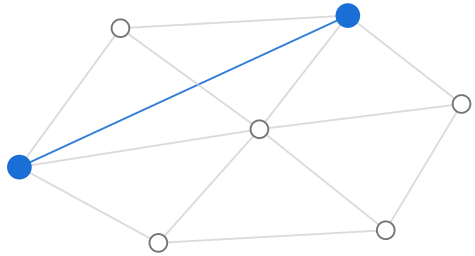
cecil.wöbker × remberg

CONTAINERDAYS HAMBURG — 2 SEPTEMBER 2026

Beyond Traditional VPNs

WireGuard mesh networking for Kubernetes with Tailscale

Cecil Wöbker
Co-Founder & CTO, remberg
@cwoebker



How we got here

BEFORE KUBERNETES

Dedicated VPN & Servers

SSH keys, revoked by hand

AFTER THE MIGRATION

AWS Client VPN & Cloud

Unstable, and a one-way door

What it cost us

Certificates expire, and someone has to rotate them

One concentrator server that can go down.

On the network or off it, with nothing in between

Two shapes of private access

HUB-AND-SPOKE

Every packet through a concentrator:

- One box to size & patch
- A latency detour via the hub
- Access network-shaped: in or out

MESH

Peers talk directly, encrypted E2E:

- No concentrator to run
- Direct paths, NAT traversal handled
- Access is identity-shaped

Beyond Traditional VPNs

BEFORE KUBERNETES

Dedicated VPN & Servers

SSH keys, revoked by hand

AFTER THE MIGRATION

AWS Client VPN & Cloud

Unstable, and a one-way door

TODAY

WireGuard mesh

Access defined in Git

Enter the Tailscale Kubernetes Operator

It watches Services and turns them into mesh nodes. Two ways to ask it.

THE SHORTCUT

```yaml
metadata:
  annotations:
    tailscale.com/expose: "true"
```

One line on the Service you already have.

WHAT WE RUN

```yaml
metadata:
  annotations:
    tailscale.com/hostname: grafana-prod
spec:
  type: LoadBalancer
  loadBalancerClass: tailscale
```

DNS names you choose:

```
grafana-prod.<tailnet>
argo-prod.<tailnet>
```

The mesh doesn't stop at the cluster edge

One `Connector` puts ElastiCache, Amazon MQ and your managed DB on the tailnet.

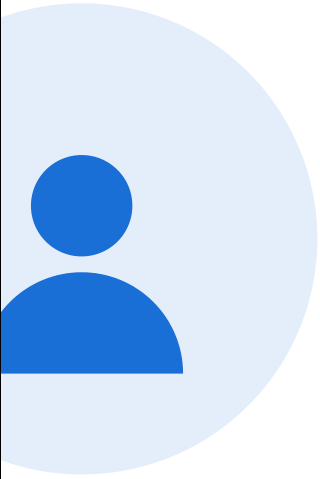
```yaml
apiVersion: tailscale.com/v1alpha1
kind: Connector
metadata:
  name: elasticache-cluster-connector-remberg-env
spec:
  subnetRouter:
    advertiseRoutes:
      - 10.10.10.11/32   # elasticache-env-0001-001
      - 10.10.10.12/32   # elasticache-env-0001-002
```

A route is not a permission. The same `/32`s routes have to be enabled in `policy.hujson`.

Routes are per node, not per subnet.

Identity, not network location

Access follows the person, not the network they are on.



1. SSO groups decide what resolves
2. Revoke a real person, not a certificate
3. Audit trails with names

Access as code

One policy file in the infrastructure repo. The merge is the deploy.

```
// tailscale/policy.hujson
{
  "src": ["group:dev"],
  "dst": ["tag:k8s-operator-dev"],
  "ip":  ["443"],
  "app": { "tailscale.com/cap/kubernetes": [
    { "impersonate": { "groups": ["tailnet-readers"] } }
  ]},
}
```

CI tests effective access on the PR, applies on merge — the console is read-only

CODEOWNERS gates it, so widening access needs a review

Need more? Open a PR. `git log` is the audit trail

Access an agent can use

An agent on my laptop queries dev logs. Three layers, each defined in code.

- **Tailscale** — the path to `grafana-dev.<tailnet>`, from `policy.hujson`
- **AWS IAM** — `GetSecretValue` on exactly one secret ARN, from CDK
- **Grafana** — a Viewer service-account token, read-only

Dev environment only of course.

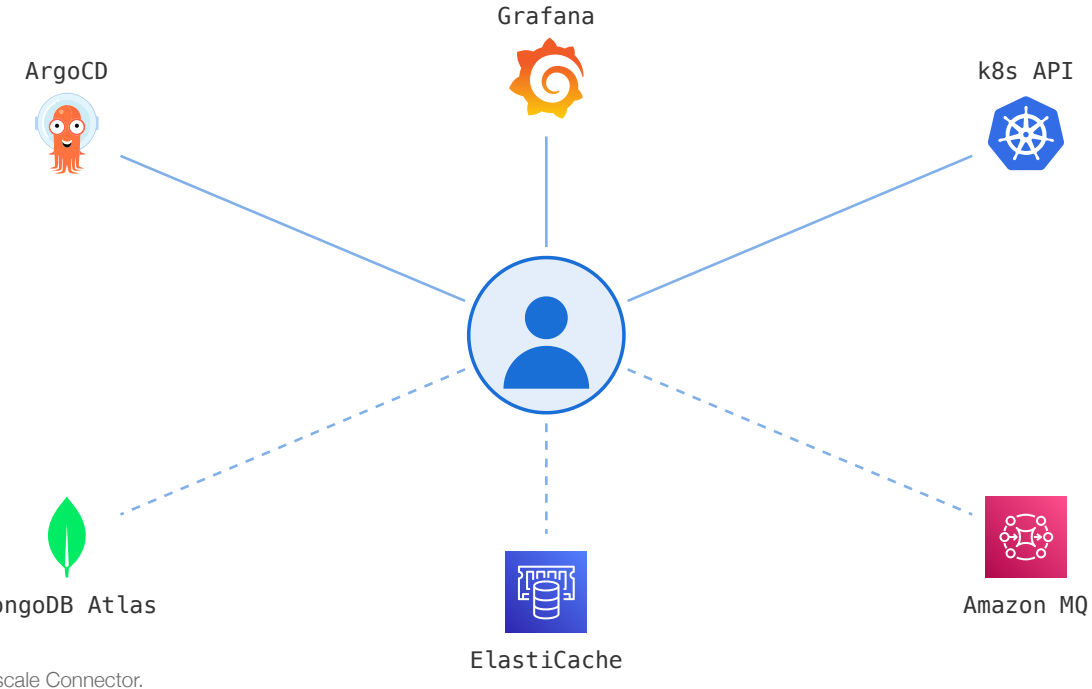
Going fully open source

If you want none of this on a vendor, almost every piece already has a replacement.

- **Client** — already BSD-licensed. Point it at your own server.
- **Control plane** — Headscale, self-hosted, never in the data path
- **Identity** — any OIDC provider: Keycloak, Authentik, Authelia
- **Relays** — Headscale ships its own DERP server
- **Policy** — the same HuJSON file, applied with `headscale policy set`

The operator is the gap. It wants Tailscale's v2 API and an OAuth client; Headscale has neither, by decision. Run a sidecar per workload.

One identity, everything behind it



Solid — exposed from the cluster.
Dashed — reached through a Tailscale Connector.

thanks.

@cwoebker

cwoebker.com/talks/beyond-traditional-vpns