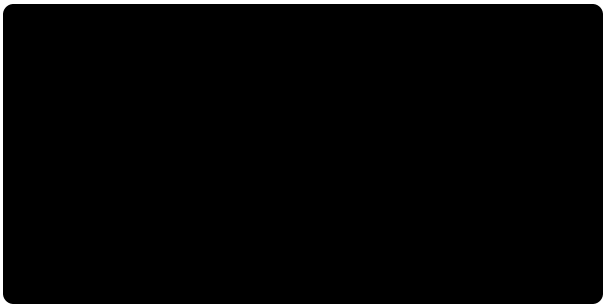
tailscale

Tailscale Funnel now available in beta

Parker Higgins, Shayne Sweeney, Maisem Ali and David Crawshaw on March 30, 2023

Tailscale Funnel, a tool that lets you share a web server on your private tailnet with the public internet, is now available as a beta feature for all users. With Funnel enabled, you can share access to a local development server, test a webhook, or even host a blog.



Funnel provides a DNS name tied to your node that becomes publicly accessible once enabled. When a user on the public internet requests your service, we use a secure Tailscale tunnel to forward those requests along.

Given our commitment to user privacy, we've set up Funnel so we never see decrypted packets going to and from your service. All connections require TLS encryption, and Tailscale will automatically provision valid certificates with which your node does the TLS termination. Because Funnel addresses are subdomains of ts.net, we could technically acquire valid certificates and do that termination server-side, but we believe that preserving user privacy is more important. You can always confirm we haven't obtained certificates for Funnel services by reviewing the public Certificate Transparency logs.

To get started, point Tailscale to your local development server and turn on Funnel. For example:

```
$ tailscale serve https / http://localhost:3000
$ tailscale funnel 443 on
```

Your local server is now reachable at its DNS name over the internet. You can turn Funnel on and off using the `tailscale funnel` command. To view the status of your Funnel, run:

```
$ tailscale funnel status
https://example.pango-lin.ts.net (Funnel on)
```

```
|-- / proxy http://127.0.0.1:3000
```

It's a simple syntax, but a powerful tool that we're hoping opens up a lot of possibilities for Tailscale users. We've got some examples of usage ideas for Funnel and `tailscale serve` in our docs; for programmatic uses beyond that, our experimental `tsnet` library can embed Funnel in your Go app.

To learn more about how to set up Funnel on your own tailnet, check out our documentation, or get your hands dirty with a detailed 101-style walk-through on our Tailscale community site about configuring Funnel to receive GitHub webhooks.

Funnel for the whole family

That covers most of the mechanics of setting up and using Funnel on your device, but we know developers will be curious about how it works and what happens on Tailscale's end. The tl;dr: When you turn on Funnel, we create public DNS records for your `node.tailnet.ts.net` name that points to a set of ingress servers we operate around the world, and then we give those servers very limited access to your tailnet — just enough to offer a TCP connection, which your nodes can accept or reject. We go into a lot more detail in our Funnel announcement post, and you can always find more information in the Funnel section of our knowledge base.

What's new since alpha

We've learned a lot since launching Funnel as an invite-only alpha feature last November, and thanks to great feedback from alpha testers, we've been able to make a few key improvements.

DNS

> It's not DNS
> There's no way it's DNS
> It was DNS
>
> – networking haiku

Throughout the alpha period, we've been improving the way we handle DNS requests and how we configure public records.

Tailscale offers MagicDNS to resolve DNS requests for the names of nodes on your tailnet. This is a handy feature since remembering names (`amelie.pango-lin.ts.net`) is easier than remembering CGNAT IP addresses (`100.x.y.z`). Since launching Funnel alpha, we now need to forward `*.ts.net` DNS requests to a public resolver. (Before Funnel, we wouldn't resolve DNS queries from outside your current tailnet for `ts.net` names.)

In addition to splitting DNS, we need a system to track active Funnel nodes and create or update their DNS records to point at the nearest geo-local ingress server. We have a number of ingress servers across the globe that handle inbound connections from the

internet and forward them over a secure tunnel to your node. We pick the closest (by latency) ingress server to your Funnel-enabled node. We also keep it up to date in case you're on the go!

The whole process from enabling Funnel to assigning the right public DNS record revealed a number of opportunities for improving the way we send updates across tailnets. We need to coordinate with the control plane to inform the ingress servers which in turn kick off a DNS process and send an update to your node. We sometimes call this "waking up nodes," and since the alpha launch, we've fixed a few bugs and optimized the number of updates.

CLI

One of the best ways to test a user interface is to use it! For our alpha release we released the initial set of CLI commands, and we've received feedback from the community (and employees!) on how we could improve the ergonomics and better separate Funnel from the equally useful `tailscale serve` command.

In Tailscale v1.38.1, we released the overhauled set of commands. We've updated how you start local servers via `tailscale serve`, and we've separated Funnel into its own command, `tailscale funnel`.

In updating the CLI commands we tried to find the right balance between clarity and brevity. Some commands ended up being shorter than their equivalent with the old interface — and we hope you agree that the new syntax is just as clearly understood.

Here are a few examples of before and after:

```
# Start an HTTPS server on port 443, forwarding to a local server running at htt

# both: default to port 443 for HTTPS
(OLD) $ tailscale serve / proxy http://localhost:3000
(NEW) $ tailscale serve https / http://localhost:3000

# The same, but using an alternate port 8443
(OLD) $ tailscale serve --serve-port=8443 / proxy http://localhost:3000
(NEW) $ tailscale serve https:8443 / http://localhost:3000

# Start an HTTPS server on port 443, serving files from /home/amelie/docs
(OLD) $ tailscale serve / path /home/amelie/docs
(NEW) $ tailscale serve https / /home/amelie/docs

# Forward incoming TCP connections on port 10000 to a local TCP server on port 2
# (e.g. to run OpenSSH in parallel with Tailscale SSH):
(OLD) $ tailscale serve --serve-port=10000 tcp 22
(NEW) $ tailscale serve tcp:10000 tcp://localhost:22

# You can now use any valid port for a `tailscale serve` command.
# With the old CLI it was restricted to valid Funnel ports (443, 8443, 10000)

(OLD) $ tailscale serve --serve-port=222 tcp 22 # ERROR!
(NEW) $ tailscale serve tcp:2222 tcp://localhost:22
```

Now in beta

Tailscale Funnel is available in beta, which means you no longer need an invite to use it, but you will need to enable it in your tailnet's settings, and you'll need to be running Tailscale 1.38.3 or later. If you want to share how you're using Funnel or other Tailscale features, we'd love to hear from you at devrel@tailscale.com!

Share via

← Back to index

YouTube

tailscale

tailscale

Privacy & Terms