-ing into

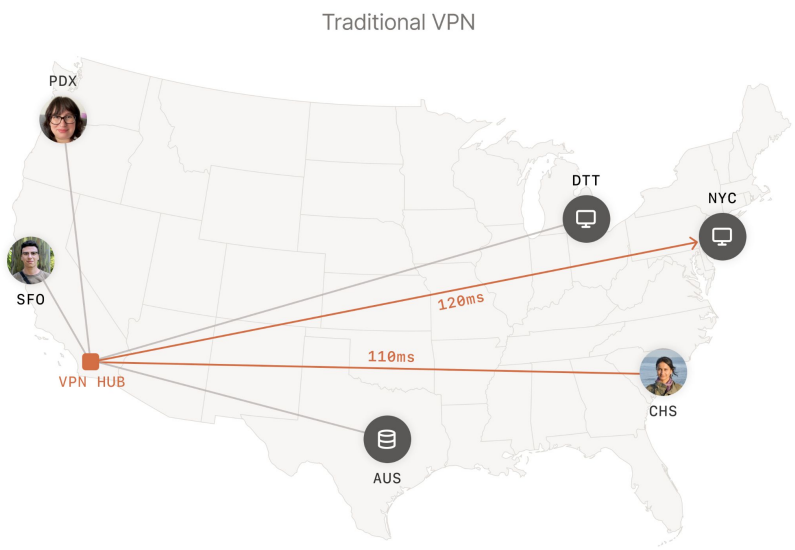
Production over

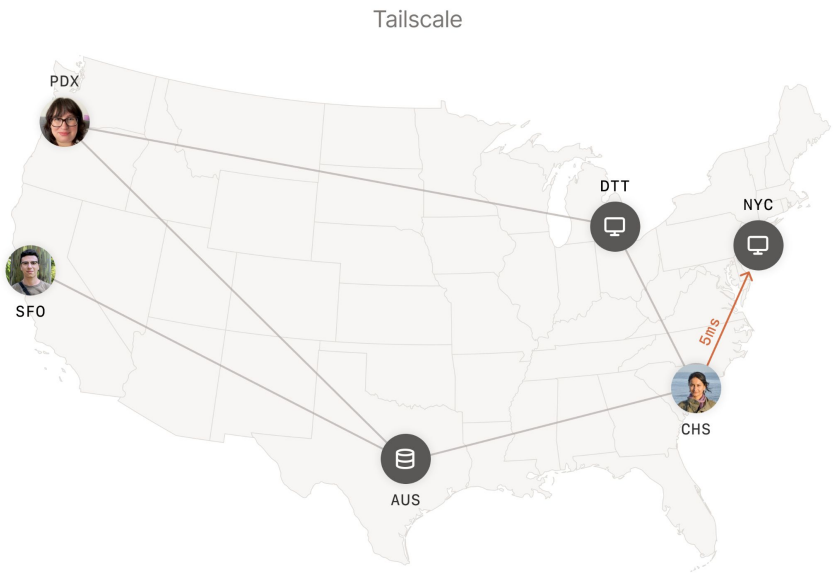
Agenda

- What Tailscale
- What Teleport
- Tailscale + Teleport to securely access infrastructure

tailscale

What Tailscale?



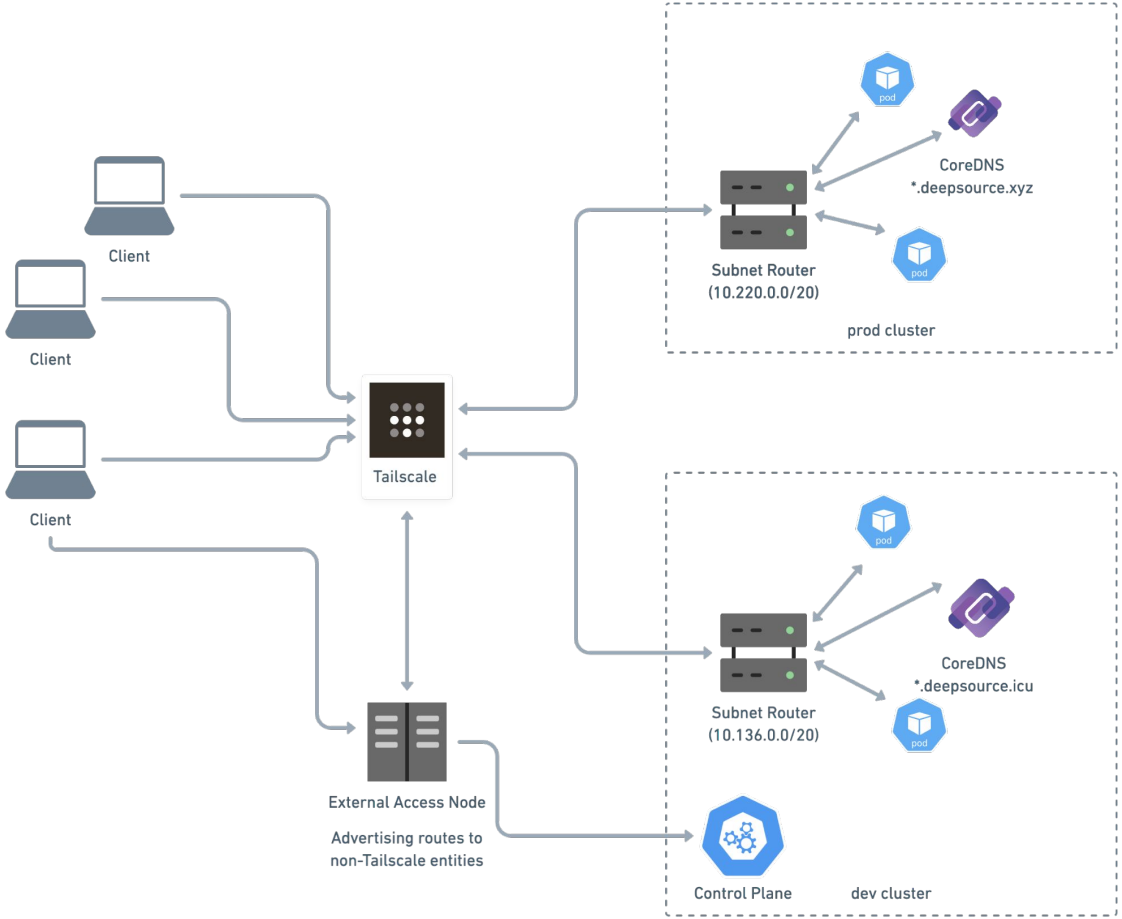


Source – https://tailscale.com/kb/1151/what-is-tailscale/

Why Tailscale?


LINUS
WIREGUARD
WIREGUARD
OPENVPN
OPENVPN
OPENVPN

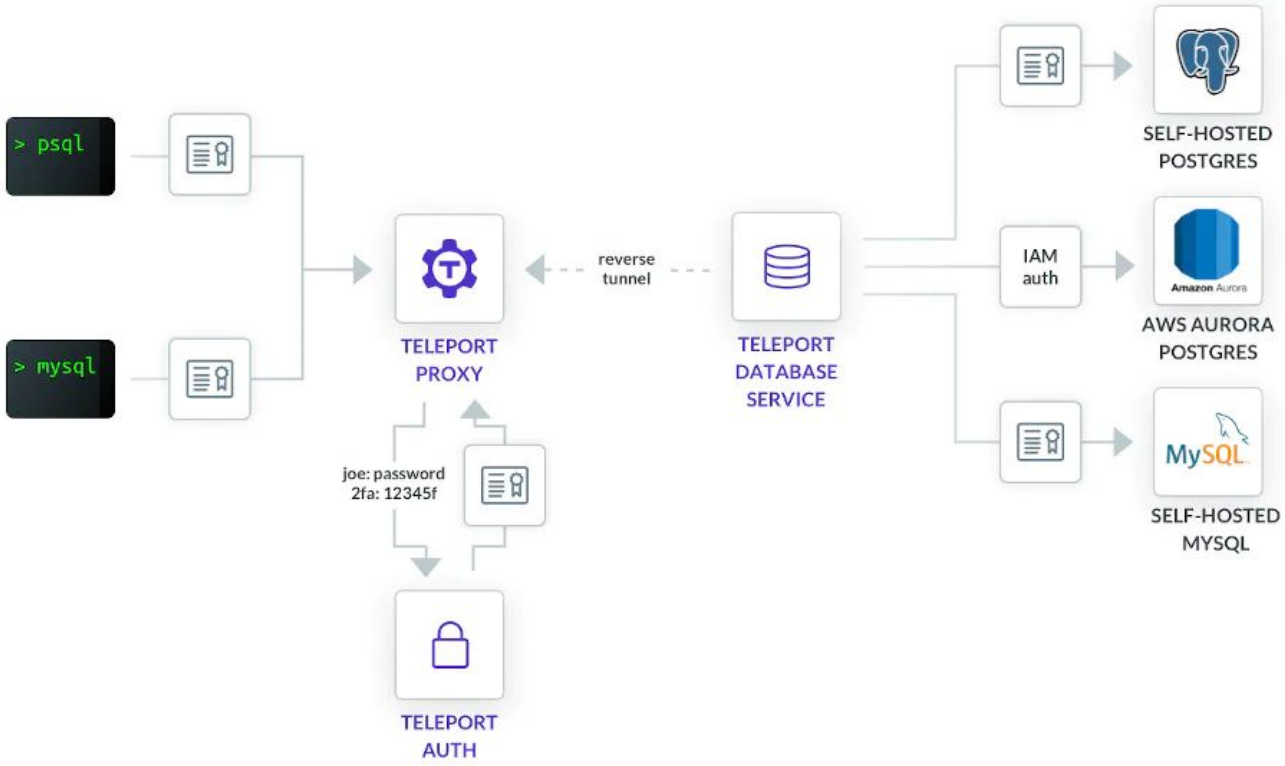
Tailscale @ DeepSource



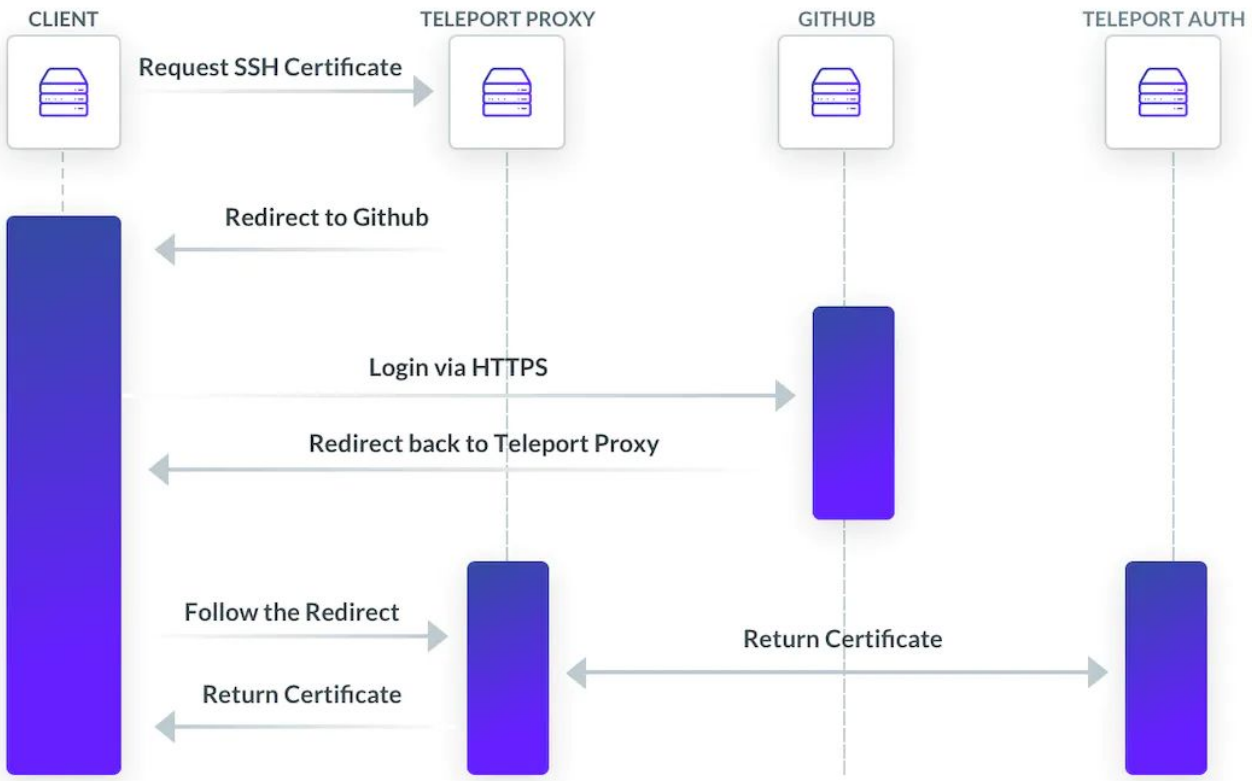
Source – https://deepsource.com/blog/tailscale-at-deepsource/

Teleport

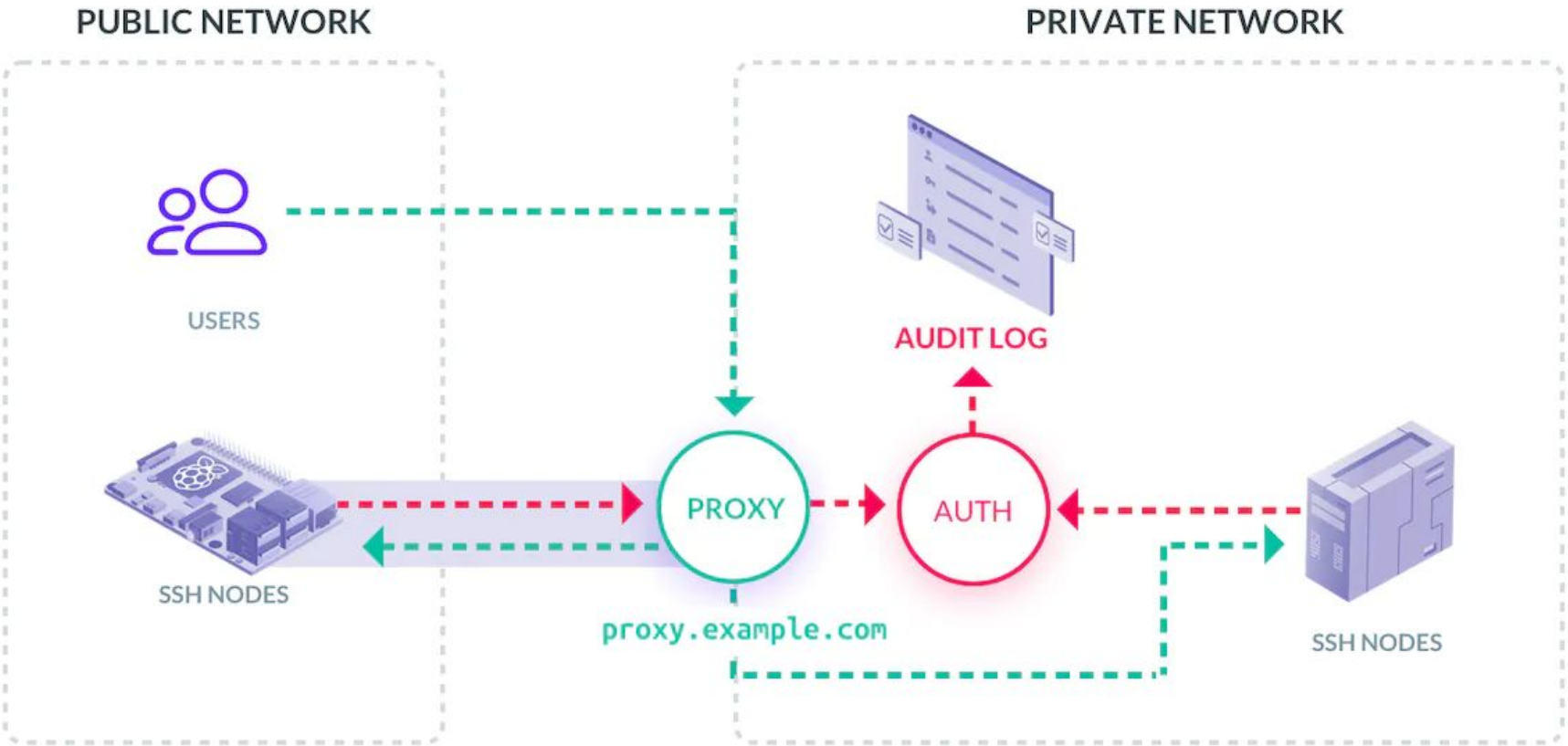
What Teleport?



Source – https://goteleport.com/how-it-works/database-access/



Source – https://goteleport.com/how-it-works/certificate-based-authentication-ssh-kubernetes/



Source – https://goteleport.com/how-it-works/certificate-based-authentication-ssh-kubernetes/

Am I a part-time marketer for Tailscale and Teleport?

OVER

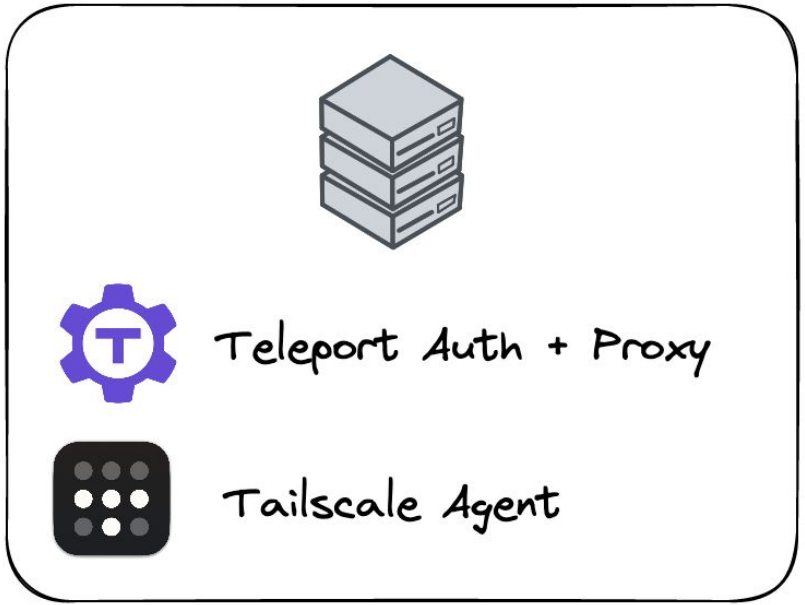
Requirements

To create a minimal Teleport cluster, you must launch three services:

- **Teleport Auth Service**. The certificate authority of the cluster. It issues certificates to clients and maintains the audit log.
- **Teleport Proxy Service**. The proxy allows access to cluster resources from the outside. Typically it is the only service available from the public network.
- **Teleport agents**. A Teleport agent runs in the same network as a target resource and speaks its native protocol, such as the SSH, Kubernetes API, HTTPS, PostgreSQL, and MySQL wire protocols. Think of a "smart sidecar" that routes user requests to its target resource.

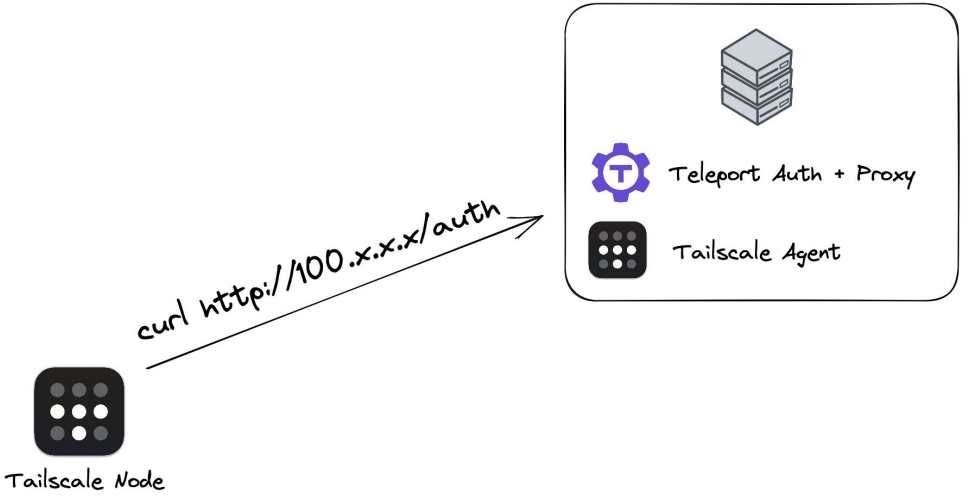
Source – https://goteleport.com/how-it-works/

Step 1 – Auth+Proxy Server


Teleport Auth + Proxy
Tailscale Agent

Step 2 – Talking to the Auth Server

Simple enough, right!?



Well, turns out

Prerequisites

- A Linux host with only port `443` open to ingress traffic. You must be able to install and run software on the host. Either configure access to the host via SSH for the initial setup (and open an SSH port in addition port `443`) or enter the commands in this guide into an Amazon EC2 user data script, Google Compute Engine startup script, or similar.

  WARNING

  This guide is not intended for local environments, e.g., a Docker container on your workstation. For guides to trying out a containerized Teleport deployment locally, see the following:

  - Local Kubernetes Cluster
  - Docker Compose
  - Single Docker Container

- A two-factor authenticator app such as Authy, Google Authenticator, or Microsoft Authenticator
- `python3` installed on your Linux host. We will use this to run a simple HTTP file server, so you can use another HTTP server if you have one installed.

You must also have one of the following:

- A registered domain name.
- An authoritative DNS nameserver managed by your organization, plus an existing certificate authority. If using this approach, ensure that your browser is configured to use your organization's nameserver.

Source: https://goteleport.com/docs/try-out-teleport/linux-server/


DANC

MagicDNS

The Hero we didn’t deserve but needed

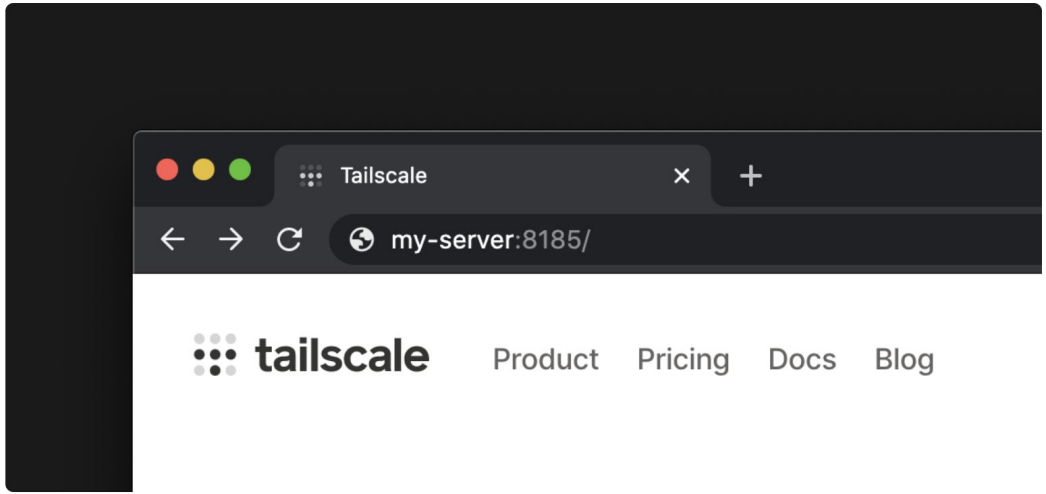
Docs › How-to Guides

MagicDNS

MagicDNS automatically registers DNS names for devices in your network.

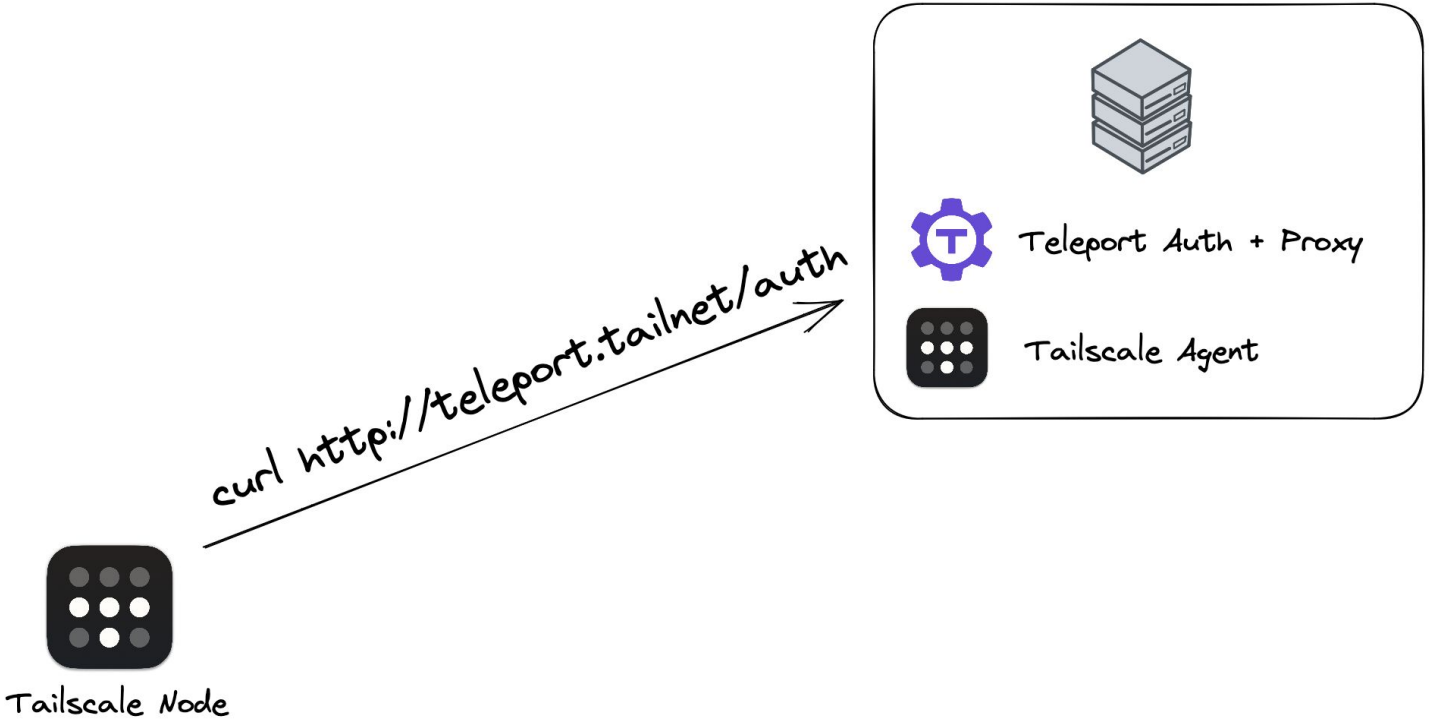
MagicDNS is available for **all plans**.

If you add a new webserver called `my-server` to your network, you no longer need to use its Tailscale IP: using the name `my-server` in your browser's address bar or on the command line will work.



Source: https://tailscale.com/kb/1081/magicdns/

Cool, now this should work, right?



Well ...

Prerequisites

- A Linux host with only port `443` open to ingress traffic. You must be able to install and run software on the host. Either configure access to the host via SSH for the initial setup (and open an SSH port in addition port `443`) or enter the commands in this guide into an Amazon EC2 user data script, Google Compute Engine startup script, or similar.

  WARNING

  This guide is not intended for local environments, e.g., a Docker container on your workstation. For guides to trying out a containerized Teleport deployment locally, see the following:

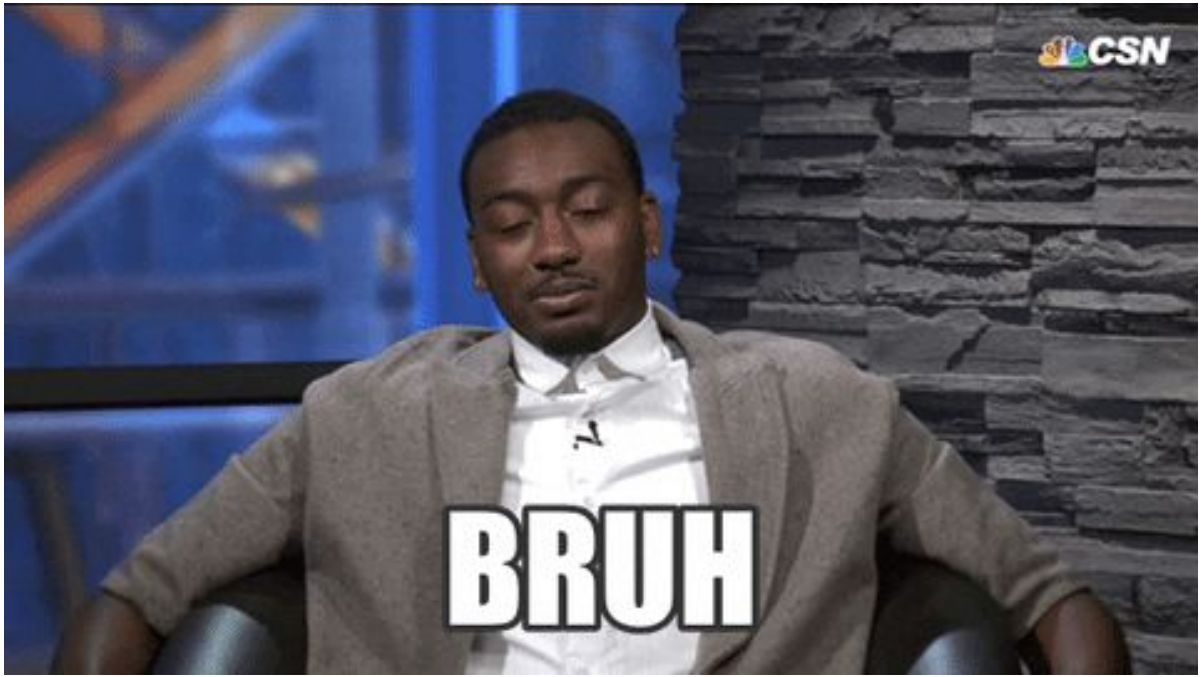
  - Local Kubernetes Cluster
  - Docker Compose
  - Single Docker Container

- A two-factor authenticator app such as Authy, Google Authenticator, or Microsoft Authenticator
- `python3` installed on your Linux host. We will use this to run a simple HTTP file server, so you can use another HTTP server if you have one installed.

You must also have one of the following:

- A registered domain name.
- An authoritative DNS nameserver managed by your organization, plus an existing certificate authority. If using this approach, ensure that your browser is configured to use your organization's nameserver.

Source: https://goteleport.com/docs/try-out-teleport/linux-server/


CSN
BRUH

Thankfully

Docs › How-to Guides

Enabling HTTPS

Connections between Tailscale nodes are secured with end-to-end encryption. Browsers, web APIs, and products like Visual Studio Code are not aware of that, however, and can warn users or disable features based on the fact that HTTP URLs to your tailnet services look unencrypted since they're not using TLS certificates, which is what those tools are expecting.

To protect a website with an HTTPS URL, you need a TLS certificate from a public Certificate Authority (CA).

This feature uses the active tailnet name for your tailnet.

> This feature is currently in beta. To try it, follow the steps below to enable it for your network using Tailscale v1.14 or later.

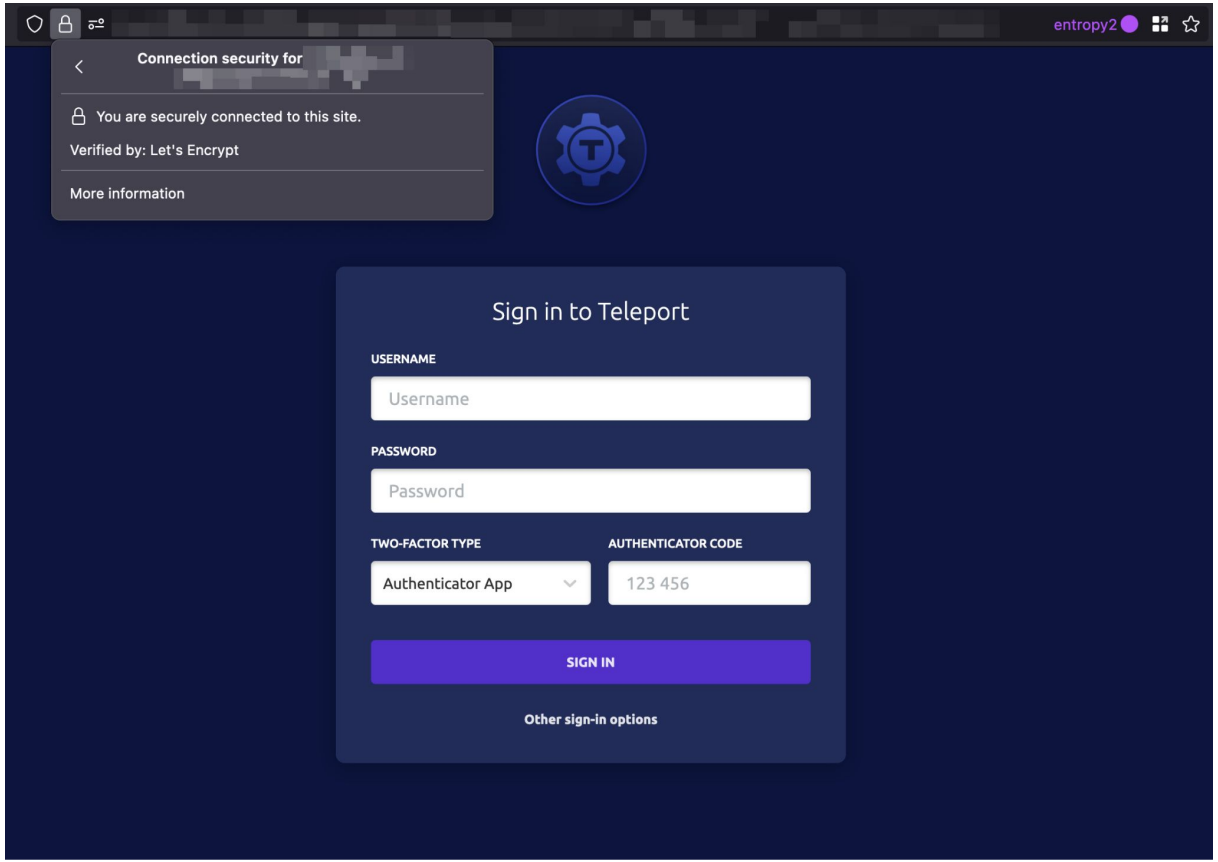
Configure HTTPS

To be able to provision TLS certificates for devices in your tailnet, you need to:

1. Navigate to the **DNS** page of the admin console.
2. Enable MagicDNS if not already enabled for your tailnet.
3. Under **HTTPS Certificates**, click **Enable HTTPS**.
4. Acknowledge that your machine names and your tailnet name will be published on a public ledger.
5. For each machine you are provisioning with a TLS certificate, run `tailscale cert` on the machine to obtain a certificate.

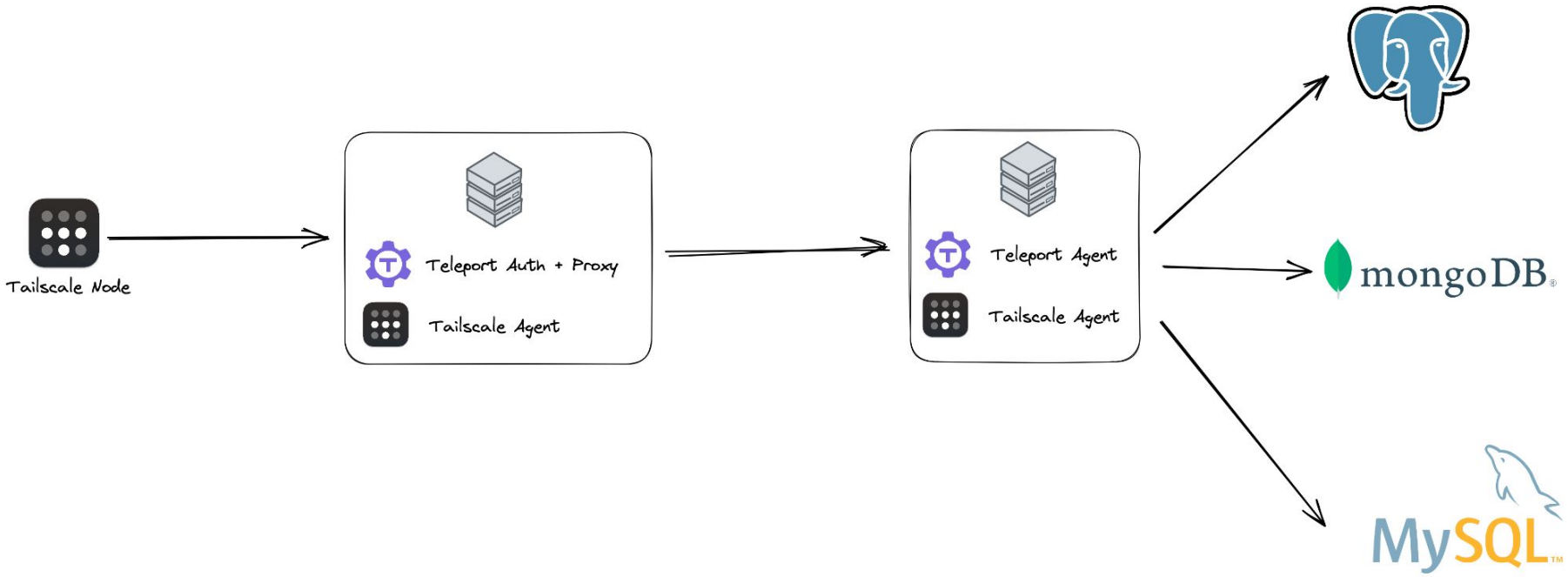
Source: https://tailscale.com/kb/1153/enabling-https/

Et Voilà!


entropy2
Connection security for
You are securely connected to this site.
Verified by: Let's Encrypt
More information
Sign in to Teleport
USERNAME
Username
PASSWORD
Password
TWO-FACTOR TYPE
Authenticator App
AUTHENTICATOR CODE
123 456
SIGN IN
Other sign-in options

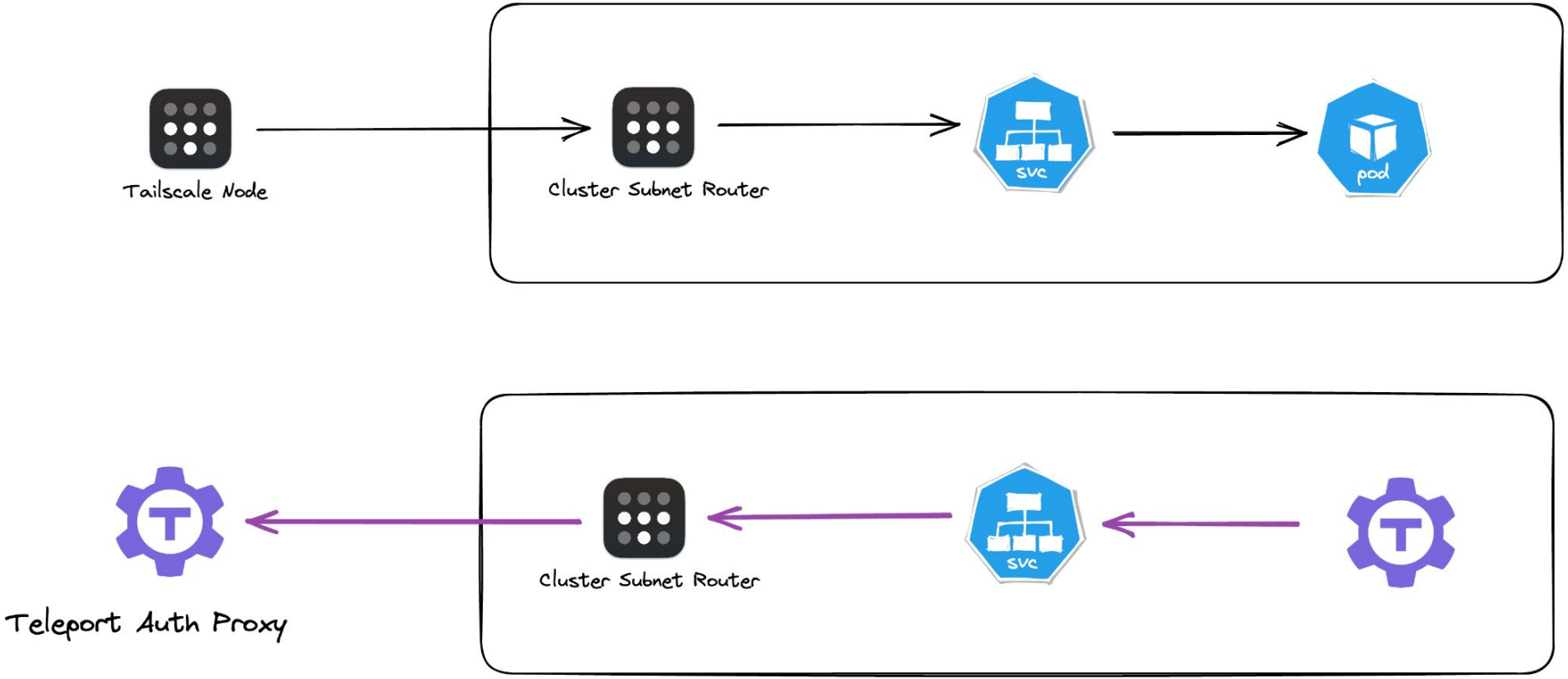
Step 3 - Connecting Infrastructure

Databases

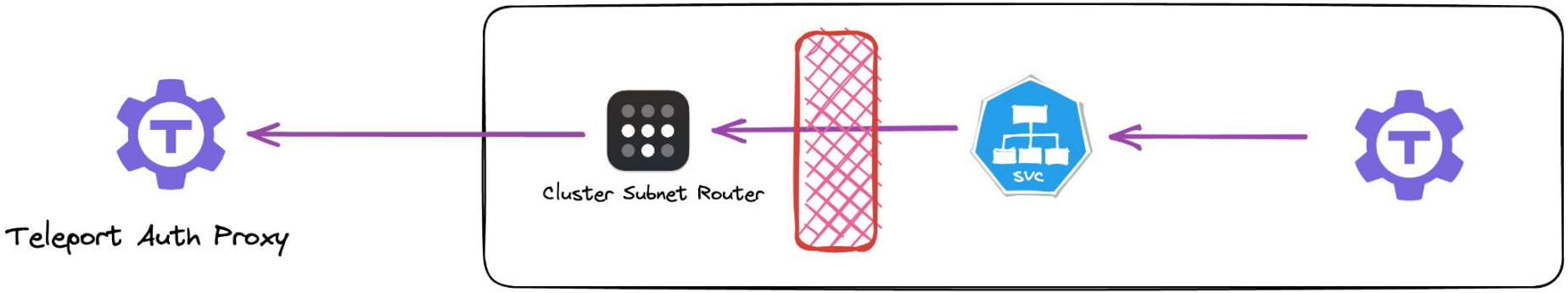

Tailscale Node
Teleport Auth + Proxy
Tailscale Agent
Teleport Agent
Tailscale Agent
mongoDB
MySQL

Kubernetes

Original Idea



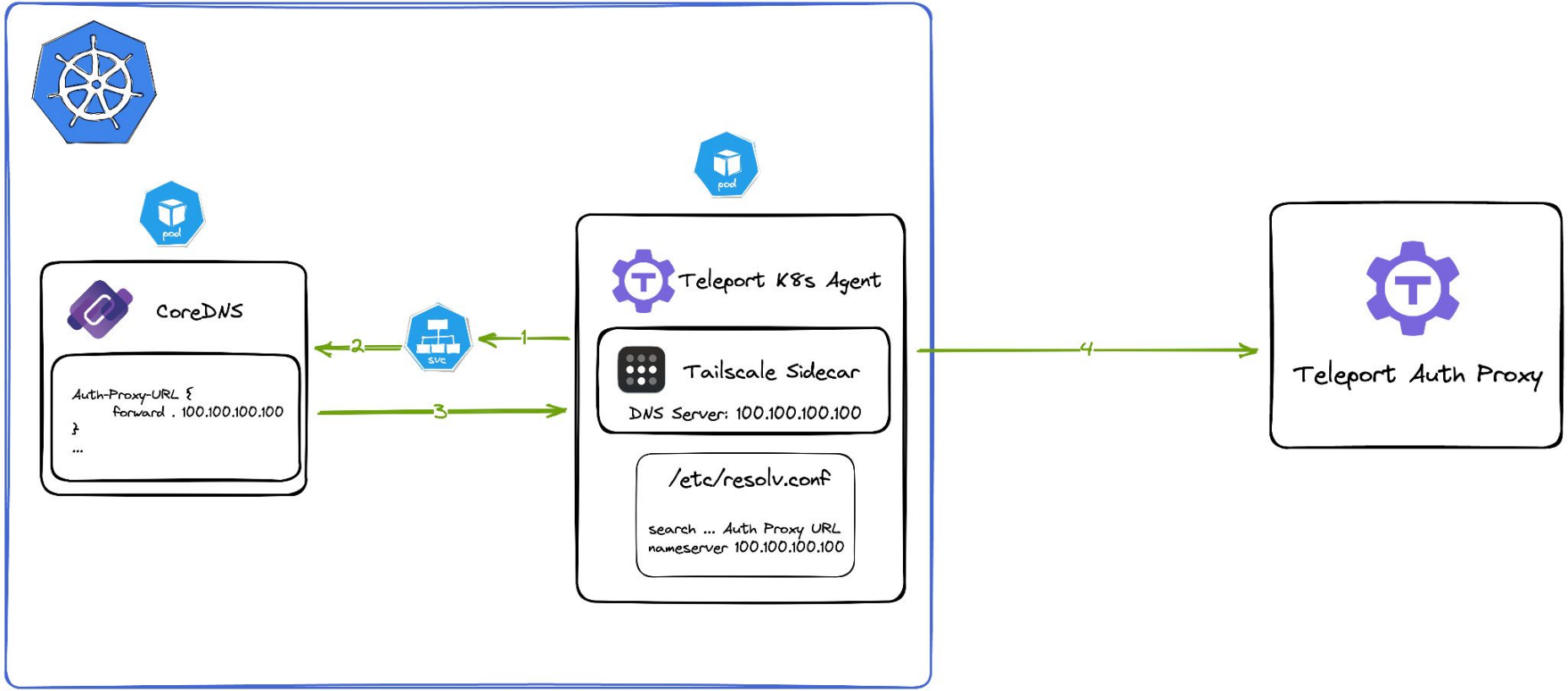
Which doesn’t really work



The Subnet Router doesn't listen for requests, it's not bi-directional


peacock

The Solution



Questions!?