RustDesk - pfSense and and HAProxy

You can configure self-hosted RustDesk (hbbs ID/rendezvous server + hbbr relay server) behind your pfSense + HAProxy Docker setup using `hostname.domain.com` as the external ID server address. However, there are important caveats due to how RustDesk works.

Important Limitations with HAProxy for RustDesk

RustDesk uses a **custom protocol** (not HTTP/HTTPS) with:

- **TCP ports**: 21115 (hbbs signaling), 21116 (hbbs), 21117 (hbbr relay)
- **UDP port**: 21116 (critical for NAT traversal / hole punching)
- Optional WebSocket ports 21118/21119 (only needed for the web client)

HAProxy (even in TCP mode) is not ideal for the core RustDesk traffic:

- It cannot properly proxy UDP 21116 (HAProxy's UDP support is very limited and not suitable here).
- Even in TCP mode, the real client IP is hidden (hbbs sees the HAProxy container's IP), which breaks NAT hole-punching and forces everything through the relay.
- Domain-based routing (`hostname.domain.com`) doesn't help because RustDesk clients connect directly to the resolved IP + specific ports — there is no Host header or SNI to route on.

Recommended approach: Use **pfSense Port Forward / NAT rules** for the RustDesk ports (direct to your RustDesk Docker host/container). Keep HAProxy for your HTTP/HTTPS services only. This is the most reliable way and what most people successfully use with pfSense + RustDesk.

If you *must* route everything through HAProxy Docker, it's possible for the TCP ports only (see advanced section at the end), but UDP will still need a direct forward and performance/reliability will suffer.

Step-by-Step Recommended Setup (Direct pfSense NAT + Domain)

1. **Set up DNS**
   - Create an A record (and AAAA if IPv6) for `hostname.domain.com` pointing to your **pfSense WAN public IP**.
   - No special subdomains or paths needed.
2. **Run RustDesk Server in Docker** (on your internal host/VM/LXC behind pfSense) You can also choose to use the official Docker Compose (download `oss.yml` from RustDesk or use this standard one):

```
services:
  hbbs: # ID / signaling server
    container_name: hbbs
    image: rustdesk/rustdesk-server:latest
    command: hbbs -r {REMOTE HOSTNAME}:21117
    logging:
      driver: json-file
      options:
        max-file: ${DOCKERLOGGING_MAXFILE:-10}
        max-size: ${DOCKERLOGGING_MAXSIZE:-200k}
    ports:
      - "21115:21115"
      - "21116:21116"
      - "21116:21116/udp"
      - "21118:21118"
    volumes:
      - ${DOCKERFOLDER}/rustdesk/data:/root
    depends_on:
      - hbbr
    restart: unless-stopped
  hbbr: # Relay server
    container_name: hbbr
    image: rustdesk/rustdesk-server:latest
    command: hbbr
    logging:
      driver: json-file
      options:
        max-file: ${DOCKERLOGGING_MAXFILE:-10}
```

```
    max-size: ${DOCKERLOGGING_MAXSIZE:-200k}
  ports:
    - "21117:21117"
    - "21119:21119"
  volumes:
    - ${DOCKERFOLDER}/rustdesk/data:/root
  restart: unless-stopped
```

- Run with `docker compose up -d`.
- The `./data` folder will contain your `encryption` key pair (`id_ed25519.pub`).
- Check logs: `docker logs hbbs` → look for the **public key** (you'll need this in clients).

3. **pfSense Configuration (Port Forwards + Firewall Rules)** Go to **Firewall → NAT → Port Forward** and create these rules (all source = WAN, destination = your RustDesk Docker host's internal IP):

Protocol	Destination Port	Forward to IP	Forward to Port	Notes
TCP	21115	RustDesk Docker IP	21115	hbbs signaling
TCP	21116	RustDesk Docker IP	21116	hbbs
UDP	21116	RustDesk Docker IP	21116	Critical for hole punching
TCP	21117	RustDesk Docker IP	21117	hbbr relay

   - Set **Source port range** to *any* (RustDesk clients use random source ports).
   - Auto-create the associated firewall rules (or manually create matching WAN rules under **Firewall → Rules → WAN** allowing the same ports).

   Optional but recommended:
   - Enable **System → Advanced → Firewall & NAT → NAT Reflection** (Pure NAT mode) if you want LAN clients to reach the server via the public domain.
4. **Client Configuration** On every RustDesk client (Windows/macOS/Linux/Android/iOS):
   - Go to **Settings → ID/Relay Server**
   - **ID Server**: hostname.domain.com (or `hostname.domain.com:21115` if you changed the port)
   - **Relay Server**: leave blank (hbbs will tell clients to use `hostname.domain.com:21117`)
   - **Key**: paste the public key from your `./data/id_ed25519.pub` file
5. **Test**
   - From outside your network, connect using an ID from one of your machines.
   - Check connection type in the session (should be "Direct" most of the time; falls back to "Relay" if needed).
   - Monitor logs: `docker logs hbbs` and `docker logs hbbr`.

Advanced: Forcing Everything Through HAProxy Docker (Not Recommended)

If you really want HAProxy as the single entry point:

- Forward **all** the ports above from pfSense WAN → your **HAProxy Docker container IP** (instead of directly to RustDesk).
- In your HAProxy config (or pfSense HAProxy package if you're using that), add TCP-mode frontends/backends for each TCP port:

```
frontend rustdesk_hbbs_21115
    bind :21115
    mode tcp
    default_backend rustdesk_hbbs

backend rustdesk_hbbs
    mode tcp
    server hbbs rustdesk-docker-ip:21115

# Repeat for 21116 (TCP only), 21117, etc.
```

- **UDP 21116** → still do a direct pfSense NAT forward to the RustDesk container (bypassing HAProxy).
- You will likely have NAT traversal issues because of the hidden client IP.

For the RustDesk **Web Client** only (if you use it), you can proxy the WebSocket ports (21118/21119) through HAProxy in HTTP mode with proper `Upgrade` headers — similar to the official Nginx example in the RustDesk docs.

This direct pfSense NAT + domain setup is what works reliably for almost everyone running RustDesk behind pfSense. Let me know if you hit any specific error logs or need the exact Docker Compose / HAProxy snippets adjusted for your environment!

Revision #0
Created 28 April 2026 01:19:18 by Admin
Updated 24 October 2026 14:10:46 by Admin