



Search

CTRL

K

Overview

What is ngrok?

How does ngrok work?

Why ngrok?

What's new?

Getting Started

ENDPOINTS

HTTP

TLS

TCP

CONNECTIVITY

Agent

Overview

Web Inspection Interface

CLI

Configuration file

API

SSH Reverse Tunnel

Ingress Address

Changelog

Version Support Policy

Diagnose

Agent SDKs

Kubernetes

PLATFORM

Network Edge

API

Observability

IAM

Errors

WALKTHROUGHS

Guides

Integrations

FAQ



>

Agent

>

SSH Reverse Tunnel

SSH Reverse Tunnel Agent

Overview

SSH reverse tunneling (`ssh -R`) is an alternative mechanism deliver services via ngrok without running an [ngrok agent](#) or [Agent SDK](#).

The SSH reverse tunnel agent should not be confused with creating remote access to an SSH server via ngrok. If you want to use ngrok to create access to your own SSH server for remote access, please refer to the [using ngrok with ssh](#) documentation.

You should only ngrok via SSH if you really can't use an Agent or Agent SDK. The SSH reverse tunnel agent has [many functional limitations compared to the ngrok agent](#).

Example Usage

Random HTTP Endpoint

```
ssh -R 443:localhost:80 v2@connect.ngrok-agent.com http
```

Custom Domain

```
ssh -R example.ngrok.app:443:localhost:80 v2@connect.ngrok-agent.com http
```

Basic Auth

```
ssh -R 443:localhost:80 v2@connect.ngrok-agent.com http \  
--basic-auth "username1:password1" \  
--basic-auth "username2:password2"
```

OAuth

```
ssh -R 443:localhost:80 v2@connect.ngrok-agent.com http --oauth=google
```

Forward to non-local service

```
ssh -R 0:192.168.1.2:80 v2@connect.ngrok-agent.com http
```

Random TCP Endpoint

```
ssh -R 0:localhost:22 v2@connect.ngrok-agent.com tcp
```

Fixed TCP Endpoint

ON THIS PAGE

[Overview](#)

Example Usage

Random HTTP Endpoint

Custom Domain

Basic Auth

OAuth

Forward to non-local service

Random TCP Endpoint

Fixed TCP Endpoint

TLS Endpoint

Explicit Region Selection

Authentication

Command Syntax

Versioning

Differences from the Agent

Pricing

```
ssh -R 1.tcp.eu.ngrok.io:12345:localhost:3389 connect.eu.ngrok-agent.com tcp
```

TLS Endpoint

```
ssh -R app.example.com:443:localhost:443 v2@connect.ngrok-agent.com tls
```

Explicit Region Selection

Normally you will connect to ngrok's closest point of present via [Global Server Load Balancing](#), but you can also explicitly choose a region.

```
ssh -R 443:localhost:80 v2@connect.eu.ngrok-agent.com http
```

Authentication

Instead of an ngrok authtoken, when you use ngrok via the SSH reverse tunnel agent, it uses a public key for authentication. You'll first need to upload yours to the [SSH Public Keys page on your ngrok dashboard](#).

Copy your default SSH public key with:

Mac OS **Linux**

```
cat ~/.ssh/id_rsa.pub | pbcopy
```

or:

```
cat ~/.ssh/id_ed25519.pub | pbcopy
```

Command Syntax

ngrok does its best to honor the syntax of `ssh -R`. You may wish to consult `man ssh`, and the section devoted to the `-R` option for additional details. ngrok uses additional command line options to implement features that are not otherwise available via the `-R` syntax.

Let's break down the following command.

```
ssh -R \  
  app.example.com:443:127.0.0.1:8080 \  
  v2@connect.ngrok-agent.com \  
  http --basic-auth 'user:password'
```

An `ssh -R` command has the following components:

```
ssh -R \  
  "<remote name>:<remote port>:<local name>:<local port>" \  
  <user>@connect.ngrok-agent.com \  
  <command> [flags]
```

In our example:

- Remote Name: `app.example.com`. ngrok will listen on the domain 'app.example.com'. You may omit this value. If you do, ngrok chooses a random endpoint name.
- Remote Port: `443`. ngrok will listen for HTTPS traffic on port 443. The only valid values for HTTP endpoints are 80 and 443. For TLS endpoints it must be 443. You may `0` and ngrok will simply choose the appropriate port for you.
- Local Name: `127.0.0.1`. This is the local hostname or IP address that traffic will be sent to. It's most commonly `localhost`.
- Local Port: `8080`. This is the local port that traffic will be sent to.
- User: `v2`. ngrok uses the user portion of the command to version the command options. You may omit this value. If you do, ngrok will use the latest version.
- Command: `http`. This the type of endpoint to create. ngrok accepts either `http`, `tls` or `tcp`. This value is required.
- Flags: `--basic-auth 'user:password'`. Run the same command with the `--help` flag to get the list of supported flags or consult the [Agent CLI reference](#).

Versioning

ngrok uses the user portion of the SSH command to version the CLI syntax. The latest version is `v2`.

Differences from the Agent

When you use ngrok via SSH reverse tunnel, you will need to [upload an SSH public key](#) to authenticate with instead of using an ngrok authToken like the agent.

Additionally, you'll find that using ngrok via SSH has many functional limitations compared to the experience with the agent. An incomplete list of differences from the ngrok agent includes:

- Your endpoints won't automatically reconnect if there is a network interruption
- There is no equivalent to the agent's [traffic inspection interface](#)
- You can't create endpoints for multiple services with the same command
- You can't [forward to upstream https services](#)
- You can't create multiple endpoints over the same connection
- You can't [serve file system directories](#) with the `file://` protocol
- You can't terminate TLS at the agent when doing [zero-knowledge TLS](#)
- You can't run labeled tunnels for use with [Edges](#).

Pricing

The SSH reverse tunnel agent is available to all ngrok users at no additional charge. You only incur costs if resources you provision via its usage incur a cost.

 [Edit this page](#)

Last updated on **Oct 31, 2024** by **Danielle Thomas**

Previous

[← API](#)

Next

[Ingress Address →](#)

NGROK SERVICE

- Get Started
- Sign up
- Log in
- Download
- Docs
- Status

NGROK.COM

- Product
- Pricing
- Customers
- Solutions
- Partners
- Trust Portal

LEGAL

- Abuse
- DPA
- Privacy
- Security
- Terms of service
- Privacy Preferences

MORE

- Careers
- Blog
- Community
- Twitter
- LinkedIn
- GitHub

© ngrok 2024