

podman

Overview: All 30 containerized services (41 containers total) on `u1` run exclusively under **rootless Podman** (`podman 6.1.4`) managed as native `systemd --user` services via **Podman Quadlets** under the dedicated unprivileged account `docker-data` (`UID 1100` , `GID 1100`). Rootful Docker (`docker.service` , `docker.socket` , and `containerd.service`) is completely disabled and decommissioned, eliminating the privileged root container daemon attack surface while preserving per-service ULA IPv6 reverse-proxy isolation on `eth77` .

1. Architecture & Security

Comparison: Rootless Podman vs. Rootful Docker

Under traditional Docker, `dockerd` and `containerd` execute permanently as `root` (`UID 0`) inside the host init user namespace (`user: [4026531837]`). Every container spawned by `dockerd` without user-namespace remapping shares kernel `UID 0` whenever the container process runs as `root` , mutates host `iptables/nftables` chains directly as `root` , and requires root socket access (`/var/run/docker.sock`) for orchestration or maintenance timers.

Under rootless Podman on `u1` , **there is no root container daemon anywhere on the host:**

- `systemd` starts an unprivileged user manager (`user@1100.service` via lingering) for `docker-data` .
- Each Quadlet `.container` unit directly spawns lightweight `conmon` and `crun` processes inside an unprivileged Linux user namespace (`user: [4026534031]`).
- Shared stateful databases (**PostgreSQL** on `127.0.0.1:5432` and **Valkey** on `127.0.0.1:6379`) run natively on the host locked strictly to loopback, with zero LAN or Docker bridge exposure.

1.1 User Namespace UID/GID Translation Model

Rootless Podman translates container UIDs and GIDs into unprivileged host IDs using `/etc/subuid` and `/etc/subgid` (`docker-data:624288:65536`). Under the default rootless mapping, any container UID `$u` in `[0, 65536]` maps deterministically to a host kernel UID:

$$UID_{host}(u) = \begin{cases} 1100 & \text{if } u = 0 \\ 624287 + u & \text{if } 1 \leq u \leq 65536 \end{cases}$$

Common live worker translations visible in `ps` / `podman-map-tree` on `ul`:

- Container `root` (`u = 0`) $\rightarrow$ Host `UID 1100` (`docker-data`) with zero host capabilities.
- Container `www-data` (`u = 33`, e.g., `nextcloud`, `yourls`, `webrcade`) $\rightarrow$ Host `UID 624320`.
- Container `nginx` (`u = 101`, e.g., `audiomass`, `bentopdf`, `excalidraw`, `penpot-frontend`, `tlldraw`, `wick`) $\rightarrow$ Host `UID 624388`.
- Container `postgres` (`u = 999`, e.g., `penpot-postgres`) $\rightarrow$ Host `UID 625286`.
- Container `authentik` / `degoog` (`u = 1000`) $\rightarrow$ Host `UID 625287`.
- Container `penpot` (`u = 1001`) $\rightarrow$ Host `UID 625288`.

When a binary explicitly refuses to start if internal `getuid() == 0` (specifically `drawpile-srv`), setting `UserNS=keep-id` with `User=1100:1100` preserves `UID 1100` across the namespace boundary:

$$UID_{keep-id}(1100) \stackrel{\text{users}}{\rightarrow} 1100 \stackrel{\text{docker-data}}{\rightarrow} UID_{docker-data}$$

Why Default Rootless UserNS Beats `UserNS=keep-id` for Large ROCm Images: Avoid setting `UserNS=keep-id` unless an application explicitly aborts when `getuid() == 0`. On multi-gigabyte images such as `immich-machine-learning:v3.3.1-rocm` (23.3 GB) or `subocr`, `UserNS=keep-id` triggers a full recursive ownership copy-up across thousands of Python/ROCm files that stalls startup for over 5 minutes. Under the default rootless user namespace, container `UID 0` already maps directly to host `docker-data` (`1100:1100`) with zero layer copy-up and sub-2-second startup.

1.2 Side-by-Side Comparison

Dimension	Legacy Rootful Docker (<code>dockerd</code>)	Current Rootless Podman (<code>docker-data</code> Quadlets)
Daemon Model	Persistent root daemons (<code>dockerd</code> + <code>containerd</code>) running as <code>UID 0</code>	Daemonless fork/exec (<code>user@1100</code> $\rightarrow$ <code>common</code> $\rightarrow$ <code>crun</code>) running as <code>UID 1100</code>
Host Kernel UID of Container <code>root</code>	Host <code>UID 0</code> (<code>root</code> in init <code>user: [4026531837]</code>)	Host <code>UID 1100</code> (<code>docker-data</code> in isolated <code>user: [4026534031]</code>)

Dimension	Legacy Rootful Docker (<code>dockerd</code>)	Current Rootless Podman (<code>docker-data</code> <code>Quadlets</code>)
Host Kernel UID of Container Worker (<code>u > 0</code>)	Same numeric host UID (<code>33</code>) collides with host <code>http</code> (<code>999</code>) collides with system accounts)	Isolated subordinate UID <code>\$624287 + u\$ (624288 -- 689823)</code> , zero host UID collisions)
Service Definition & Lifecycle	<code>docker-compose.yml</code> managed by <code>dockerd</code>	Declarative <code><svc>.container</code> + <code><svc>.network</code> <code>Quadlets</code> in <code>/data/docker/<svc>/</code>
Offline Boot Resilience	Cached locally unless pull policy overridden	Explicit <code>Pull=never</code> on all 41 <code>.container</code> units; 100% WAN-outage-safe boot
Host Firewall Mutation	Mutates host <code>iptables</code> / <code>nftables</code> chains as <code>root</code>	Zero host firewall mutation; unprivileged user sockets bind directly to <code>eth77</code> ULA IPs
Privileged Ports (<code>389</code> / <code>636</code>) LDAP/LDAPS)	Direct root port bind in container + mounted <code>/etc/nginx/certs</code>	Rootless <code>authentik-ldap</code> on <code>:3389</code> / <code>:6636</code> fronted by host HAProxy TCP passthrough (<code>/etc/haproxy/conf.d/ldap.cfg</code>); zero TLS key mounts in containers
Shared Host PostgreSQL & Valkey	Exposed on Docker bridges (<code>172.16.0.0/12</code>) + LAN (<code>10.0.0.0/24</code>)	Locked strictly to <code>127.0.0.1</code> / <code>:::1</code> ; reached from containers via <code>pasta</code> <code>--map-host-loopback 169.254.1.2</code>
ROCm GPU Access (<code>/dev/kfd</code> , <code>/dev/dri</code>)	<code>devices:</code> + <code>group_add</code> as <code>root</code>	<code>AddDevice=</code> + <code>GroupAdd=keep-groups</code> inheriting <code>docker-data</code> 's <code>video</code> (<code>91</code>) and <code>render</code> (<code>991</code>) groups

2. Host Configuration & Intra-Service Networking on `ul`

2.1 Unprivileged Account, Subordinate IDs & Systemd Linging

- Service Account (`docker-data`):**
 - Primary UID/GID: `1100:1100` (`docker-data`), home `/home/docker-data`.
 - Supplementary Groups: `video` (`91`) and `render` (`991`) for AMD Radeon AI PRO R9700 (`gfx1201`) ROCm compute access (`/dev/dri` and `/dev/kfd`).
- Subordinate UID/GID Ranges (`/etc/subuid` & `/etc/subgid`):**

`docker-data:624288:65536`
- Systemd Linging & Automatic Maintenance:**
 - Linging is enabled at `/var/lib/systemd/linger/docker-data` (`sudo loginctl enable-linger docker-data`) so `user@1100.service` starts automatically at boot without an

interactive login.

- Weekly image pruning runs rootlessly via `/etc/systemd/system/podman-prune.{service,timer}` (`User=docker-data`, Sundays at `04:00` EDT/EST), replacing the retired `docker-prune.timer`.

2.2 Global Configuration Files & Storage Layout

- `/etc/containers/containers.conf`: Configures `pasta` (`passt`) as the default rootless network backend and maps the synthetic guest address `169.254.1.2` (`host.containers.internal`) cleanly to host `127.0.0.1` loopback:

```
[network]
default_rootless_network_cmd = "pasta"
pasta_options = ["--map-host-loopback", "169.254.1.2"]
```

- `/etc/containers/registries.conf`:

```
unqualified-search-registries = ["docker.io"]
```

- **Storage Paths:**

- Persistent image/layer store (`GraphRoot`): `/home/docker-data/.local/share/containers/storage` (`overlay` driver owned by `docker-data`, backed by dedicated ZFS dataset `data/podman` mounted at `/home/docker-data/.local/share/containers` with `compression=lz4`, `xattr=sa`, `acltype=posix`, and `atime=off` so container images never consume space on the `/` NVMe).
- Ephemeral runtime state & DNS tables (`RunRoot`): `/run/user/1100/containers` (including `/run/user/1100/containers/networks/aardvark-dns/`).
- Quadlet unit directory: `/home/docker-data/.config/containers/systemd/` (symlinks to `/data/docker/<service>/*. {container,network}`).
- Per-service persistent data & definitions: `/data/docker/<service>/` on the ZFS storage pool.

2.3 The Four Intra-Service Networking Patterns on `u1`

All container traffic on `u1` follows four clean, deterministic patterns without shared mega-bridges like the retired `infra_net`:

1. **Ingress (`Nginx` $\rightarrow$ **Container ULA IPv6 Socket**):** Host Nginx (`UID 33` `http`) terminates public HTTPS (`*.drheck.dev`) and proxies upstream HTTP/WebSocket traffic to each service's dedicated `/64` ULA IPv6 address on `eth77`:

```
$$ \text{Client HTTPS} \longrightarrow \text{Host Nginx } (\text{UID 33}) \\ \longrightarrow [\text{fd77:7777:5010:0:1337:f1ac:d00d:hex}]:P_{\text{host}} \\ \longrightarrow \text{Rootless Container } (\text{UID 1100}) \$\$
```

2. **Container $\longrightarrow$ Native Host Daemons (PostgreSQL , Valkey , Ollama)**
via 169.254.1.2: Native postgresql.service (127.0.0.1:5432), valkey.service (127.0.0.1:6379), per-service ACL users), and ollama.service (127.0.0.1:11434) listen exclusively on host loopback. Inside any rootless container, pasta translates 169.254.1.2 directly to host 127.0.0.1:
- ```
$$ \text{Container App} \longrightarrow \text{169.254.1.2}:(5432, 6379, 11434) \\ \stackrel{\text{pasta}}{\longrightarrow} \text{Host } 127.0.0.1:(5432, 6379, 11434) \$\$
```

3. **Container  $\longrightarrow$  Host OIDC / WOPI / LDAPS Hairpin ( AddHost=auth.drheck.dev:169.254.1.2 ):** Inside a rootless pasta network namespace, connecting to an IP address owned by the host itself ( 10.0.0.100 , 10.0.0.3 , or a host fd77:... ULA address) loops back inside the container namespace ( Connection refused ) rather than hairpinning out to host Nginx or HAProxy. Every OIDC, WOPI, or LDAPS client container therefore declares:

```
AddHost=auth.drheck.dev:169.254.1.2
```

(plus AddHost=cloud.drheck.dev:169.254.1.2 and AddHost=office.drheck.dev:169.254.1.2 on Nextcloud + Collabora). This preserves full Let's Encrypt TLS SNI and hostname verification while routing cleanly over loopback to host Nginx ( 127.0.0.1:443 ) and HAProxy ( 127.0.0.1:636 ).

4. **Multi-Container Private Stack Bridges ( <service>.network + aardvark-dns ) & Privileged Port Proxy ( HAProxy ):**
- Multi-container stacks ( authentik , immich , penpot , nextcloud + collabora , bookstack , romm , yourls , wipstudio , webshell + step-ca ) each have a dedicated isolated bridge ( 10.89.x.0/24 ) with container-name DNS served by /usr/lib/podman/aardvark-dns . Database sidecars ( bookstack\_db , romm-db , yourls-db , penpot-postgres , penpot-redis ) publish **zero** host ports.
  - Because unprivileged UID 1100 does not bind privileged ports < 1024 , authentik-ldap listens on [fd77:...:aaaa]:3389 (LDAP) and :6636 (LDAPS), and host HAProxy ( /etc/haproxy/conf.d/ldap.cfg ) performs zero-copy Layer-4 TCP passthrough from 127.0.0.1 , 10.0.0.3 , 10.0.0.100 , and [fd77:...:aaaa] ports 389 / 636 to [fd77:...:aaaa]:3389 / 6636 .

---

## 3. Complete Rootless Podman Service Inventory (30 Services / 40

# Containers)

| Service                 | Systemd Unit(s) ( <code>docker-data@</code> )                                                                         | Network                    | Published Host Binding(s)                                                                                                  | Backing Store / Notes                                                                              |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <code>anchor</code>     | <code>anchor.service</code>                                                                                           | <code>anchor_net</code>    | <code>[fd77:...:c4a1]:3000 -&gt; 3000</code>                                                                               | Host PG ( <code>169.254.1.2:543</code> )                                                           |
| <code>audiomass</code>  | <code>audiomass.service</code>                                                                                        | <code>podman</code>        | <code>[fd77:...:a0d1]:9004 -&gt; 80</code>                                                                                 | Stateless WebAudio edito                                                                           |
| <code>authentik</code>  | <code>authentik-server.service</code><br><code>authentik-worker.service</code><br><code>authentik-ldap.service</code> | <code>authentik_net</code> | <code>[fd77:...:aaaa]:9000,9443 (internal worker)</code><br><code>[fd77:...:aaaa]:3389,6636 (via HAProxy :389,:636)</code> | Host PG + Host Valkey ( <code>169.254.1.2:636</code> )<br>TLS key mounts                           |
| <code>bentopdf</code>   | <code>bentopdf.service</code>                                                                                         | <code>podman</code>        | <code>[fd77:...:bdf]:8080 -&gt; 8080</code>                                                                                | Stateless PDF toolkit                                                                              |
| <code>bookstack</code>  | <code>bookstack.service</code><br><code>bookstack_db.service</code>                                                   | <code>bookstack_net</code> | <code>[fd77:...:b00c]:6875 -&gt; 80</code><br><i>(internal MariaDB 3306)</i>                                               | MariaDB sidecar + OIDC                                                                             |
| <code>cam</code>        | <code>cam.service</code>                                                                                              | <code>cam_net</code>       | <code>[fd77:...:ca30]:8000 -&gt; 8000</code> , <code>[127.0.0.1:8089 -&gt; 8000</code>                                     | <code>/proc/net/arp:ro</code> , YOLOv5<br>Baichuan/RTSP                                            |
| <code>collabora</code>  | <code>collabora.service</code>                                                                                        | <code>nextcloud_net</code> | <code>[fd77:...:7a31]:9980 -&gt; 9980</code>                                                                               | Strict <code>c:\cloud\.drcheck\.dev</code>                                                         |
| <code>degoog</code>     | <code>degoog.service</code>                                                                                           | <code>degoog_net</code>    | <code>[fd77:...:de90]:4444 -&gt; 4444</code>                                                                               | Host Valkey ( <code>db 4</code> ), update<br><code>/usr/local/bin/degoog-up</code>                 |
| <code>demucs</code>     | <code>demucs.service</code>                                                                                           | <code>podman</code>        | <code>[fd77:...:de3]:8000 -&gt; 8000</code>                                                                                | ROCm GPU ( <code>/dev/dri</code> , <code>/dev/shm</code> ,<br><code>GroupAdd=keep-groups</code> )  |
| <code>drawpile</code>   | <code>drawpile.service</code>                                                                                         | <code>podman</code>        | <code>[fd77:...:d7a0]:27750-27752</code>                                                                                   | <code>UserNS=keep-id</code> , <code>User=116</code><br>sessions                                    |
| <code>excalidraw</code> | <code>excalidraw.service</code>                                                                                       | <code>podman</code>        | <code>[fd77:...:e8ca]:9003 -&gt; 80</code>                                                                                 | Stateless whiteboard                                                                               |
| <code>funkwhale</code>  | <code>funkwhale.service</code>                                                                                        | <code>funkwhale_net</code> | <code>[fd77:...:f7a0]:8085 -&gt; 80</code>                                                                                 | Embedded PG/Redis + LL<br><code>auth.drcheck.dev:636</code> via ( <code>169.254.1.2:636</code> )   |
| <code>grill</code>      | <code>grill.service</code>                                                                                            | <code>grill_net</code>     | <code>[fd77:...:fb02]:8000</code> , <code>[10.0.0.3:8000,8088 -&gt; 8000</code>                                            | SQLite + ESP32-C3 HTTP                                                                             |
| <code>immich</code>     | <code>immich_server.service</code><br><code>immich_machine_learning.service</code>                                    | <code>immich_net</code>    | <code>[fd77:...:2283]:2283 -&gt; 2283</code><br><i>(internal ML worker)</i>                                                | Host PG ( <code>pgvector.rs</code> ) + PG<br>( <code>169.254.1.2:5432</code> ) + ROCm GPU          |
| <code>navidrome</code>  | <code>navidrome.service</code>                                                                                        | <code>podman</code>        | <code>[fd77:...:5a70]:4533 -&gt; 4533</code>                                                                               | <code>/data/music/flac:ro</code> , SQL                                                             |
| <code>nextcloud</code>  | <code>nextcloud.service</code>                                                                                        | <code>nextcloud_net</code> | <code>[fd77:...:7a31]:8081 -&gt; 80</code>                                                                                 | Host PG ( <code>nextcloud_db</code> ) + PG<br>( <code>169.254.1.2:5432</code> ) + OIDC + Collabora |

| Service                  | Systemd Unit(s) ( <code>docker-data@</code> )                                                                                                                                                    | Network                    | Published Host Binding(s)                                                                | Backing Store / Notes                                                              |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <code>penpot</code>      | <code>penpot-frontend.service</code><br><code>penpot-backend.service</code><br><code>penpot-exporter.service</code><br><code>penpot-postgres.service</code><br><code>penpot-redis.service</code> | <code>penpot_net</code>    | <code>[fd77:::be49]:9001 -&gt; 8080</code><br>(4 internal services)                      | ZFS<br><code>/data/docker/penpot/{pos</code><br>+ OIDC                             |
| <code>romm</code>        | <code>romm.service</code><br><code>romm-db.service</code>                                                                                                                                        | <code>romm_net</code>      | <code>[fd77:::7033]:8080 -&gt; 8080</code><br>(internal MariaDB 3306)                    | MariaDB 11 sidecar + <code>/d</code><br>OIDC                                       |
| <code>scontrol</code>    | <code>scontrol.service</code>                                                                                                                                                                    | <code>scontrol_net</code>  | <code>[fd77:::5c01]:8000 -&gt; 8000</code>                                               | <code>id_scontrol</code> SSH orchest<br><code>kvm-ssh</code>                       |
| <code>sopro</code>       | <code>sopro.service</code>                                                                                                                                                                       | <code>podman</code>        | <code>[fd77:::509]:7860 -&gt; 7860</code>                                                | Local TTS image ( <code>localhc</code>                                             |
| <code>step-ca</code>     | <code>step-ca.service</code>                                                                                                                                                                     | <code>webshell_net</code>  | <code>[fd77:::ca]:9000 -&gt; 9000</code>                                                 | Smallstep SSH & X.509 C<br><code>webshell</code> & <code>ipctl</code> )            |
| <code>subocr</code>      | <code>subocr.service</code>                                                                                                                                                                      | <code>subocr_net</code>    | <code>[fd77:::c70]:8000 -&gt; 8000</code> , <code>127.0.0.1:8077 -&gt; 8000</code>       | ROCm MIGraphX GPU + <code>t</code><br><code>169.254.1.2:11434</code> )             |
| <code>tldraw</code>      | <code>tldraw.service</code>                                                                                                                                                                      | <code>podman</code>        | <code>[fd77:::71d7]:9002 -&gt; 80</code>                                                 | Stateless canvas                                                                   |
| <code>vaultwarden</code> | <code>vaultwarden.service</code>                                                                                                                                                                 | <code>podman</code>        | <code>[fd77:::a117]:8222 -&gt; 80</code>                                                 | SQLite + OIDC ( <code>AddHost=auth.drheck.dev:</code>                              |
| <code>vikunja</code>     | <code>vikunja.service</code>                                                                                                                                                                     | <code>vikunja_net</code>   | <code>[fd77:::71c0]:3456 -&gt; 3456</code>                                               | Host PG + Host Valkey ( <code>t</code><br>JSON <code>HealthCmd</code>              |
| <code>webrcade</code>    | <code>webrcade.service</code>                                                                                                                                                                    | <code>podman</code>        | <code>[fd77:::a1ca]:8080 -&gt; 80</code>                                                 | <code>/data/docker/webrcade/co</code><br><code>/data/games:ro</code>               |
| <code>webshell</code>    | <code>webshell.service</code>                                                                                                                                                                    | <code>webshell_net</code>  | <code>[fd77:::771d]:7681 -&gt; 7681</code>                                               | Dual-stack <code>tttyd</code> + epher<br>SSH certs                                 |
| <code>wick</code>        | <code>wick.service</code>                                                                                                                                                                        | <code>podman</code>        | <code>[fd77:::41c8]:9005 -&gt; 80</code>                                                 | Static HTML5 studio on <code>n</code>                                              |
| <code>wipstudio</code>   | <code>wipstudio.service</code><br><code>wipstudio-minio.service</code>                                                                                                                           | <code>wipstudio_net</code> | <code>[fd77:::7190]:3000 -&gt; 3000</code><br><code>[fd77:::7190]:9000 -&gt; 9000</code> | Host PG + MinIO S3 sidec<br><code>DISABLE_SIGN_UP=true</code> )                    |
| <code>yourls</code>      | <code>yourls.service</code><br><code>yourls-db.service</code>                                                                                                                                    | <code>yourls_net</code>    | <code>[fd77:::77]:8080 -&gt; 8080</code><br>(internal MariaDB 3306)                      | MariaDB 10.11 sidecar ( <code>YOURLS_PRIVATE=true</code> + <code>Fc</code><br>API) |

## Reference Quadlet Examples

**Pattern A: OIDC + Host PostgreSQL/Valkey Service (e.g.,**  
`/data/docker/vikunja/vikunja.container` )

```
[Unit]
Description=Vikunja Task Manager (Rootless Podman)
After=network-online.target
Wants=network-online.target

[Container]
Pull=never
ContainerName=vikunja
Image=docker.io/vikunja/vikunja:latest
Network=vikunja.network
AddHost=auth.drheck.dev:169.254.1.2
PublishPort=[fd77:7777:5010:0:1337:flac:d00d:71c0]:3456:3456
EnvironmentFile=/data/docker/vikunja/.env
Volume=/data/docker/vikunja/config.yml:/etc/vikunja/config.yml:ro
Volume=/data/docker/vikunja/files:/app/vikunja/files:rw
HealthCmd=["/app/vikunja/vikunja", "healthcheck"]
HealthInterval=30s
HealthRetries=3

[Service]
Restart=always
RestartSec=5
TimeoutStartSec=120

[Install]
WantedBy=default.target
```

## Pattern B: Multi-Container ROCm GPU Stack (e.g.,

`/data/docker/immich/immich_machine_learning.container`)

```
[Unit]
Description=Immich ROCm Machine Learning Worker (Rootless Podman)
After=network-online.target
Wants=network-online.target

[Container]
Pull=never
ContainerName=immich_machine_learning
```

```
Image=ghcr.io/immich-app/immich-machine-learning:v3.3.1-rocm
```

```
Network=immich.network
```

```
AddDevice=/dev/dri:/dev/dri
```

```
AddDevice=/dev/kfd:/dev/kfd
```

```
GroupAdd=keep-groups
```

```
EnvironmentFile=/data/docker/immich/.env
```

```
Volume=/data/docker/immich/model-cache:/cache:rw
```

```
[Service]
```

```
Restart=always
```

```
RestartSec=5
```

```
TimeoutStartSec=180
```

```
[Install]
```

```
WantedBy=default.target
```

## 4. Day-to-Day Operations: Status, Logs & Lifecycle

**Important Execution Context on Arch Linux:** Rootless Podman containers run under `docker-data` (UID 1100), not `root` or `luke`. Two Arch Linux `systemd` / `podman` details matter in CLI commands:

1. Always run `sudo -u docker-data` from a world-traversable directory such as `cd /tmp` (otherwise `podman` tries to `stat("/home/luke")` on mode `0700` and warns about `Permission denied`).
2. On `systemd 258+`, `systemctl --user -M docker-data@ ...` works directly, whereas `journalctl --user -M docker-data@` returns `Connecting to a machine as non-root is not supported`. Always read user unit journals via `sudo -u docker-data XDG_RUNTIME_DIR=/run/user/1100 journalctl --user -u <service>.service`.

### 4.1 Checking Service, Container & Namespace Status

```
1. Host-wide health summary across all nginx, native systemd, and rootless Podman services
$ status_all
$ services

2. Live container-to-host UID, PID, and UserNS translation audit tree
$ sudo /usr/local/sbin/podman-map-tree

3. Check systemd unit status for any Quadlet container
$ sudo systemctl --user -M docker-data@ status authentik-server.service

4. List all running rootless Podman containers, networks, ports, and health states
$ (cd /tmp && sudo -u docker-data HOME=/home/docker-data XDG_RUNTIME_DIR=/run/user/1100 podman
ps -a)

5. Live CPU, memory, and I/O telemetry across all 41 containers
$ (cd /tmp && sudo -u docker-data HOME=/home/docker-data XDG_RUNTIME_DIR=/run/user/1100 podman
stats --no-stream)
```

## 4.2 Viewing Service & Container Logs

```
Follow live systemd user unit logs (includes systemd lifecycle + container stdout/stderr)
$ sudo -u docker-data XDG_RUNTIME_DIR=/run/user/1100 journalctl --user -u authentik-
server.service -f

View the last 100 journal lines without paging
$ sudo -u docker-data XDG_RUNTIME_DIR=/run/user/1100 journalctl --user -u
immich_server.service -n 100 --no-pager

Follow container stdout/stderr directly via podman logs
$ (cd /tmp && sudo -u docker-data HOME=/home/docker-data XDG_RUNTIME_DIR=/run/user/1100 podman
logs --tail 100 -f bookstack)
```

## 4.3 Starting, Stopping, Restarting & Applying Quadlet Edits

```
Restart a running Quadlet container
$ sudo systemctl --user -M docker-data@ restart vaultwarden.service

Reload systemd generator after editing /data/docker/<service>/<service>.container
$ sudo systemctl --user -M docker-data@ daemon-reload
$ sudo systemctl --user -M docker-data@ restart <service>.service
```

## 4.4 Updating Images & Rebuilding Local Dockerfiles

Because every `.container` unit sets `Pull=never` for offline boot safety, image updates are always explicit:

```
Pull updated registry image into docker-data's store and restart
$ (cd /tmp && sudo -u docker-data HOME=/home/docker-data XDG_RUNTIME_DIR=/run/user/1100 podman
pull docker.io/vaultwarden/server:latest)
$ sudo systemctl --user -M docker-data@ restart vaultwarden.service

Rebuild a locally built Dockerfile image (use TMPDIR=/dev/shm for large ROCm images)
$ (cd /data/docker/subocr && sudo -u docker-data HOME=/home/docker-data
XDG_RUNTIME_DIR=/run/user/1100 TMPDIR=/dev/shm podman build -t localhost/subocr:latest
/data/docker/subocr)
$ sudo systemctl --user -M docker-data@ restart subocr.service
```

---

## 5. Troubleshooting & Diagnostics Runbook

### 5.1 Symptom: OIDC Login or Container LDAPS Fails with `Connection refused` to `auth.drheck.dev`

- **Root Cause:** Inside a rootless `pasta` network namespace, `auth.drheck.dev` resolves via DNS to host `10.0.0.100` or `fd77:::aaaa`, which loops back inside the container's own network namespace where nothing listens on `:443` or `:636`.
- **Fix:** Add `AddHost=auth.drheck.dev:169.254.1.2` under `[Container]` in `<service>.container`, reload `daemon-reload`, and restart the service so OIDC and LDAPS connections route cleanly through `--map-host-loopback` to `127.0.0.1:443` / `127.0.0.1:636`.

## 5.2 Symptom: Multi-Container Service Fails to Connect to Sibling Hostname (`penpot-redis` / `bookstack_db`) After Abrupt Kill

- **Root Cause:** If a container on a custom `<service>.network` bridge is killed abruptly without running `netavark teardown`, a stale container ID + dead IP line can remain ahead of the new container's IP in `/run/user/1100/containers/networks/aardvark-dns/<net_name>`. Clients that connect only to the first `A` record returned by DNS (such as Java's Lettuce client in `penpot-backend`) time out against the stale IP.
- **Diagnosis & Fix:**
  1. Inspect `/run/user/1100/containers/networks/aardvark-dns/<net_name>` against live container IDs from `podman ps --no-trunc`.
  2. Remove any line whose 64-character container ID is no longer running, then reload `aardvark-dns` without restarting healthy containers:

```
$ sudo kill -HUP "$(cat /run/user/1100/containers/networks/aardvark-dns/aardvark.pid)"
```

## 5.3 Symptom: Permission denied Inside Non-Root Worker Container (UID 624320 / 625288) on `/data/docker/<svc>/...`

- **Root Cause:** Worker processes running as container UID `$u > 0$` execute on the host as subordinate UID `$624287 + u$` (outside group `1100`). If the parent directory `/data/docker/<svc>` has mode `2770` (`drwxrws---`, missing `other` execute `+x`), subordinate UIDs cannot traverse `/data/docker/<svc>` to reach child bind mounts, and tools run on the host (like `git pull`) can accidentally write `root`-owned files over `625287`-owned files.
- **Fix:**

1. Keep parent directories traversable without allowing directory listing (`sudo chmod 2771 /data/docker/<svc>`, i.e., `drwxrws--x`).
2. Execute inside-container file updates inside the container namespace (or via `podman unshare`), just as `/usr/local/bin/degoog-update` runs `git fetch` and `bun install` inside `degoog`.

## 5.4 Symptom: Distroless Container Marks `unhealthy` Despite Listening Normally (e.g., `vikunja`)

- **Root Cause:** Distroless images contain no `/bin/sh`, so string-form `HealthCmd=/app/vikunja/vikunja healthcheck` fails because Podman invokes `/bin/sh -c` unless `HealthCmd` is formatted as a JSON exec array.
- **Fix:** Write `HealthCmd=["/app/vikunja/vikunja", "healthcheck"]` as a JSON array in `<service>.container`.

## 5.5 Symptom: `502 Bad Gateway` on Dual-Stack Service When Passing IPv6-Only Flag (e.g., `ttyd -6 -i ::`)

- **Root Cause:** Rootless port forwarding may connect into the container over IPv4 even when accepting connections on an `eth77` ULA IPv6 socket. Passing IPv6-only bind flags inside the container rejects IPv4 loopback/bridge forwards.
- **Fix:** Let internal container daemons listen on default dual-stack (`0.0.0.0 / ::`) rather than forcing IPv6-only flags.

## 5.6 Validating Quadlet Syntax Before Restarting

```
$ (cd /tmp && sudo -u docker-data HOME=/home/docker-data XDG_RUNTIME_DIR=/run/user/1100 /usr/lib/systemd/user-generators/podman-system-generator --user --dry-run)
```

Revision #8

Created 2026-10-19 13:35:23 UTC by luKe

Updated 2026-10-20 11:53:18 UTC by luKe