

Bloch — Node Setup Guide

Mainnet-beta node operators: `--peer` connectivity & mining · Ubuntu / systemd

This is the prescriptive setup for a Bloch mainnet-beta node that connects to peers and mines. Follow every step. Do not skip the firewall step — it is the most common cause of a node that "runs but never syncs."

1. Open your firewall

Bloch uses two ports. Open both. **TCP 16110 must be reachable inbound** or peers cannot serve you and you cannot serve them.

Port	Protocol	Purpose
16110	TCP	P2P — peer connections (libp2p)
16210	TCP	RPC — <code>bloch-cli</code> / dashboard

On Ubuntu with UFW, run exactly:

```
sudo ufw allow 16110/tcp
sudo ufw allow 16210/tcp
sudo ufw reload
```

If you are behind a cloud security group (AWS/GCP/etc.), open `16110/tcp` inbound there too. Keep `16210` (RPC) restricted to trusted sources — do not expose it to the public internet.

2. The systemd service

Create `/etc/systemd/system/bloch.service`. Use this `ExecStart` line as your reference — set the values in angle brackets to your own:

```
[Unit]
Description=Bloch node
After=network-online.target
Wants=network-online.target

[Service]
User=ubuntu
ExecStart=/home/ubuntu/bloch/target/release/bloch \
  --data-dir /home/ubuntu/bloch-data \
  --rpc-public \
  --rpc-api-key-file /home/ubuntu/bloch-rpc.key \
  --peer /ip4/<SEED_IP>/tcp/16110 \
  --max-peers 500 \
  --mine \
```

```
--miner-address <YOUR_BLOCH_ADDRESS>
Restart=always
RestartSec=5
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target
```

Then enable and start it:

```
sudo systemctl daemon-reload
sudo systemctl enable --now bloch
sudo journalctl -u bloch -f
```

3. Peer connectivity — the `--peer` flag

`--peer` takes a **libp2p multiaddr** in the form `/ip4/<ip>/tcp/16110`. Optionally pin the peer identity by appending `/p2p/<peerid>`:

```
--peer /ip4/203.0.113.10/tcp/16110
--peer /ip4/203.0.113.10/tcp/16110/p2p/<peerid>
```

Pass multiple `--peer` flags to seed from several nodes — do this; more reachable peers means faster, more resilient sync:

```
--peer /ip4/203.0.113.10/tcp/16110 \
--peer /ip4/198.51.100.7/tcp/16110 \
--peer /ip4/192.0.2.44/tcp/16110
```

Discovered peers are persisted to `known_peers.json` inside your `--data-dir` (e.g. `/home/ubuntu/bloch-data/known_peers.json`). If you ever need a clean re-bootstrap, stop the node, remove that file, and restart with fresh `--peer` flags.

4. Mining

Mining is enabled by two flags together:

- `--mine` — turns the miner on.
- `--miner-address <YOUR_BLOCH_ADDRESS>` — where block rewards go. Use your own address in the form `bloch1q...`.

Both are already in the reference `ExecStart` above. A node only mines productively once it is synced to the chain tip — if the miner appears idle, check sync first (Section 5).

5. Verify

Use `bloch-cli` against your RPC port. Confirm two things: your DAG is advancing and you have live peers.

```
bloch-cli dag
```

Check that `chain_length` is **climbing** and that `tip_blue_score` is advancing and tracks the network. A frozen `chain_length` means you are not syncing — recheck the firewall (Section 1) and your `--peer` targets.

```
bloch-cli peers
```

Confirm you have **established, reachable peers**. Zero peers means port `16110` is blocked or your seed addresses are wrong. Watch the logs while you debug:

```
sudo journalctl -u bloch -f
```

Checklist

- TCP `16110` open inbound (P2P) and `16210` reachable to you (RPC).
- `ExecStart` points at your real `--data-dir`, RPC key, and `bloch1q...` miner address.
- At least one — preferably several — `--peer /ip4/<ip>/tcp/16110` flags.
- `--mine` and `--miner-address` both present.
- `bloch-cli dag` shows `chain_length` climbing; `bloch-cli peers` shows established peers.