# Network Architecture

Details on the physical and logical networking, including the opnSense gateway, internal subnets, and reverse proxy configuration.

# LAN Layout & Gateway

# LAN Layout & Gateway

## □□ Internet Edge

- **ISP:** AT&T Fiber (1Gbps Up/Down)
- **Gateway/ONT:** HUMAX BGW320-500 (Located in Sunroom)
- **Mode:** Bridge/Pass-through mode enabled (February 2026 upgrade)

## □□ Core Routing (Server Closet)

- **Device:** Protectli FW2B (Pluto)
- **OS:** Ubuntu 24.04 LTS (ZFS on Root)
- **Role:** Primary Gateway & Firewall (Replacing opnSense architecture)
- **Internal LAN:** 192.168.13.0/24 (Native), 10.13.x.y (VLANs - Deploying)

### Core Services

- **Firewall:** nftables (v2 production Ruleset active)
- **DHCP:** Technitium DHCP (192.168.13.0/24 scope active)
- **DNS:** Technitium DNS (DNS-over-TLS enabled via Cloudflare/Google)
- **Identity:** JumpCloud (Fully configured on Pluto for identity management)
- **Monitoring:** Beszel-agent + Vector (Shipping to ClickHouse)
- **Backup:** Syncoid/Sanoid (OS snapshots and replication to Luna)
- **VPN:** WireGuard (Pluto Gateway + MacBook Client - Deployed March 18, 2026)

## □□ Physical Switching & Wireless

- **Switch:** 24-port Gigabit Managed Switch with 12x PoE+ ports.
- **AP:** 2x Grandstream Mesh Access Points (PoE powered).

- **Update (Feb 26, 2026):** Backbone migration to the new Ubuntu-based routing on Pluto is complete. Network is stable on 192.168.13.x.

# □□ Security Updates (March 19, 2026)

- **WireGuard:** Tunnel configured for `pluto` and Macbook client; nftables updated for tunnel traffic.
- **Fail2Ban:**
  - Refined `default` and `SSH` jail configs.
  - Added **Recidive** jail (repeat offenders).
  - Added **Portscan** detection jail.
- **Edge Hardening:**
  - **Port Closures:** Removed Minecraft Bedrock (19132) and Bittorrent (52341) forwards.
  - **Rate Limiting:** Enabled `nftables` rate-limiting for Minecraft Java (25565), Voice Chat (24454), and CoTurn (3478).
  - **Geo-blocking:** Implemented automated `nftables` blocklist (CN, KP, RU, IR) via `systemd` timer + lookup sets.

# Reverse Proxy (Caddy)

# Reverse Proxy (Caddy)

Caddy serves as the primary ingress point for all web services in the Flora Family homelab. It handles SSL termination (ACME), local CA management, and request routing.

## □□ Structure

- **Location:** `/srv/caddy`
- **User:** `51004:61004` (Caddy Service Account)
- **Snippets Path:** `/main/appdata/caddy/files/snippets` (Mapped internally to `/etc/caddy/snippets`)

## □□ Log Exporting

- **Mechanism:** UDP/514 forwarding via `rsyslog` on `pluto`.
- **Path:** Raw JSON logs exported to `/var/log/caddy/access.log`.
- **Fail2Ban:** Integrated via local file watcher (jail: `caddy`).

## □□ Caddyfile Strategy

The Caddyfile is organized into modular sites located in `sites-enabled/`. This mimics traditional Apache/Nginx structures for clean management.

### Common Snippets

1. **`security_headers.caddy`**: Implements HSTS, X-Frame-Options (DENY), and nosniff.
2. **`common_tls_internal.caddy`**: Uses `tls internal` for services only accessible on the LAN, backed by the local Caddy CA.
3. **`common_tls_external.caddy`**: Standard ACME/Let's Encrypt for public-facing services.

# Internal CA & Trust

When using `tls internal`, Caddy acts as its own Certificate Authority.

- **Root Cert Location:** `/main/appdata/caddy/data/caddy/pki/authorities/local/root.crt`
- **Permissions:** Access to the `pki` directory requires membership in the **Caddy Group (GID 61004)**.
- **Distribution:** This root certificate is mounted into downstream containers (like SearXNG or OpenWebUI) and added to their system trust stores to allow secure internal communication.

# Typical Route Block

```
searxng.flora.family {
    reverse_proxy searxng:8080
    import /etc/caddy/snippets/security_headers.caddy
    import /etc/caddy/snippets/common_tls_internal.caddy
}
```