ngrok

lets say I have to forward service called serviceX (you can imagine any as you want like http, ftp …) to your machine or in internet so you can access that service , so I need to run agnet in my machine in background that agent can make this possible that is called port forwarding and that agent is ngrok.

(there is many services that provide port forwarding not ngrok only)

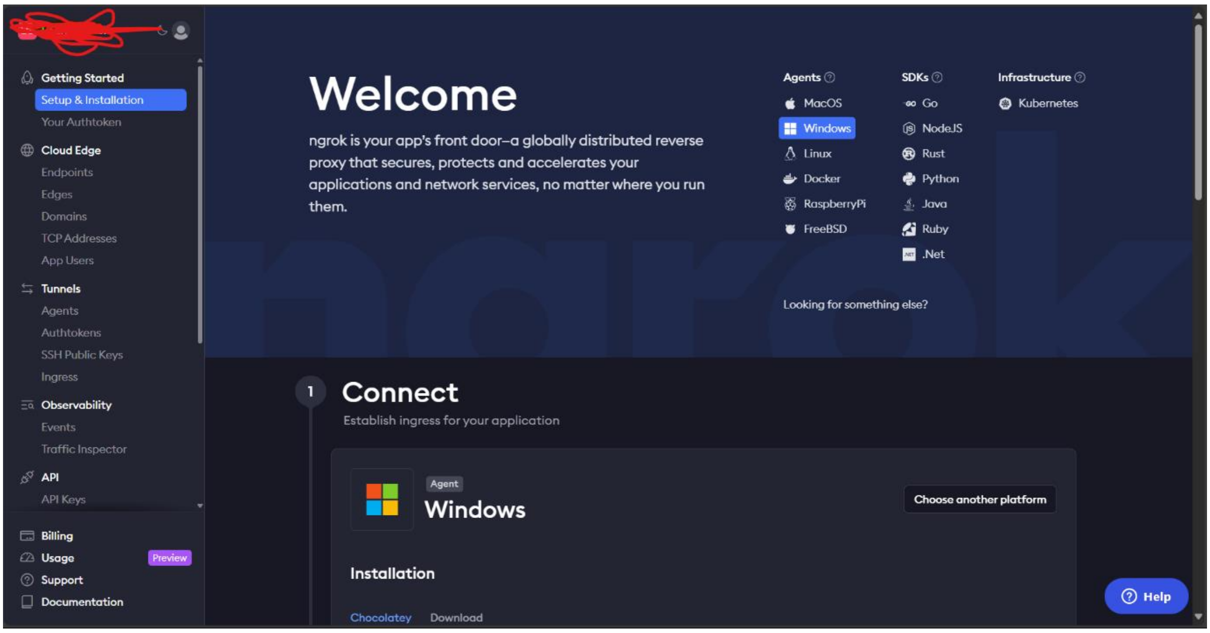
ngrok bind my serviceX to port number generated randomly by ngrok (also you can select specific one) so that service now accessible in WAN.

Example: I host website , to access that service is http so I will bind my website at specific port using ngrok → mysite.com:4444

So anyone in WAN can access mysite.com using browser.

How to Download and use ngrok in kali linux:

1. First goto ngrok website and register and then login with that account so with that unique account we can create our own ngrok for our kali machine.
   After login you will redirected to screen like this



[redacted]

Welcome

ngrok is your app's front door–a globally distributed reverse proxy that secures, protects and accelerates your applications and network services, no matter where you run them.

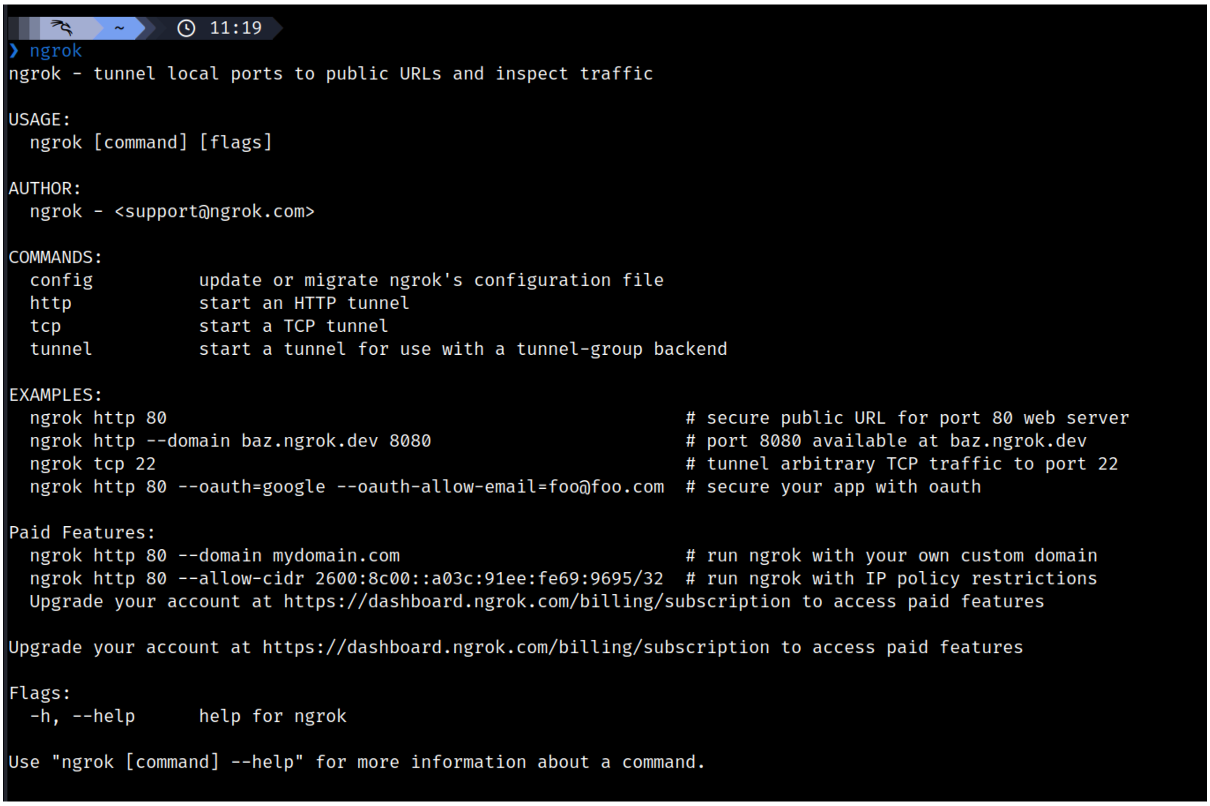
Looking for something else?

1 Connect

Establish ingress for your application

2. Now choose linux operating system and after that you can see all the steps , do each step one by one after that you can use ngrok command or service successfully in kali linux.

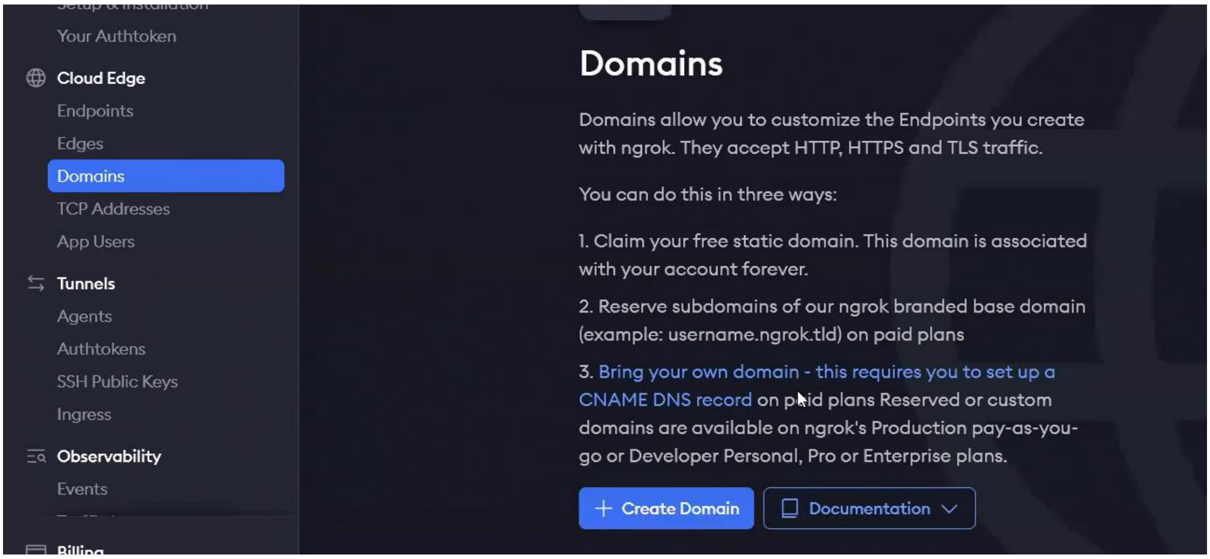
After installation type ngrok and you can see help file it shows ngrok installed successfylly.



```
~ 11:19
❯ ngrok
ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
  ngrok [command] [flags]

AUTHOR:
  ngrok - <support@ngrok.com>

COMMANDS:
  config          update or migrate ngrok's configuration file
  http            start an HTTP tunnel
  tcp             start a TCP tunnel
  tunnel          start a tunnel for use with a tunnel-group backend

EXAMPLES:
  ngrok http 80                                                # secure public URL for port 80 web server
  ngrok http --domain baz.ngrok.dev 8080                       # port 8080 available at baz.ngrok.dev
  ngrok tcp 22                                                 # tunnel arbitrary TCP traffic to port 22
  ngrok http 80 --oauth=google --oauth-allow-email=foo@foo.com  # secure your app with oauth

Paid Features:
  ngrok http 80 --domain mydomain.com                          # run ngrok with your own custom domain
  ngrok http 80 --allow-cidr 2600:8c00::a03c:91ee:fe69:9695/32  # run ngrok with IP policy restrictions
  Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Flags:
  -h, --help      help for ngrok

Use "ngrok [command] --help" for more information about a command.
```

To setup port forwarding :

Go to ngrok account and go to domain and then click to create a domain



Domains

Domains allow you to customize the Endpoints you create with ngrok. They accept HTTP, HTTPS and TLS traffic.

You can do this in three ways:

1. Claim your free static domain. This domain is associated with your account forever.

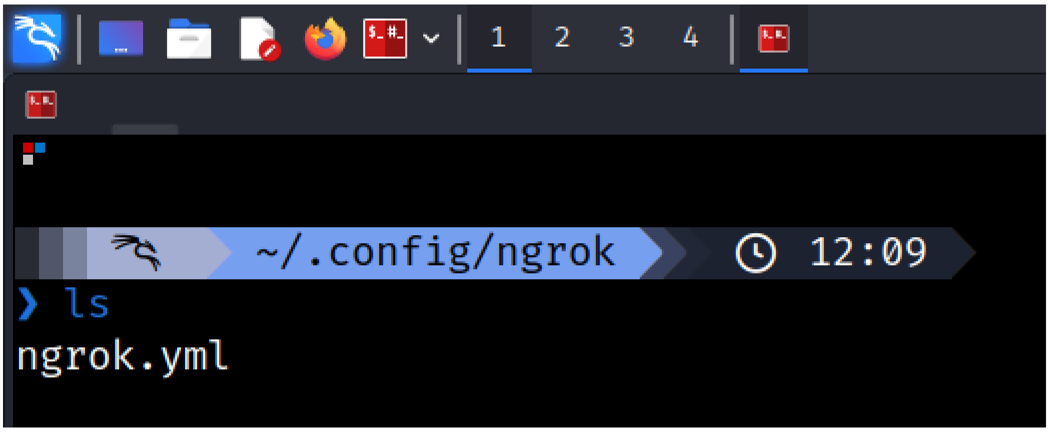
2. Reserve subdomains of our ngrok branded base domain (example: username.ngrok.tld) on paid plans

3. Bring your own domain - this requires you to set up a CNAME DNS record on paid plans Reserved or custom domains are available on ngrok's Production pay-as-you-go or Developer Personal, Pro or Enterprise plans.

Random domain generated by ngrok if you not satisfy with that name you can delete that and create new one until you found you love.

Copy that domain and we have to paste in one file, that file created automatically when we set up ngrok

Location of that file is : ~/.config/ngrok/ngrok.yml

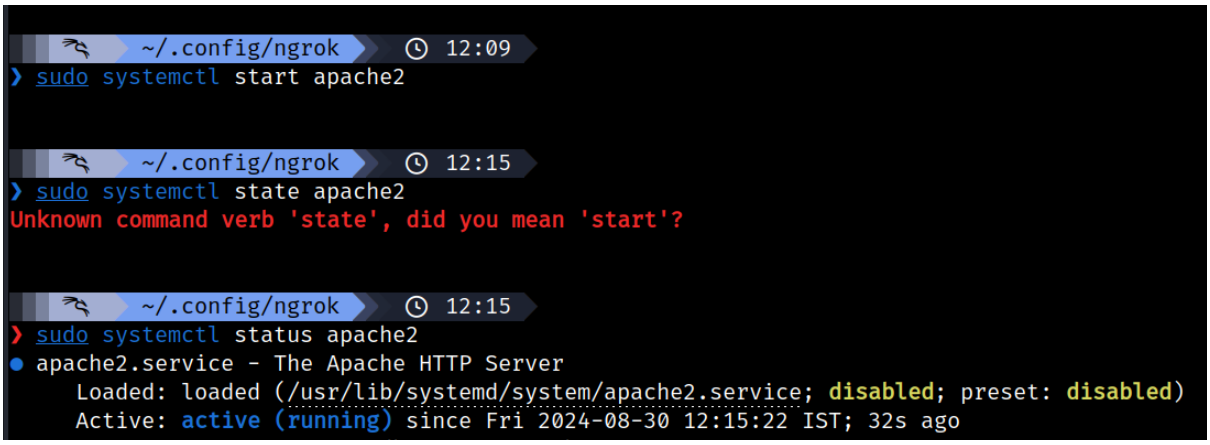


```
~/.config/ngrok   12:09
❯ ls
ngrok.yml
```

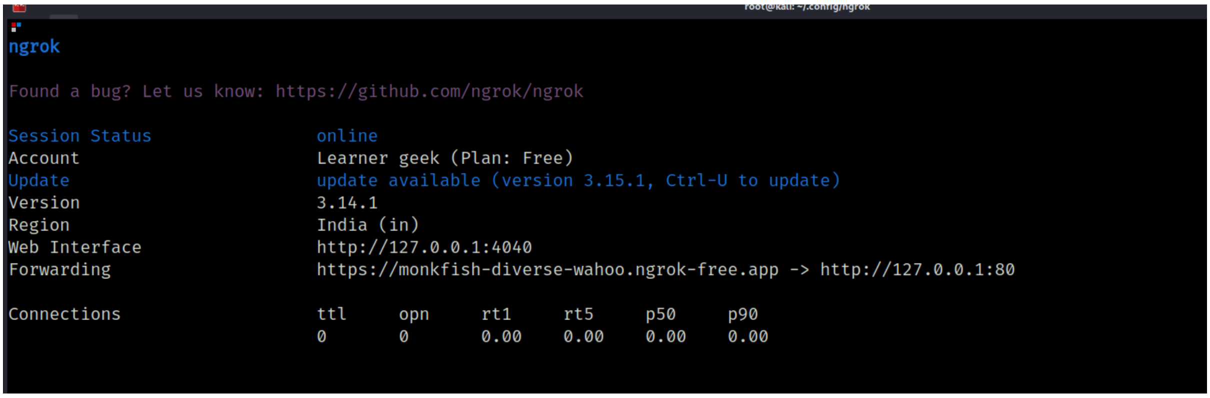
Paste that domain you choose from ngrok account in this file , I will not show this here because there is my account auth key.

After that you can run command ‘ngrok start -all’ but before that start apache2 service on your machine.

After apache2 active then you can run ‘ngrok start -all’ and you got your domain in running stat (means you can access your domain in WAN until ‘ngrok start -all command’ is running)



```
~/.config/ngrok   12:09
❯ sudo systemctl start apache2

~/.config/ngrok   12:15
❯ sudo systemctl state apache2
Unknown command verb 'state', did you mean 'start'?

~/.config/ngrok   12:15
❯ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-08-30 12:15:22 IST; 32s ago
```



```
ngrok

Found a bug? Let us know: https://github.com/ngrok/ngrok

Session Status                online
Account                       Learner geek (Plan: Free)
Update                        update available (version 3.15.1, Ctrl-U to update)
Version                       3.14.1
Region                        India (in)
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://monkfish-diverse-wahoo.ngrok-free.app -> http://127.0.0.1:80

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

Now you can hold ctrl and press on your domain in above image you can access your domain.

If you go to domain https://monkfish-diverse-wahoo.ngrok-free.app a page open is default apache2 page that is located at /var/www/html.

In this directory you can find index.html file so you can edit this index.html file as you want

Example, you can make your own website or webpage and move all files at /var/www/html

You can save apache2 files at other location because that is essential.