

Your own Bitcoin solo-mining pool

Public 0% Solo Pool — Operator Manual

Running dvb-WarpPool as an open, non-custodial, zero-fee pool

Version 1.52.0 · Edition November 2026

Contents

1 What a public 0% pool is

2 Turning on per-worker mode

3 What miners enter

4 Public wallet pages & the owner panel

5 Operator privacy

6 Accepting donations (optional)

7 Public hardening profile

8 TLS

9 Behind a reverse proxy

10 Deployment checklist

11 Support & links

1 · What a public 0% pool is

A public 0% pool turns dvb-WarpPool into an **open, non-custodial, zero-fee solo pool**: any miner who connects mines to *their own* Bitcoin address — taken from the Stratum login — and the full block reward goes straight to whoever finds the block, on-chain. The pool never holds funds. There is nothing to withdraw, and nothing to trust.

Technically this is **per-worker payout mode**. In the default `single` mode the whole reward pays one operator address — the ordinary "solo + friends" setup, where every miner works for *your* wallet. In **per-worker** mode each miner supplies its own address as the Stratum username, and the coinbase is built per-connection to pay exactly that address: **one output, the full subsidy, no fee.**

The four properties that define it

- **Non-custodial.** The pool never touches anyone's coins. The Bitcoin network pays the finder directly.
- **Per-worker.** Different miners can use different addresses; each keeps the full reward for the blocks it finds.
- **Full reward to the finder.** A single coinbase output pays the whole subsidy plus fees to the address in the winning miner's login — 0% skim.
- **Fail-closed.** The daemon refuses to start if a fee or operator donation is configured together with per-worker mode, so you cannot accidentally run a "public" pool that quietly skims the reward.

> **Honesty guardrail — please read.** dvb-WarpPool has **not yet found a mainnet block.** The consensus path (per-worker coinbase → a real Bitcoin Core node accepts the block → the reward pays the finder) is verified end-to-end on regtest and signet, but the first mainnet block is still **"not yet"**. Advertise the pool for exactly what it is — a fresh, open, non-custodial solo pool — and **never** as "battle-tested". If you publish a landing page, keep that guardrail visible.

> **Contrast with `single` mode.** If you only want to mine your own rigs, or a fixed group of friends, all paying one wallet, stay in `single` mode — see the main *User Manual*. This operator manual is specifically about opening the pool up so strangers can point their miners at it and each keep what they find.

2 · Turning on per-worker mode

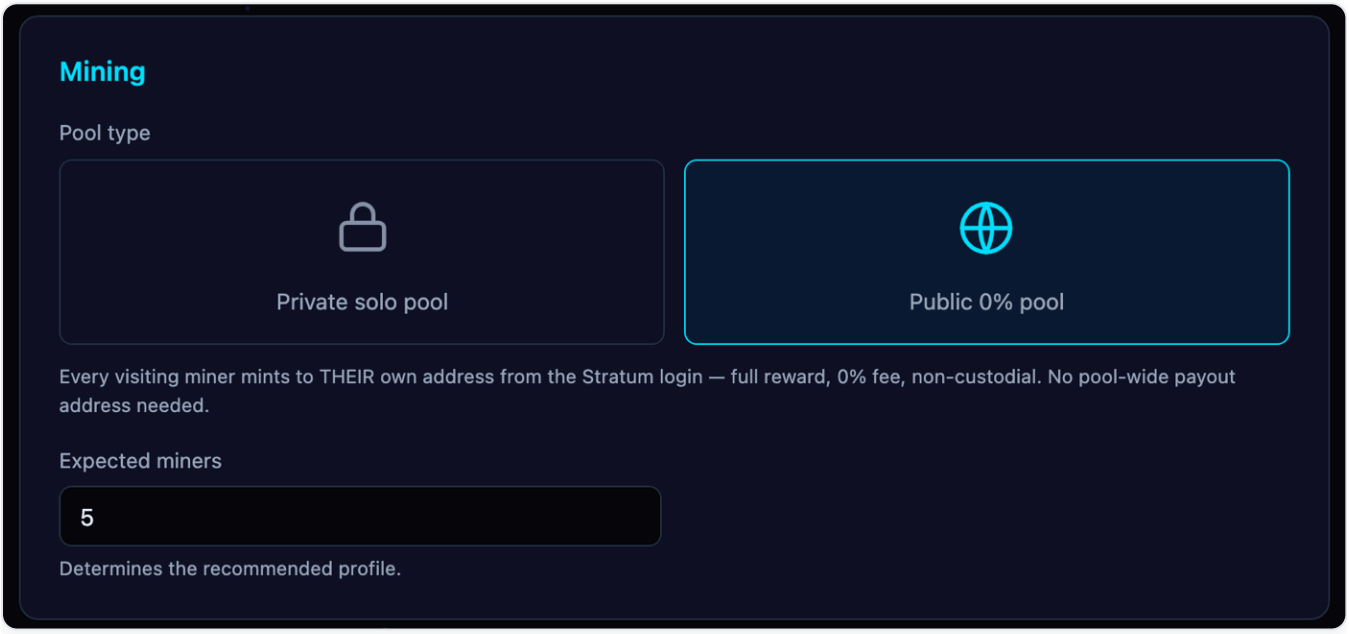
There are two routes to the same result: the setup wizard (recommended for a new install) or a one-line edit in `config.toml`. Both write the same `payout_mode = "per_worker"` flag.

Route A — the setup wizard

When you install through the setup wizard, its **Mining** card offers a **Pool type** choice: **Private solo pool** versus **Public 0% pool**. Picking **Public 0% pool** does three things for you:

- writes `payout_mode = "per_worker"`;
- hides the pool-wide **payout-address** field (it is unused in this mode — each miner brings its own);
- forces the **pool fee** to 0 and hides the fee fields.

No hand-editing is required.



The setup wizard's Mining card: choosing “Public 0% pool” switches the pool to per-worker payouts and hides the payout-address and fee fields.

Route B — `config.toml`

If you manage the config by hand, one flag is the whole feature switch:

```
[mining]
payout_mode = "per_worker"
# payout_address is IGNORED in per_worker mode (each miner brings its own).
# pool_fee_percent / operator_donation_percent MUST be 0 — the daemon
# refuses to start otherwise (full reward goes to the finder).
```

Fail-closed on fee or donation

The daemon **fails closed** on startup if a non-zero `pool_fee_percent` or `operator_donation_percent` is configured together with `per_worker`. It will not boot until the fee and donation are back at zero. This is a deliberate safety invariant: a per-worker pool pays the finder the *full* reward, so there is no configuration in which it can silently take a cut.

Typos never pay the wrong person. An invalid or wrong-network login address is **rejected** at `mining.authorize` (Stratum V1) or channel-open (Stratum V2). There is **no fallback** to an operator address — a mistyped address is refused, never quietly redirected.

3 · What miners enter

A miner joining your public pool configures its Stratum login exactly as on any solo pool, with one rule: **the username is the miner's own Bitcoin address**, optionally followed by a rig label after a dot.

```
Username:  bc1qyour-own-address.rig1
Password:  x   (anything — it is not used)
```

- Everything **before the first dot** is the payout address; anything after it is a free rig label used only for statistics.
- The password can be anything (commonly `x`).

Supported address types

Any valid mainnet address works. The pool accepts:

Type	Example prefix
P2PKH (legacy)	`1…`
P2SH (incl. wrapped SegWit)	`3…`
P2WPKH (native SegWit)	`bc1q…`
P2WSH (native SegWit script)	`bc1q…`
P2TR (Taproot)	`bc1p…`

> **Wrong address = refused, not redirected.** If the login is not a valid address for this network, the miner is rejected at authorize time. There is no fallback address, so a typo simply fails to connect instead of paying the block reward to the wrong recipient.

4 · Public wallet pages & the owner panel

Every address that mines on your pool gets its own login-free status page — the "look up my address" flow a public solo pool is expected to have. Miners bookmark their own; nobody needs an account.

The public page at `/users/<address>`

Open `/users/<address>` and you see that address's:

- **total hashrate** in large type, summed across every rig logged in under the address, with a 1 m / 5 m / 1 h / 1 d / 7 d window switch;
- a **24-hour hashrate chart** with the best shares drawn as dots;
- the **best-ever share difficulty**, **rigs online** and **lifetime accepted shares**;
- a **per-rig table** — one row per `address.rig` login, with an online dot, hashrate, best share, last share and firmware, plus a WarpTune column if the owner shares that badge;
- an honest **block-odds panel** (see below).

No login, no operator involvement — anyone who has the address can open it.

The **block-odds panel** replaces the old single "estimated time to block" — a mean that reads as "never" for a home rig and hides the point of solo mining: the process is **memoryless**, every second a fresh draw with nothing getting closer. Instead it shows the real distribution: the **chance in the next 24 hours** and **within a year** (long odds as "1 in N", percentages floored so they never overstate), the **50/50 point** (median, not average), and **luck so far** — expected vs. found blocks with an honest "N % of miners with your history would also have none yet". A memorylessness footnote makes the core point explicit: no amount of waiting makes a block more likely.



A public per-wallet page at /users/<address>: aggregate hashrate, per-rig breakdown and best share — no login required.

ckpool-compatible JSON. The same numbers are served at `GET /api/users/<address>` in the ckpool field format (`hashrate1m`, `hashrate1hr`, `bestshare`, `worker[]`, …), so miner firmware and stats tools such as `ckstats` can read it directly. Both routes are plain HTTP under the status port, behind the same reverse proxy as the dashboard.

The opt-in owner panel

By default a wallet page is public. A miner who wants more control can **prove ownership** of the address and unlock an owner panel — entirely per-miner, nothing the operator configures:

1. The page hands out a short one-time **challenge message**.
2. The miner signs it in their own wallet's *Sign Message* box — the standard BIP-137 "verify address" feature in Electrum, Sparrow, Bitcoin Core and hardware wallets. **No private key leaves the wallet, no funds move, no account is created** — the pool only checks the signature.
3. A valid signature sets a short-lived owner cookie for that address and reveals three controls:
   - **Privacy** — hide the address from the public wallet list and blank its `/users/<address>` data for everyone but the verified owner (the direct link still works, for them).
   - **Notifications** — point a personal ntfy topic or webhook at this address's milestones: *block found*, a new *personal-best share* (across all its rigs), first entry into the pool's *all-time top 100*, and *rig-offline*. Each has its own cooldown against spam, and a found block sends one headline rather than three alerts.
   - **WarpTune badge** (since v1.31.0) — create a **push token** for this address, put it into the environment variable whose name `warptune.toml` gives as `pool_push_token_env`, on the machine where your dvb-WarpTune sidecar runs, and set `pool_push = true`. The sidecar then sends its snapshot to this pool and the rigs table shows a badge per rig: mode, frequency, dry run / live / control off / stale; hover it for the last decision. Visitors see a reduced badge (no last decision) only after **Show badge to visitors**. The token is shown once, can be rotated or revoked here, and the pool never sends anything back to a miner.

Coinbase text (v1.45, per-worker pools): once you have proved your address you can set up to 16 characters (letters, digits, punctuation, spaces; no `/`) that go into the coinbase of every block this pool builds for your address, right after the pool tag — your mark on the chain, visible in any block explorer. Takes effect with the next job; the operator can switch the feature off.

Taproot works too now. Ownership signing covers single-key **P2PKH / P2WPKH / P2SH-P2WPKH** addresses (BIP-137) and **Taproot (P2TR)** via **BIP-322** — sign with a wallet that supports it, such as Sparrow. The owner panel needs **no operator config** — it is driven entirely by the miner.

5 · Operator privacy

A public mainnet pool should not hand out an enumerable list of everyone's address and rig setup. dvb-WarpPool gives you **two independent layers** of control. Both default to the open behaviour, so hardening here is opt-in.

Layer A — config: directory & leaderboard exposure

Two keys in `[privacy]` set the default exposure of the whole UI. In both cases the **logged-in operator still sees the full data** — these only affect anonymous visitors, so keep admin auth enabled.

```
[privacy]
# Hide the browsable Workers/wallet directory from anonymous visitors. Each
# /users/<address> page still works by direct link; the logged-in operator
# still sees the full directory (so keep admin auth enabled).
wallet_directory_public = false
# Mask the Bitcoin address in the public best-shares leaderboard, server-side.
# The logged-in operator still sees full addresses.
leaderboard_addresses_public = false
```

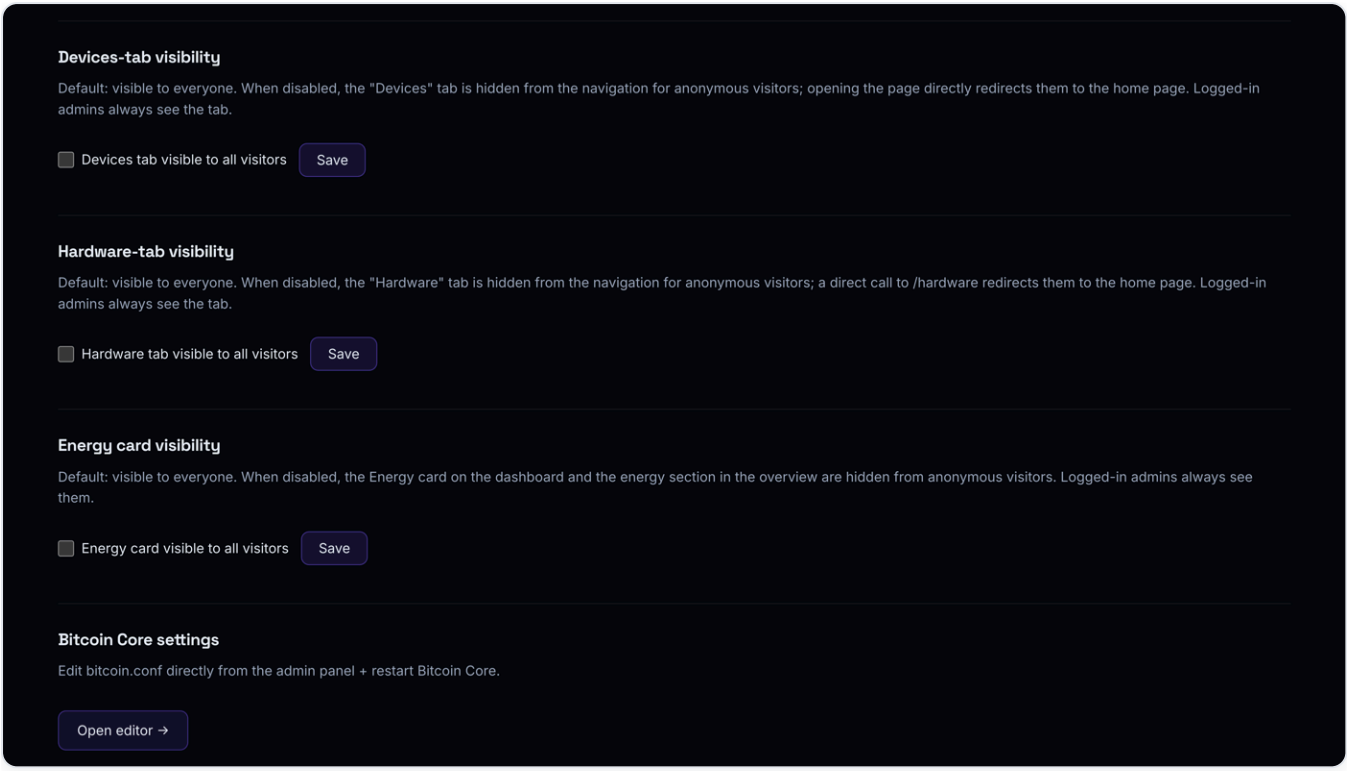
With this posture, miners reach their own page by link, but the pool doesn't publish a browsable list of everyone, and leaderboard addresses are masked server-side. It touches neither payouts nor block construction.

Layer B — Admin-UI visibility toggles

Under **Admin** → **Profile** a set of visibility toggles hide whole views from anonymous visitors:

- the **Devices** tab (the `/miners` route — the interactive device manager with per-device telemetry);
- the **Hardware** tab (host resources and size-profile picture);
- the **Energy** card on the dashboard.

> **Which tab does the toggle hide?** The visibility toggle controls the **Devices** tab — the `/miners` route with live device telemetry — **not** the Miners/workers listing. The names are easy to mix up (Workers = wallets, Miners = Stratum logins, Devices = the hardware); the toggle is about the device-manager view.



Admin → Profile: visibility toggles hide the Devices/Hardware/Energy views from anonymous visitors; addresses in the public leaderboard are masked server-side.

> **Recommended posture for public mainnet.** Set `wallet_directory_public = false` and `leaderboard_addresses_public = false`, and hide the Devices, Hardware and Energy views from anonymous visitors. Miners still reach their own `/users/<address>` page by link; the operator keeps the full picture when logged in.

Removing a worker

If a miner asks to be taken off the leaderboard, or old test names clutter it, open **Admin → Workers**. **Remove** forgets a worker that is not connected: its row, its best shares (so it leaves the leaderboard) and its VarDiff memory. Share history and found blocks stay. A connected worker cannot be removed: the miner has to disconnect first, and if it connects again later it simply starts afresh. A miner that has just stopped shows as **recently active** and can already be removed. Names whose worker entry the retention has already dropped but which are still on the leaderboard are listed in a second table, **Leftovers without a worker entry**, and removed the same way. Every attempt is recorded in the audit log.

6 · Accepting donations (optional)

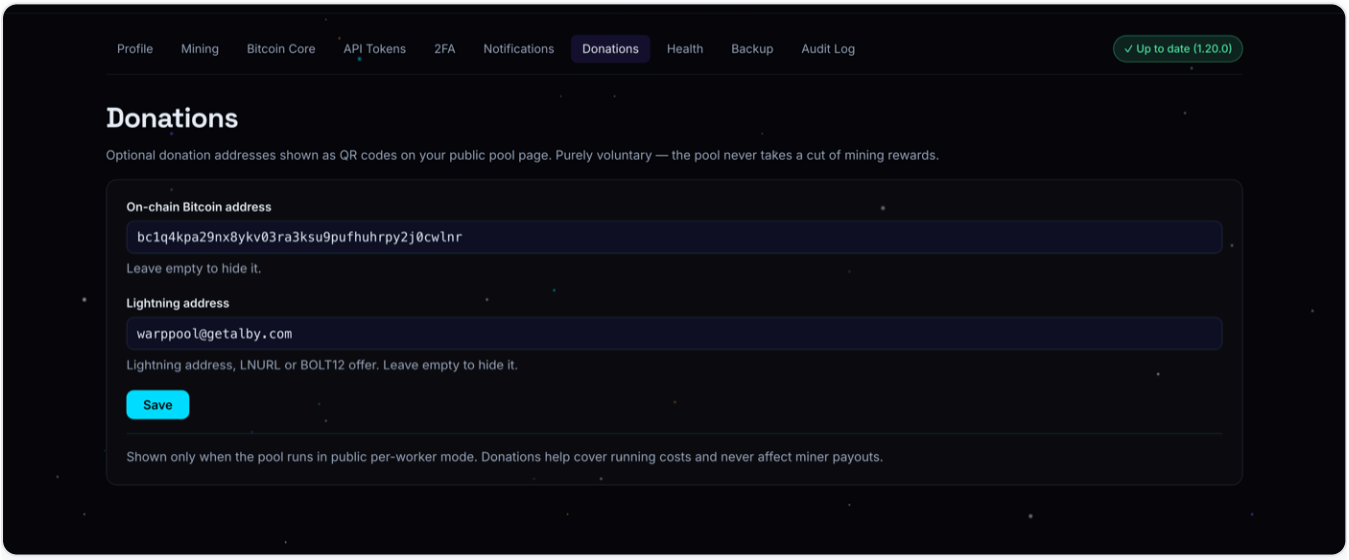
Because a per-worker pool takes **0%** of every reward, the only way to fund hosting is a voluntary donation. The dashboard has built-in support for this, and the whole feature appears **only in per-worker mode** — a private solo pool never shows it.

Setting the addresses

Set them under **Admin** → **Donations**. The nav entry is **hidden unless the pool runs in per-worker mode**.

Field	What to enter
On-chain	Any valid Bitcoin address. It is validated on save and shown as a `bitcoin:` QR code the visitor's wallet can open directly.
Lightning	A Lightning address, LNURL or BOLT12 string, stored as you enter it. Shown as its own copy field and QR code.

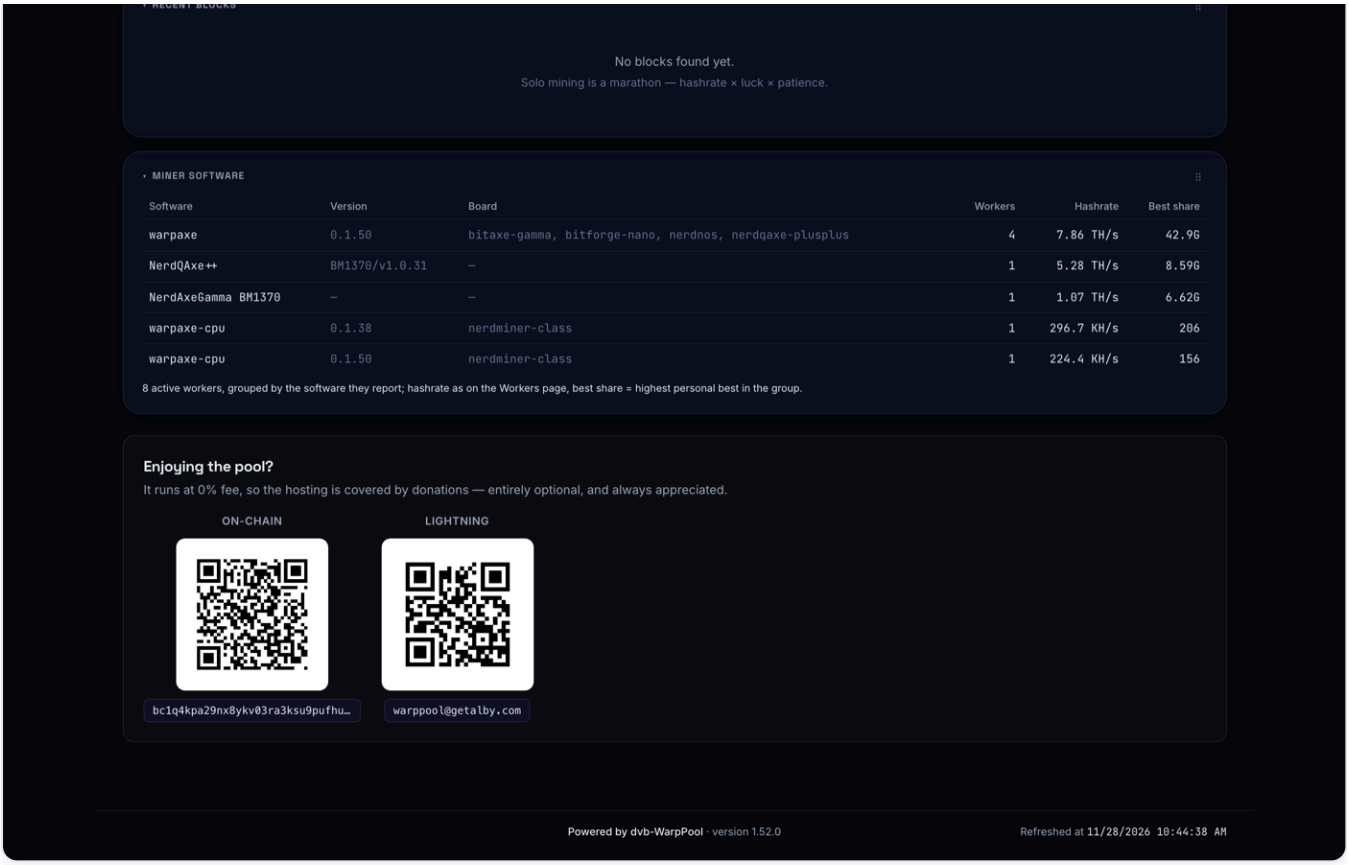
Leave either field blank to hide that option. When at least one is set, a **"Enjoying the pool?"** card with QR codes and click-to-copy addresses appears at the **bottom of the public dashboard**.



Admin → Donations (per-worker pools only): set an on-chain and/or Lightning tip address; a donation card then appears on the public dashboard.

> **Donations never touch the block reward.** These addresses are stored as pool settings (not `config.toml`), so saving takes effect immediately with **no restart**. They are plain operator metadata — they are **never mixed into coinbase construction**, so setting one cannot affect the fail-closed rule that a per-worker pool pays the finder the full reward. A "donation" here is a visitor sending sats to a displayed address, not a share of anyone's block reward.


dvb-WarpPool
Dashboard
Hashrate
Overview
Blocks
Workers
Miners
RPC READY
IPC
LARGE (≤100 MINERS)
LIVE
EN
Login
Dashboard
Live status of your Bitcoin solo pool.
How to connect your miner
POOL
POOL HASHRATE LIVE
15.63 TH/s
24h avg 14.88 TH/s · 32.0 /min (2.13 /worker)
BLOCKS
0
WORKERS
16
39 miner
BEST SHARE
42.92G
bc1qtg…eqz0.bitaxe602
upd. 7/23/2026
UPTIME
1d 4h
ACTIVE MINERS
4 / 15
4 worker · 15 miner
3 TLS · 6 SV2
POOL FEE
0.00 %
POOL HASHRATE (24H)
Full view →
BEST SHARE
588M
441M
294M
147M
0
35.0 TH/s
30.0 TH/s
25.0 TH/s
20.0 TH/s
15.0 TH/s
10.0 TH/s
5.00 TH/s
0 TH/s
10:43
12:00
14:00
16:00
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
NETWORK
NETWORK HASHRATE
948.77 EH/s
DIFFICULTY
145.09 T
LAST DIFF CHANGE
+7.61 %
at Block #977.760
DIFF FORECAST
-4.95 %
at Block #979.776
NEXT DIFF
13d 13h
1.855 blocks remaining
BLOCK HEIGHT
977.922
BTC PRICE
$59,539
54.533 €
upd. 1m ago
CHAIN
main
Bitcoin Core 31.3.0
BLOCK
Block assembly checked by Core: OK (9m ago, height 977922)
ACTIVE WORKERS
USER
HASHRATE
DIFF
SHARES
LAST SEEN
bc1qtg…eqz0.WarpMiner
224.4 KH/s
0.009
0.36 /min
0s ago
bc1qtg…eqz0.bitaxe602
1.11 TH/s
2.84k
5.44 /min
6s ago
bc1qtg…eqz0.NerdQaxe++
4.90 TH/s
11.63k
5.89 /min
6s ago
bc1qtg…eqz0.BitForgeNano
1.83 TH/s
6.21k
4.11 /min
18s ago
15Hifh…TMUY
5.51 TH/s
14.44k
5.33 /min
40s ago
bc1qtg…eqz0.NerdNOS
189.5 GH/s
1.04k
2.56 /min
57s ago
bc1qk0…syhp.TimsNerdAxeGamma
1.07 TH/s
5.51k
3.00 /min
1m ago
bc1q4y…zspt.LilyGO
296.7 KH/s
0.005
0.50 /min
2m ago
1Bitco…9kuE
–
–
–
46m ago
All workers →
BEST SHARES
All Time
Per Worker
Rank
Worker
Difficulty
When
Hash
bc1qtg…eqz0.bitaxe602
42.92G
7/23/2026
199dee6cefbe…3c9ca18fcc56
bc1qtg…eqz0.NerdQaxe++
19.53G
11/17/2026
384af3541a97…c1afd81f916c
bc1qtg…eqz0.NerdQaxe++
11.51G
11/27/2026
5f8e5adef237…f55da7a8b992
#4
bc1qtg…eqz0.NerdOctaxe
10.71G
8/13/2026
66a824b0acd9…35508ca9498c
#5
bc1qtg…eqz0.NerdOctaxe
9.72G
8/14/2026
711d7240efea…3ae7aeb509eb
#6
15Hifh…TMUY
8.59G
9/27/2026
8002a41fdc00…ad0d47121dbb
#7
bc1qtg…eqz0.AvalonQ
8.52G
7/17/2026
8113f47a8de1…757e58a6141d
#8
bc1qtg…eqz0.NerdQaxe++
7.05G
9/12/2026
9bf171c07dc0…5befad08c821
#9
bc1qtg…eqz0.NerdAxeGamma
6.89G
8/25/2026
9f942b44525b…25197368acfd
#10
bc1qk0…syhp.TimsNerdAxeGamma
6.62G
10/28/2026
a607b0f96e9f…ddfa06a6485f
Show all (50)



The public dashboard as an anonymous visitor sees it — live pool and network stats, a best-shares leaderboard with masked addresses (each miner keeps its rig label), and the optional donation card with QR codes at the bottom.

Announcing maintenance to your miners

A public pool has visitors you cannot reach on Discord. Under **Admin → Banner** you can put a short notice — a maintenance window, a planned node resync — as a scrolling strip along the top of the pool page, where everyone mining with you will see it. You set how long it should run (minutes, hours or days) and it disappears on its own when that time is over, so a forgotten notice cannot linger; a Stop button ends it early. The text is shown as plain text and is capped at 200 characters. Announcing planned downtime this way is simply good manners on a pool other people rely on.

7 · Public hardening profile

A pool exposed to the internet needs the abuse controls a friends-only pool can skip. Every key below already exists; none is per-worker-specific — they apply to any public deployment.

```
[stratum]
# Bound how many simultaneous V1 connections one source IP may hold. On a
# public pool this stops a single host from hogging slots. Leave 0 only if you
# expect many miners behind one NAT.
max_connections_per_ip = 50

[ratelimit]
enabled = true
connects_per_sec = 5
connect_burst = 20
auths_per_sec = 1.0
auth_burst = 10

# Auto-ban IPs that keep tripping the connect limit (escalating). Loopback and
# private LAN are never banned.
autoban_enabled = true
autoban_strikes = 3
autoban_base_secs = 60
autoban_max_secs = 86400
autoban_window_secs = 600

# Ban a worker that floods invalid shares (share-flood defence). Off by
# default; turn on for a public pool.
reject_ban_enabled = true
reject_ban_min_samples = 100
reject_ban_ratio = 0.9

# Optional subnet access control. `ban_cidrs` denies ranges outright;
# `allow_only = true` flips to a strict whitelist (only allow_cidrs connect).
# ban_cidrs = ["203.0.113.0/24"]
# allow_only = false
# allow_cidrs = ["192.168.0.0/16"]
```

> **IPv6 clients are counted by their /64 (since v1.50).** An internet line with IPv6 usually gets a whole block of addresses — a "/64", the first half of the address — and the device behind it may pick any address in that block. So that one client cannot slip past the limits by changing its address, every per-client limit above (connect and login rates, the auto-ban, the share-flood ban, `max_connections_per_ip`) counts a public IPv6 client by its /64. A ban you set by hand in **Admin → Profile → IP ban list** on an IPv6 address bans its whole /64; the list still shows and unbans the address as you entered it. **Unique-local** (`fc00::/7`) and **link-local** (`fe80::/10`) addresses count as your own LAN, like private IPv4 addresses: per address, and never auto-banned. For IPv4 nothing changes.

Node outages and your miners. If Bitcoin Core delivers no block template for 10 minutes, the pool disconnects every miner and refuses new connections until templates flow again, so that miners hash on their fallback pool instead of a stale job (on by default since v1.45; `[server]`

`node_unhealthy_disconnect` and `node_unhealthy_grace_secs`). Tell the people mining on your pool to set a fallback pool in their firmware.

Invite-only variant

Combine per-worker payouts with an **address allowlist** so only known addresses may mine. A miner whose payout address is not listed is rejected on both transports (V1 authorize and Sv2 channel-open):

```
[mining]
payout_mode = "per_worker"
address_allowlist = ["bc1qalice...", "bc1qbob..."]
```

An empty `address_allowlist` (the default) leaves the pool open to anyone. Note that `allow_only` / `allow_cidrs` gate *which networks* may connect, while `address_allowlist` gates *which payout addresses* may mine — the two are independent.

8 · TLS

Firmware that only trusts public certificate authorities needs a real certificate. The relevant keys live in `[stratum]`:

```
[stratum]
stratum_tls_listen = ":3334"
tls_cert_path = "/etc/dvb-warppool/tls/cert.pem"
tls_key_path  = "/etc/dvb-warppool/tls/key.pem"
tls_reload_secs = 300
```

Leave the two paths **unset** to use the daemon's auto-generated self-signed certificate — fine for NerdMiner/Bitaxe-class firmware that lets you pin it.

With the self-signed certificate, the pool's "How to connect your miner" panel also shows a **TLS key fingerprint (SHA-256)** line, a string such as `AB:CD:…` that identifies the pool's key. Firmware that pins a self-signed certificate by its fingerprint shows the same value before it trusts the pool — **WarpAxe 0.1.41 and later** does this and asks the miner's owner to confirm it — so your miners can compare the two. With a CA-signed certificate the line is not shown, because no such check is needed.

Renewals take care of themselves. A real certificate from Let's Encrypt is replaced every couple of months, and whatever issued it — Caddy, certbot, your own PKI — simply rewrites those two files. The pool notices that and starts serving the new certificate on its own. You do not have to restart it, and miners that are connected at the time stay connected: they keep the certificate they already agreed on, and only new connections get the fresh one. `tls_reload_secs` is how often the pool looks (300 seconds by default; set it to `0` to switch the check off). If a renewal is caught half-written, the pool keeps serving the old certificate and tries again on the next check rather than dropping the listener.

> **Stratum V2.** If you also run Stratum V2, it listens on its own `sv2_listen` address (for example `:34254`). The Sv2 transport is always NOISE-encrypted, so it needs no separate TLS certificate. Miners additionally need the pool's NOISE authority public key to connect at all. Miners choose between a *standard* and an *extended* channel in their own firmware; point them at *extended*, because it lets them verify the coinbase they are hashing. Both are accepted here and neither needs configuring on your side.

> **One port for everything.** With `single_port = true` in `[stratum]`, the plain port (`[server] pool_listen`, 3333) also takes Stratum V1 over TLS and Stratum V2. Each connection is told apart by its first bytes and handed to the matching server, with the same limits, bans and counters as on that protocol's own port. TLS and Sv2 still need their own listeners configured as above, and those ports keep running. Behind a TCP balancer the PROXY header comes first and is checked before the protocol is recognised (chapter 9). Off by default.

9 · Behind a reverse proxy

When the status UI is served over HTTPS through nginx or Caddy, set three keys in `[server]`:

```
[server]
status_public_url = "https://pool.example.com"
trust_proxy_headers = true    # so per-IP limits see the real client IP
cookie_secure = true          # UI is HTTPS
```

`status_public_url` names the pool's own public address: a miner whose pool setting points at that host counts as mining here (Miners tab, "Miner switched to another pool" alert). Set `cookie_secure = true` whenever the UI is reached over HTTPS.

> **Only trust proxy headers you control.** With `trust_proxy_headers = true` the daemon trusts the `X-Forwarded-For` header to identify the real client IP. Enable it **only** when a proxy you control actually sets that header — otherwise a client could **spoof its IP** straight past your per-IP limits and bans.

Stratum behind a TCP balancer: the PROXY protocol

The Stratum ports are raw TCP and go through an L4 balancer (HAProxy TCP mode, nginx `stream`), never through the HTTP proxy. Behind such a balancer the pool would only see the balancer's address — one misbehaving miner could get it banned. Since v1.45 the balancer can send the real client address first (PROXY protocol v1/v2, HAProxy `send-proxy-v2`, nginx `proxy_protocol on`); the pool reads it **only** from the addresses you list, which then **must** send it:

```
[server]
pool_listen = "127.0.0.1:13333"       # daemon on internal ports, the balancer owns 3333/333
[stratum]
stratum_tls_listen = "127.0.0.1:13334"
sv2_listen = "127.0.0.1:13336"
proxy_protocol = true
proxy_protocol_trusted = ["127.0.0.1"]   # the balancer's address(es)
```

Rate limits, bans, the per-IP cap and the Workers page then show the miner, not the balancer. Mind that every connection from a trusted address must carry the header — a local miner or sidecar connecting to the daemon's ports directly is dropped; point it at the balancer.

10 · Deployment checklist

Run through this before opening the pool to the public:

- ☐ `payout_mode = "per_worker"`, with **no** pool fee or operator donation (the daemon starts — it fails closed otherwise).
- ☐ Bitcoin Core reachable; consider `[node] template_source = "ipc"` for lower latency (needs Core ≥ 31 built with IPC — GBT stays the default).
- ☐ `[ratelimit] enabled = true` with `autoban_enabled` and `reject_ban_enabled` on.
- ☐ `max_connections_per_ip` set (unless your miners sit behind one NAT).
- ☐ TLS in front of Stratum (your own certificate for public-CA-only firmware).
- ☐ Reverse proxy: `trust_proxy_headers` and `cookie_secure` set correctly.
- ☐ Test a miner with `your-address.rig1`; confirm it appears at `/users/<address>` and that its shares are accepted.
- ☐ If you publish a landing page, keep the honesty guardrails: no "battle-tested"; the mainnet block is "not yet".
- ☐ Optional `[legal]` footer (terms/privacy links) for a public or commercial deployment.

11 · Support & links

Online guide	docs.warppool.org/public-pool-setup.html
Source & issues	git.warppool.org/dvb-projekt/dvb-WarpPool
Feature comparison	docs.warppool.org/comparison.html
Contact	dvbprojekt@gmx.de

dvb-WarpPool is free and open-source software (AGPL-3.0-or-later, with a commercial licence available). This manual describes Version 1.52.0. Screens may differ slightly between releases. Solo mining is inherently probabilistic; nothing here is a promise of reward. A public 0% pool is non-custodial — the pool never holds anyone's coins, and the finder is paid the full reward directly by the Bitcoin network.



Open, non-custodial, zero-fee — and may the finder keep it all.