Getting Started with ngrok

This guide will get you up and running with the ngrok agent, giving you a secure way to access your local service from anywhere in the world.

We'll do this in four steps:

1. Starting a local service
2. Installing the ngrok agent
3. Connecting your agent to your ngrok account
4. Start ngrok

Step 1: Starting a local web service

First, you'll need some sort of web service running on your machine. It should be available at http://localhost:[any port]. If you already have one, you can skip to Step 2. If not, we'll set one up using Python SimpleHTTPServer (ngrok actually has a built in file server but let's not worry about that now).

If you don't have a web service running, you can set one up for this demo using Python SimpleHTTPServer.

1. Create a new directory, we'll call it `~/ngrok-rocks`
2. Move into that directory and create a file named `index.html` with a single line of text: `Hello, World!`
3. From that folder, run `python3 -m http.server`. This will start a web server on port 8000 serving the contents of that directory.
4. Open `http://localhost:8000` in your browser

You should see `Hello, World!` in your browser. If you have any trouble getting things working, see this page for help.

Unfortunately, this service is only available on your local machine for now. Now let's use ngrok to securely share it with the world. For the rest of this guide, we will assu

Help

working with a web service running at http://localhost:8000 but you should adjust the following commands to match your configuration.

Step 2: Install the ngrok Agent

The ngrok agent is the command line application that you will use to start your tunnels. The easiest way to get started is to use your favorite package manager to install ngrok.

For MacOS, use HomeBrew:

```
brew install ngrok/ngrok/ngrok
```

Yes, three times, for reasons.

For Linux, use Apt:

```
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | \
  sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && \
  echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | \
  sudo tee /etc/apt/sources.list.d/ngrok.list && \
  sudo apt update && sudo apt install ngrok
```

For Windows, use Chocolatey:

```
choco install ngrok
```

You'll need to run this in an Administrator Command Prompt.

If you don't have one of these package managers installed or prefer to install the ngrok agent yourself, visit the ngrok Download page for instructions and links.

You can test everything is working by running `ngrok -h` which should print the help text for the ngrok agent.

```
    $ ngrok -h
    NAME:
      ngrok - tunnel local ports to public URLs and inspect traffic

    USAGE:
      ngrok [command] [flags]

    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.

    Author:
      ngrok - <support@ngrok.com>

    TERMS OF SERVICE: https://ngrok.com/tos

    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 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 --domain=foo.com 443         # TLS traffic for foo.com
to port 443
      ngrok start foo bar baz                # start tunnels from the
configuration file

    COMMANDS:
      api                           use ngrok agent as an api client
      completion                    generates shell completion code
for bash or zsh
      config                        update or migrate ngrok's
configuration file
      credits                       prints author and licensing
```

```
information
      diagnose                        diagnose connection issues
      help                            Help about any command
      http                            start an HTTP tunnel
      service                         run and control an ngrok service
on a target operating system
      start                           start tunnels by name from the
configuration file
      tcp                             start a TCP tunnel
      tls                             start a TLS tunnel
      tunnel                          start a tunnel for use with a
tunnel-group backend
      update                          update ngrok to the latest version
      version                         print the version string

    OPTIONS:
          --config strings   path to config files; they are merged if
multiple
      -h, --help             help for ngrok
      -v, --version          version for ngrok
```

Step 3: Connect your agent to your ngrok account

Now that the ngrok agent is installed, let's connect it to your ngrok Account. If you haven't already, sign up (or log in) to the ngrok Dashboard and get your Authtoken. The ngrok agent uses the authtoken (sometimes called tunnel credential) to log into your account when you start a tunnel.

Copy the value and run this command to add the authtoken in your terminal.

```
ngrok config add-authtoken TOKEN
```

Step 4: Start ngrok

Start ngrok by running the following command.

```
ngrok http 8000
```

You should see something similar to the following console UI in your terminal.

```
    ngrok
(Ctrl+C to quit)

    Session Status                online
    Account                       inconshreveable (Plan: Free)
    Version                       3.0.0
    Region                        United States (us)
    Latency                       78ms
    Web Interface                 http://127.0.0.1:4040
    Forwarding                    https://84c5df439d74.ngrok-free.dev -
> http://localhost:8000

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

Now open the Forwarding URL in your browser and you should see your local web service. At first glance, it may not seem impressive, but there are a few key differences here:

- That URL is available to anyone in the world. Seriously, test it out by sending it to a friend.
- You are now using TLS (notice the 🔒 in your browser window) with a valid certificate without making any changes to your local service.

Since the whole world can access this URL, we need to secure it quickly. Let's stop the ngrok agent with `ctrl+c`.

Bonus Step 1: Securing your public endpoint

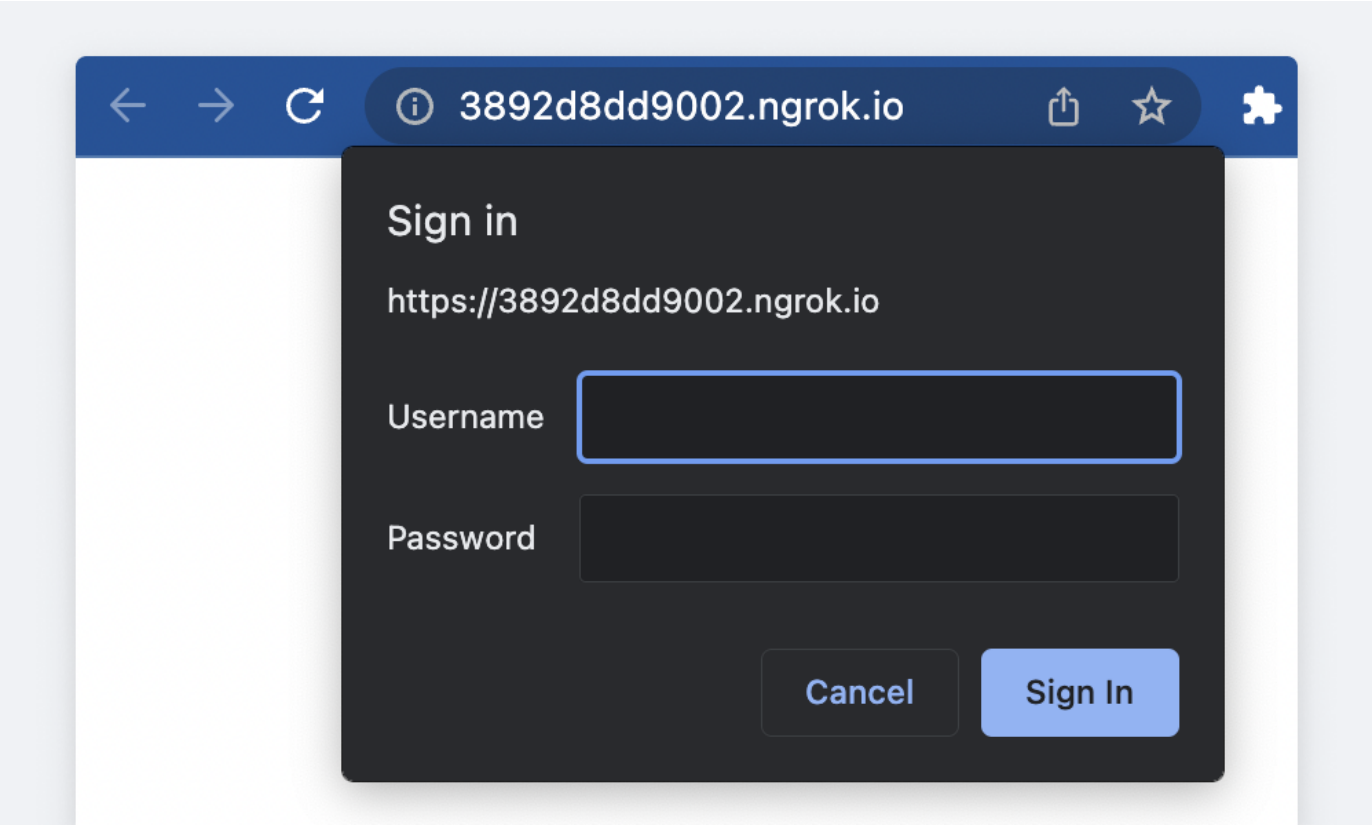
The ngrok agent allows you to dynamically add security to any public endpoint in a variety of ways with IP restrictions, HTTP Basic Authentication, OAuth 2.0, OpenID Connect, SAML, Webhook Verification, and even Mutual TLS.

To start simply, let's add HTTP Basic Authentication to your endpoint.

```
ngrok http 8000 --basic-auth 'ngrok:issecure'
```

Now when you access the new ngrok URL in your browser, you should be prompted for a username and password.

By the way, if you have a paid plan and want to keep the same URL each time you restart, use the `--domain` flag when starting the agent.



You now have a public URL for your web service secured by a username and password, still without modifying your web service.

Bonus Step 2: Add OAuth 2.0 to your web service

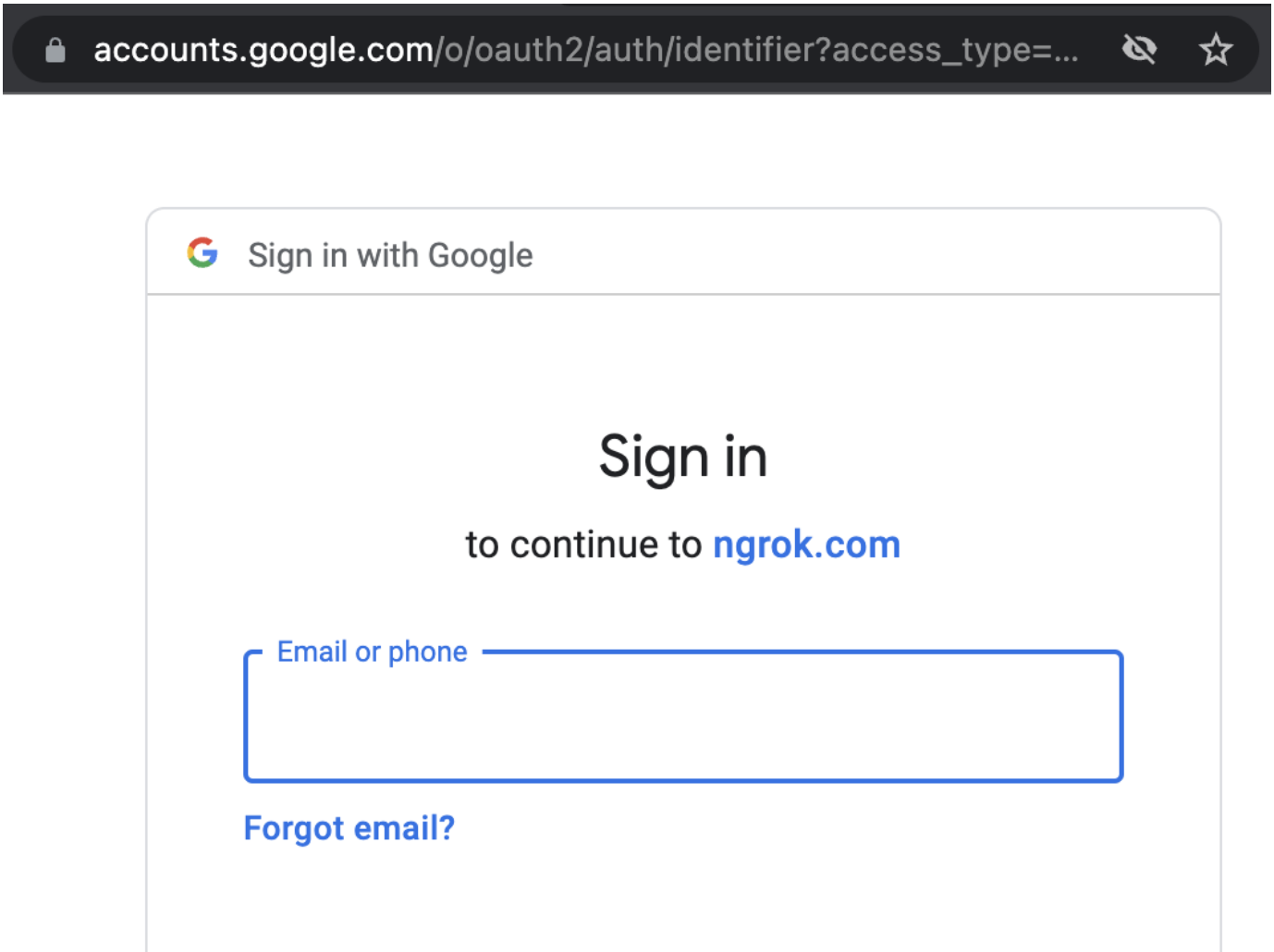
In most cases, you don't want to use a single set of shared credentials for all users (you can add as many basic auth credentials as you want, but still). Let's swap out Basic Auth for Google OAuth 2.0 in one line.

First stop the ngrok agent again (

```
ngrok http 8000 --oauth google
```

If you have a paid plan, you can explicitly specify the email addresses to allow with the `--oauth-allow-email` flag.

Now when you try to visit your new ngrok URL, you will be prompted to log in with your Google account (you can open in incognito to be sure). Again, this is without modifying your web service.



Next steps

That's it, but there's a lot more you can do with ngrok!

- Configure ngrok to use the same domain each time with `--domain`
- Tunnel other non-HTTP services such as SSH, RDP, or game servers using TCP Tunnels
- Bring your own custom domains to ngrok
- Add your API key and automate via the `ngrok api` command.
- Use ngrok Cloud Edges to dynamically reconfigure traffic to your ngrok agents

Edit this page

*Last updated on **Sep 15, 2023** by **Cody A Price***