Headscale

This supersedes the original guide. Changes from that version, all based on what you actually have running:

1. Caddy is a **native systemd service on the host**, not a Docker container — the original guide's `caddy` service block in `docker-compose.yml` and the Docker-network Caddyfile no longer apply.
2. `server_url` had the internal port (`:8080`) wrongly appended — fixed.
3. `trusted_proxies` was missing entirely — added, using `127.0.0.1/32` (correct because Caddy reaches Headscale via host loopback, not a Docker bridge network — that CIDR would only have been right if Caddy were a container in the same Compose stack, which it isn't).
4. `dns.base_domain` was left as the placeholder `example.com` — fixed to a subdomain distinct from `server_url`'s domain, per Headscale's own requirement that these must differ.

Version basis: this is checked against Headscale's current `main`-branch `config-example.yaml` and the official reverse-proxy docs, consistent with the `v0.28.0` pin from the original compose file. I don't have confirmation of which version is actually running in your container — run `docker exec headscale headscale version` and tell me if it's different, since config field names have moved between sections in past releases.

---

0. DNS — what actually needs a real domain record

Only **one** public DNS record is needed: an A (and AAAA, if your VPS has IPv6) record for **`headscale.seabee.me`**, pointing at the VPS's public IP. This is the domain Caddy serves over real HTTPS, the one Let's Encrypt validates via HTTP-01, and the one used in `server_url`.

`ts.seabee.me` (the `dns.base_domain` value) needs no DNS record at all. It's never queried against public DNS or your registrar — it's a string Headscale uses internally to construct MagicDNS device names (`hostname.ts.seabee.me`), resolved only by Headscale's own embedded DNS server, which Tailscale clients query via `100.100.100.100`. It doesn't need to exist anywhere except in this config file.

It still has to be a domain you own and aren't using for anything else, though — not because it needs to resolve anywhere, but because if you pick a domain you're actively using for real services, MagicDNS will hijack every query under that domain for any device on the tailnet and return

NXDOMAIN instead of the real record, confirmed by a filed upstream Headscale bug report showing exactly this happening to someone's real subdomain. Since `ts.seabee.me` isn't serving anything else, this doesn't affect you — just don't reuse it for a real service later without remembering why.

In the Caddyfile (Section 4 below), only `headscale.seabee.me` appears — that's correct and intentional. `ts.seabee.me` never touches Caddy.

---

1. Folder structure

You already created this. The `caddy/` subfolders (`caddy/data`, `caddy/config`) from the original `mkdir` are no longer needed since Caddy isn't running in this stack — leave them (harmless) or `rm -rf caddy/` from `~/Docker/headscale/`, your call.

Structure actually in use now:

```
~/Docker/headscale/
├── docker-compose.yml
├── headscale/
│   ├── config/config.yaml
│   ├── lib/
│   └── run/
└── headplane/
    ├── config/config.yaml
    └── lib/
```

---

2. `headscale/config/config.yaml` — corrected

Edit your existing file. The specific lines to change:

```
# WAS: server_url: https://headscale.seabee.me:8080
server_url: https://headscale.seabee.me

listen_addr: 0.0.0.0:8080          # correct as-is — Docker needs this bound inside the container
metrics_listen_addr: 127.0.0.1:9090
```

```
grpc_listen_addr: 127.0.0.1:50443
grpc_allow_insecure: false

# ADD THIS — was missing entirely:
trusted_proxies:
  - 127.0.0.1/32
  - ::1/128

noise:
  private_key_path: /var/lib/headscale/noise_private.key
prefixes:
  v4: 100.64.0.0/10
  v6: fd7a:115c:a1e0::/48
  allocation: sequential
derp:
  server:
    enabled: false
    region_id: 999
    region_code: "headscale"
    region_name: "Headscale Embedded DERP"
    verify_clients: true
    stun_listen_addr: "0.0.0.0:3478"
    private_key_path: /var/lib/headscale/derp_server_private.key
    automatically_add_embedded_derp_region: true
    ipv4: 198.51.100.1
    ipv6: 2001:db8::1
  urls:
    - https://controlplane.tailscale.com/derpmap/default
  paths: []
  auto_update_enabled: true
  update_frequency: 3h
disable_check_updates: false
ephemeral_node_inactivity_timeout: 30m
database:
  type: sqlite
  debug: false
  gorm:
    prepare_stmt: true
    parameterized_queries: true
    skip_err_record_not_found: true
```

```
    slow_threshold: 1000
  sqlite:
    path: /var/lib/headscale/db.sqlite
    write_ahead_log: true
    wal_autocheckpoint: 1000
acme_url: https://acme-v02.api.letsencrypt.org/directory
acme_email: ""
tls_letsencrypt_hostname: ""
tls_letsencrypt_cache_dir: /var/lib/headscale/cache
tls_letsencrypt_challenge_type: HTTP-01
tls_letsencrypt_listen: ":http"
tls_cert_path: ""
tls_key_path: ""
log:
  level: info
  format: text
policy:
  mode: file
  path: ""
dns:
  magic_dns: true
  # WAS: base_domain: example.com  — the untouched placeholder, and it must differ from server_url's domain
  base_domain: ts.seabee.me
  override_local_dns: true
  nameservers:
    global:
      - 1.1.1.1
      - 1.0.0.1
      - 2606:4700:4700::1111
      - 2606:4700:4700::1001
    split: {}
  search_domains: []
  extra_records: []
unix_socket: /var/run/headscale/headscale.sock
unix_socket_permission: "0770"
logtail:
  enabled: false
randomize_client_port: false
taildrop:
  enabled: true
```

`base_domain: ts.seabee.me` is a placeholder suggestion — any subdomain of `seabee.me` other than `headscale.seabee.me` (which `server_url` already uses) works. Pick whatever you actually want your devices' MagicDNS names to end in.

---

3. `docker-compose.yml` — drop the Caddy service

```
services:
  headscale:
    image: headscale/headscale:0.28.0
    container_name: headscale
    restart: unless-stopped
    command: serve
    volumes:
      - ./headscale/config:/etc/headscale
      - ./headscale/lib:/var/lib/headscale
      - ./headscale/run:/var/run/headscale
    ports:
      - "127.0.0.1:8080:8080"
      - "127.0.0.1:9090:9090"
    networks:
      - headscale-net

  headplane:
    image: ghcr.io/tale/headplane:v0.6.1
    container_name: headplane
    restart: unless-stopped
    ports:
      - "127.0.0.1:3000:3000"
    volumes:
      - ./headplane/config/config.yaml:/etc/headplane/config.yaml
      - ./headplane/lib:/var/lib/headplane
      - ./headscale/config/config.yaml:/etc/headscale/config.yaml
      - /var/run/docker.sock:/var/run/docker.sock:ro
    depends_on:
      - headscale
```

```
    networks:
      - headscale-net

networks:
  headscale-net:
    driver: bridge
```

Pin the Headplane tag to whatever's actually current at https://github.com/tale/headplane/releases — `v0.6.1` here is a placeholder, not a verified-current release.

If your existing compose file still has a `caddy:` block from the original guide, delete it — running two things bound to host port 443/80 (native Caddy plus a would-be Caddy container) will conflict, and you don't want the container one anyway now.

---

4. Native Caddy config

I don't have confirmation of your VPS's distro or how Caddy was installed, so I can't state the config path or reload command as fact. **If it's the official Caddy apt repo on Debian/Ubuntu**, the convention is `/etc/caddy/Caddyfile`, reloaded with `sudo systemctl reload caddy` — verify this against your actual install before running it, since a manually-built binary, snap package, or a different distro's package manager may differ.

Caddyfile content (path-independent, this part is safe regardless of how Caddy was installed):

```
headscale.seabee.me {
    handle /admin* {
        reverse_proxy 127.0.0.1:3000
    }
    handle {
        reverse_proxy 127.0.0.1:8080
    }
}
```

Note the target is `127.0.0.1:8080` / `127.0.0.1:3000` — host loopback, not a Docker container hostname like `headscale:8080`. That hostname-based addressing only resolves inside the `headscale-net` Docker bridge network, which your native Caddy process isn't part of.

---

5. `headplane/config/config.yaml`

Unchanged from the original guide, provided you generate a real `cookie_secret` and fill in `api_key` after first boot (step 6 below):

```
server:
  host: "0.0.0.0"
  port: 3000
  cookie_secret: "REPLACE_WITH_32_CHAR_RANDOM_STRING"
  base_url: "https://headscale.seabee.me"

headscale:
  url: "http://headscale:8080"
  config_path: "/etc/headscale/config.yaml"
  api_key: "REPLACE_AFTER_HEADSCALE_STARTS"

integration:
  docker:
    enabled: true
    container_name: "headscale"
```

`headscale.url` stays as `http://headscale:8080` here — that's correct, because this is Headplane-to-Headscale traffic, which stays inside the Docker network (both containers are on `headscale-net`), unlike Caddy which is outside it.

Generate the cookie secret:

```
openssl rand -hex 32
```

---

6. Bring it up

```
cd ~/Docker/headscale
docker compose up -d headscale
docker exec headscale headscale users create myuser
docker exec headscale headscale apikeys create --expiration 90d
```

Paste that API key into `headplane/config/config.yaml` → `headscale.api_key`, then:

```
docker compose up -d
```

Reload Caddy (command depends on your actual install — see section 4). Then visit `https://headscale.seabee.me/admin`.

Revision #3
Created 5 July 2026 00:08:07 by Conor
Updated 5 July 2026 02:55:04 by Conor