◆ BLOCH PROTOCOL · GENESIS-3

Bloch Mining Manual

Mining BLCH on Genesis-3 mainnet · SHA-256d

A practical operator's guide: pool and solo mining, hardware setup, reward mechanics, and chain verification for the Bloch Genesis-3 mainnet. **Honest scope:** this is a young, unaudited, ownerless proof-of-work network — this manual tells you how to mine it, not why you should.

2026-07-30 · v1.0

genesis #0 c7522d0e…aff348

SECTION 1

What you're mining

Bloch Genesis-3 is a pure proof-of-work chain with no owner, no foundation and no privileged keys. Its proof-of-work is **SHA-256d** — the same double SHA-256 that Bitcoin ASICs compute — encoded **little-endian from height 0**, so standard SHA-256 ASICs and Stratum miners work natively, with no firmware changes.

NETWORK FACTS — VERIFY THESE YOURSELF, DON'T TRUST THIS PAGE	
Chain	**Bloch Genesis-3 mainnet** · chain-id `0xB10C_0004`
Proof-of-work	**SHA-256d** (double SHA-256, little-endian from height 0, ASIC-native)
Block time	**30 s** target · difficulty retargets every **60 blocks**
Genesis #0 hash	`c7522d0ef29fe67463be45a8095db7f5e23b9542dde867363ea3131647aff348`
Block reward	**8,400 BLCH** · halving every **1,036,800** blocks (≈360 days) · **100 BLCH** perpetual tail
Address format	`bloch1q…`
Pool stratum	`stratum+tcp://stratum.posternpool.com:3333`
Explorer	`https://blochl1.com`
Public RPC	`https://g2rpc.posternpool.com` (JSON-RPC over HTTPS POST)

ONE RULE ABOVE ALL

Any node or pool claiming to be Genesis-3 must serve a block #0 that hashes to **c7522d0e…aff348**. If it doesn't, you are mining the wrong chain, whatever the branding says. Section 7 shows the one-line check.

› In this manual

SECTION 2

Quick start — pool mining

The fastest way in. Postern Pool is a convenience front-end run by Postern Labs that forwards work from a Genesis-3 node; the chain itself stays ownerless. If you'd rather not trust a pool, skip to §4 and solo-mine.

1 Get a Bloch payout address

You need a mainnet address in the `bloch1q…` format. Create one with the `bloch-cli` wallet (built alongside the node, see §4) or any Genesis-3 wallet you control the keys for. The pool pays this address directly — there is no account signup.

2 Point your miner at the stratum endpoint

```
URL:       stratum+tcp://stratum.posternpool.com:3333
Username:  bloch1qYOURADDRESS        # optionally bloch1qYOURADDRESS.rigname
Password:  x                         # any value is accepted
Algorithm: sha256d
```

3 Confirm shares, then confirm the chain

Your miner should report accepted shares within a minute or two. Then open **https://blochl1.com** and check that block heights are climbing roughly every 30 seconds — that's the network you're feeding.

IMPORTANT — USE THE STRATUM SUBDOMAIN

The bare apex `posternpool.com:3333` does **not** carry raw stratum — it is a Cloudflare-fronted website and TCP stratum will time out there. Miners must connect to **stratum.posternpool.com:3333**.

› Direct / fallback endpoints

If DNS is unavailable or your firmware dislikes hostnames, these reach the same node by IP:

ENDPOINT	ROLE
`192.248.190.123:3333`	Node stratum (same backend as the hostname)
`192.248.190.123:3335`	Pool-proxy with **vardiff** — for ASICs that need variable difficulty
`192.248.190.123:3336`	Pool-proxy with **vardiff** (second instance)

SECTION 3

Hardware setup

Anything that computes SHA-256d and speaks Stratum V1 can mine Genesis-3. In practice that means Bitcoin-class ASICs are the serious tier; GPUs are marginal; CPUs are for testing your plumbing, not for earning.

› ASIC (Bitmain / Antminer-style panel)

Any SHA-256 ASIC (Antminer S-series, Whatsminer, Avalon, …) works with stock firmware. In the miner's web panel, configure a pool entry:

PANEL FIELD	VALUE
Pool URL	`stratum+tcp://stratum.posternpool.com:3333`
Worker	`bloch1qYOURADDRESS` — optionally append `.rigname` to tell rigs apart
Password	`x`

If your ASIC's firmware struggles with the node's fixed share difficulty (high reject rates, difficulty warnings), point it at a vardiff proxy instead: `192.248.190.123:3335` or `:3336`. Same node, same payouts — the proxy adapts share difficulty to your hashrate.

› GPU — ccminer

```
ccminer -a sha256d \
  -o stratum+tcp://stratum.posternpool.com:3333 \
  -u bloch1qYOURADDR -p x -q
```

GPUs are orders of magnitude behind ASICs on SHA-256d. A GPU is a reasonable way to participate on a young chain and to validate your setup, but do not expect it to stay competitive if ASIC hashrate grows.

› CPU — cpuminer / minerd (testing only)

```
minerd -a sha256d \
  -o stratum+tcp://stratum.posternpool.com:3333 \
  -u bloch1qYOURADDR -p x
```

Useful to verify connectivity, share acceptance and payout wiring end-to-end. Against ASIC hashrate a CPU's expected block income is effectively zero — treat it as a diagnostic tool.

All three setups work identically against your own node — just swap the pool URL for your node's stratum address (§4). Nothing about the miner changes.

SECTION 4

Solo mining — run your own node

The trustless path: build the node from source, sync the chain yourself, and point your miner at your own stratum port. No pool, no intermediary — block rewards pay your address directly from your own coinbase.

› 4.1 Build from source

Requirements: Linux x86_64 (Ubuntu 22.04+ recommended), Rust ≥ 1.90 via rustup, plus `clang`, `cmake`, `pkg-config` (needed by RocksDB and the post-quantum crypto crates). 2+ vCPU, 4+ GB RAM, 20+ GB disk.

```
git clone https://github.com/tiagobeltraoacioli-sketch/bloch-sis-pow bloch
cd bloch
git checkout v0.3.0-genesis3
cargo build --release --features euvm --bin bloch
# binary lands in target/release/bloch
```

`--features euvm` compiles the native eUTXO VM, which is active from genesis on Genesis-3. Building from source is the strongest verification you can do; prefer it over any pre-built binary.

› 4.2 Get the carry-over snapshot

Genesis-3 opens with a carried-over ledger (413,743 UTXOs from the prior chain). The node **refuses to start without** `carryover.tsv`, and verifies it at boot against baked-in constants (SHAKE-256 root, UTXO count, total supply) — fail-closed. Download it from the downloads page linked at **https://blochl1.com** and sanity-check: size ≈ 50,062,903 bytes, 413,743 lines.

› 4.3 Run the node

```
./bloch --genesis3 --archive \
  --carryover-snapshot /path/to/carryover.tsv \
  --data-dir ~/.bloch-g3 \
  --listen /ip4/0.0.0.0/tcp/16110 \
  --rpc-bind 127.0.0.1 --rpc-port 16210 \
  --peer 192.248.190.123:16116
```

- `--genesis3` is **required** — without it the node runs a retired pre-Genesis-3 chain.
- `--peer 192.248.190.123:16116` dials the canonical archival seed; the node gossips and discovers more peers after the first connection. Repeat `--peer` to add seeds.
- Keep RPC on `127.0.0.1` — never expose it raw. Open **16110/tcp** inbound if you want to be dialable.
- Older seed IPs published before the relaunch (45.76.x, 45.77.x, 137.66.x) serve a **retired chain** — do not use them.

› 4.4 Sanity-check the genesis, then mine

After the node syncs, verify block #0 **before** pointing hashrate at it:

```
curl -s -X POST 127.0.0.1:16210 -H 'Content-Type: application/json' \
  -d '{"method":"getblockbyheight","params":[0],"id":1}'
# "hash" MUST equal:
# c7522d0ef29fe67463be45a8095db7f5e23b9542dde867363ea3131647aff348
```

A different hash means the wrong chain — wipe the data dir and re-sync from the canonical peer. Once verified, restart the node with mining flags:

```
./bloch --genesis3 --archive \
  --carryover-snapshot /path/to/carryover.tsv \
  --data-dir ~/.bloch-g3 \
  --listen /ip4/0.0.0.0/tcp/16110 \
  --rpc-bind 127.0.0.1 --rpc-port 16210 \
  --peer 192.248.190.123:16116 \
  --miner-address bloch1qYOURADDRESS \
  --stratum --stratum-addr 0.0.0.0:3333 --stratum-mode solo
```

Then aim your ASIC/GPU at `stratum+tcp://<your-host>:3333` with the same username/password convention as §2. Adding `--mine --mine-threads 2` instead runs the built-in CPU miner — fine for testing, slow for earning. For unattended operation, wrap the command in a systemd unit with `Restart=always` (see the Run-a-Node guide for a ready-made unit file).

SECTION 5

Rewards & economics

BLOCK REWARD
8,400 BLCH per block, from height 0

HALVING INTERVAL
1,036,800 blocks ≈ 360 days at 30 s

PERPETUAL TAIL
100 BLCH floor — emission never reaches zero

RETARGET
Every 60 blocks toward the 30 s target

At 30-second blocks the network mints ~2,880 blocks/day — about **24.2 million BLCH per day** in the first halving era, shared across all miners in proportion to hashrate. The reward halves every 1,036,800 blocks until it reaches the 100 BLCH perpetual tail, which continues indefinitely (so BLCH has a large nominal schedule but no absolute hard cap — the tail is a deliberate, Monero-style security subsidy).

› How difficulty behaves

Difficulty retargets every **60 blocks** (a ~30-minute window at target speed), moving up to 4× per window. On a young chain this matters in both directions: when a large ASIC lands on low difficulty it can burst hundreds of blocks in minutes until the retarget catches up; when hashrate leaves, blocks slow until the next downward retarget. Both are normal and self-correct within a few windows.

› Realistic expectations

Your long-run share of blocks equals your share of network hashrate. As of 2026-07-30 the network is doing on the order of **~94 TH/s** (live snapshot via `getchainstats` — check the current figure yourself, it will move). One modern ~200 TH/s ASIC would dominate today's network; one GPU at a few GH/s would statistically wait a very long time for a solo block, which is why small miners use the pool. Two honest caveats: hashrate on a young chain is volatile, and BLCH currently has no established market price — mine for participation and conviction, not projected income.

SECTION 6

Verify you're on the right chain

Three independent checks, strongest first. Do at least one before committing serious hashrate.

1 Genesis hash (definitive)

Against your own node (§4.4) or the public RPC:

```
curl -s -X POST https://g2rpc.posternpool.com -H 'Content-Type: application/json' \
  -d '{"method":"getblockbyheight","params":[0],"id":1}'
# hash must be c7522d0ef29fe67463be45a8095db7f5e23b9542dde867363ea3131647aff348
```

2 DAG health

```
curl -s -X POST https://g2rpc.posternpool.com -H 'Content-Type: application/json' \
  -d '{"method":"getdaginfo","params":[],"id":1}'
```

Healthy: `tip_count` small (ideally 1) and `tip_height` climbing roughly every 30 s across repeated calls. A large, static tip count or a frozen height means something is wrong — don't mine into it.

3 Explorer cross-check

Open **https://blochl1.com** and compare recent block hashes with what your node or miner reports. `getrecentblocks` over RPC gives the same data for scripted checks.

SECTION 7

Troubleshooting

SYMPTOM	CAUSE & FIX
Connection timed out on `posternpool.com:3333`	You're hitting the website apex — Cloudflare-fronted, it cannot carry raw stratum TCP. Use **`stratum.posternpool.com:3333`**, or the direct IP `192.248.190.123:3333` if DNS is the problem.
Shares accepted, but no blocks after launch or a hashrate jump	Normal when large hashrate meets low difficulty — the network briefly outruns the retarget. Difficulty adjusts every **60 blocks** (up to 4× per window) back toward 30 s. Keep mining; watch it climb via `getchainstats`.
High share reject rate (tens of percent)	Difficulty mismatch: your miner or a proxy is applying its own vardiff instead of the node's `set_difficulty`. Connect to the vardiff proxies `192.248.190.123:3335` / `:3336` instead.
Node prints a different genesis hash	Wrong or retired chain (old seed IP, or missing `--genesis3`). Wipe the data dir, restart with `--genesis3` and `--peer 192.248.190.123:16116`, re-run the §6 check.
Node refuses to start, complains about the snapshot	Missing or corrupted `carryover.tsv` — the node verifies its SHAKE-256 root, UTXO count and supply at boot, fail-closed. Re-download from the official downloads page and retry.
Miner says "unknown algorithm"	Genesis-3 is plain **sha256d**, same as Bitcoin. Use `-a sha256d` (not sha256, not sha3) — any Bitcoin-capable miner or firmware has it.

SECTION 8

Honest notes & disclaimer

- **Young, unaudited mainnet.** The node software has not undergone an independent security audit. Run it with the same caution you'd apply to any early network software.
- **Hashrate is currently concentrated.** A small number of machines supply most of today's hashrate. Until that decentralizes, deep reorganizations are possible; finality is depth-based, and confirmations should be treated as **reversible while the security ramp matures**. Size any economic reliance accordingly.
- **Postern Pool is a convenience, not the protocol.** The pool and its endpoints are an owned Postern Labs service and may change or go down. The Bloch protocol itself is ownerless: no company, foundation, or individual controls, owns, or represents it — including Postern Labs. Solo mining (§4) requires trusting no one.
- **BLCH is not a security.** It is a mined proof-of-work asset with no issuer, no promise of profit and no one obligated to support its value. It currently has no established market price. Mine because you want to run and secure open infrastructure — not on the expectation of return.
- **Verify everything.** Every hard number in this manual — genesis hash, chain-id, reward schedule, endpoints — can be checked against the chain itself with the commands in §6. Prefer your own node's answers over any document, including this one.

Bloch — Mining Manual · v1.0 · 2026-07-30. Facts verified against the live chain on the date of writing (genesis `c7522d0e…aff348`, tip height ~2,530 and climbing, ~94 TH/s). Endpoints and live figures are subject to change; the protocol constants are not, short of an open flag-day upgrade adopted by the network.