (venv) chinni@LAPTOP-UVNKR98P:~/W3$ ./ngrok authtoken 2kPTy7EmTYyHoK
Authtoken saved to configuration file: /home/chinni/.ngrok2/ngrok.yml
```

Step 4: Running Your Application

- Start your Python application (assuming it's named `app.py`):

...

```
python3 app.py
```

...

```
• (venv) chinni@LAPTOP-UVNKR98P:~/W3$ python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production
  environment.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 417-416-366
```

Step 5: Create a Tunnel (Open a new terminal window.)

In the new terminal, create an HTTP tunnel on the port where your app.py is running. Replace your-app-port with the actual port number where your application is running:

```
...  
./ngrok http your-app-port  
...
```

```
• (venv) chinni@LAPTOP-UVNKR98P:~/W3$ ./ngrok http 5000
```

This will generate a public URL for your local application, allowing you to access it remotely using the same port as your app.py is running on.

```
ngrok by @inconshreveable  
  
Session Status      online  
Account             skotturu655@student.sfbu.edu (Plan: Free)  
Version             2.3.41  
Region              United States (us)  
Web Interface       http://127.0.0.1:4040  
Forwarding           http://92d5-24-6-80-12.ngrok-free.app -> http://localhost:5000  
Forwarding           https://92d5-24-6-80-12.ngrok-free.app -> http://localhost:5000  
  
Connections         ttl    opn    rt1    rt5    p50    p90  
                   0      0      0.00   0.00   0.00   0.00
```

Click on the link generated to see your website.

