

AEQUITAS NODE-BETREIBER-ANLEITUNG

v1.0 · Juni 2026 · aequitas.digital

Vollstaendige Schritt-fuer-Schritt-Anleitung · Keine Vorkenntnisse noetig · ca. 20–30 Min.

Vor dem Start — Was du brauchst

1. Ein Aequitas-Konto: Du musst zuerst als Mensch auf Aequitas registriert sein. Installiere die Android-App, schliesse die biometrische Registrierung ab und notiere deine Wallet-Adresse. Ohne dies kannst du keine Validator-Belohnungen erhalten.
2. Ein GitHub-Konto (kostenlos): Erstelle eines auf github.com. Du brauchst es um den Aequitas-Code zu forken, damit Railway ihn deployen kann.
3. Ein Railway-Konto (kostenlos): Gehe zu railway.app und melde dich mit GitHub an. Railway ist eine Hosting-Plattform die deinen Node in der Cloud betreibt — kein eigener Server oder Terminal erforderlich.
4. Node Signing-Key (RELAYER_PRIVATE_KEY): Dein Node braucht eine dedizierte Ethereum-Wallet zum Signieren. Das kann jede MetaMask-Wallet sein. Exportiere den privaten Schluessel: MetaMask → Kontodetails → Privaten Schluessel anzeigen → Passwort eingeben → kopieren. Streng geheimhalten. WICHTIG: Um Validator-Belohnungen zu erhalten, muss NODE_OPERATOR_WALLET deine registrierte Aequitas-Mensch-Wallet sein (die mit AequitasBio verifizierte). Nur verifizierte Menschen koennen Validator-Belohnungen verdienen.
5. 10–30 Minuten deiner Zeit. Railway erledigt den Grossteil automatisch.

Schritt 1 — Umgebungsvariablen

Sicherheitswarnung: Dein RELAYER_PRIVATE_KEY ist wie ein Master-Passwort. Wer ihn hat, kontrolliert deine Node-Wallet. Niemals oeffentlich teilen, niemals in Chat oder E-Mail einfuegen. Verwende fuer RELAYER_PRIVATE_KEY eine separate Wallet. NODE_OPERATOR_WALLET (fuer Belohnungen) muss deine registrierte Aequitas-Mensch-Wallet sein.

Variable	Erforderlich?	Was eintragen
DATABASE_URL	JA	Dein PostgreSQL-Verbindungsstring. Auf Railway: automatisch gesetzt wenn PostgreSQL im gleichen Projekt. Format: postgres://user:pass@host:5432/dbname
RELAYER_PRIVATE_KEY	JA	Privater Schluessel deiner Node-Wallet (0x..., 66 Zeichen). MetaMask: Kontodetails → Privaten Schluessel anzeigen → Passwort → kopieren.
RELAYER_ADDRESS	Empfohlen	Wallet-Adresse (0x..., 42 Zeichen) passend zu RELAYER_PRIVATE_KEY. Aus MetaMask kopieren. Verhindert Startfehler.
NODE_OPERATOR_WALLET	Fuer Bel.	Deine Aequitas-Mensch-Wallet — die via Android-App registrierte. Erhaelt taeglich Validator-Belohnungen (40% aller Protokollgebuehren). Muss ein registrierter Mensch sein.
NODE_OPERATOR_BINDING_SIGNATURE	Fuer Bel.	Beweist dass dir NODE_OPERATOR_WALLET gehoert. Erzeugen unter aequitas.digital/node-binding: angezeigte Nachricht mit deiner Mensch-Wallet in MetaMask signieren, Signatur hier eintragen. Ohne sie laeuft der Node trotzdem, kann sich aber nicht fuer Belohnungen registrieren.

Variable	Erforderlich?	Was eintragen
PEER_SECRET	Optional/Legacy	Legacy-Fallback. Nicht mehr erforderlich — Nodes authentifizieren sich automatisch per Challenge-Response (RELAYER_PRIVATE_KEY). Nur fuer Rueckwaertskompatibilitaet mit aelteren Deployments benoetigt.
SELF_URL	Multi-Node	Eigene oeffentliche HTTPS-URL des Nodes (z.B. https://mein-node.up.railway.app). In Railway: Settings → Networking → Public Networking.
PRIMARY_NODE_URL	Multi-Node	Auf https://aequitas.digital setzen — der Primaer-Node bei dem sich dein Node registriert. Beim Start postet der Node URL + Signing-Adresse und bekommt die Peer-Liste zurueck.
BOOTSTRAP_SNAPSHOT_URL	Empfohlen	Auf https://aequitas.digital/api/snapshot setzen — laesst einen neuen Node mit dem aktuellen Netzwerk-Stand starten statt die gesamte Historie ab Genesis nachzuspielen. Deutlich schnellerer Erststart.
BOOTSTRAP_SIGNER	Mit Snapshot	Signing-Adresse des Primaer-Nodes — prueft die Echtheit des Snapshots vor dem Import. Aktuellen Wert unter https://aequitas.digital/api/status → "signing_address" finden. Erforderlich wenn BOOTSTRAP_SNAPSHOT_URL gesetzt ist.
SNAPSHOT_TOKEN	Optional	Nicht erforderlich zum Bootstrappen eines neuen Nodes — auch ohne erhaeltst du alles Noetige (Accounts, Salden, Pool, Config). Schaltet nur den vollen Export frei (Nullifier/Wallet-Verknuepfung + bio_registrations) fuer den autoritativen Resync eines bereits divergenten Nodes. Beim Netzwerkbetreiber erfragen falls noetig.
RESYNC_FROM_SNAPSHOT	Nur Recovery	GEFAEHRlich, temporaer: nur zusammen mit BOOTSTRAP_SNAPSHOT_URL und BOOTSTRAP_SIGNER setzen, um einen vom Netzwerk abgewichenen Node zu reparieren. Ersetzt den lokalen Zustand komplett. Einmal neu starten, dann diese Variable wieder entfernen — sonst erfolgt bei jedem Neustart ein voller Resync.
AUTO_HEAL_ON_DIVERGENCE	Dringend empfohlen	Wichtig fuer Sicherheit und Geschwindigkeit des Netzwerks: Wenn die Chain deines Nodes je vom Netzwerk abweicht (z.B. nach Downtime oder einem fehlgeschlagenen Neustart), erkennt er das selbststaendig — durch regelmaessigen Abgleich mit PRIMARY_NODE_URL alle paar Minuten — und synct sich automatisch neu, ganz ohne manuelles RESYNC_FROM_SNAPSHOT. Ein abweichender Node, der weiter eigene Bloecke verbreitet, kann das gesamte Netzwerk fuer alle anderen Betreiber verlangsamen oder destabilisieren. Zusammen mit PRIMARY_NODE_URL, BOOTSTRAP_SNAPSHOT_URL und BOOTSTRAP_SIGNER auf true setzen.
PORT	Nein	Auf Railway nicht setzen — wird automatisch gesetzt. Standard ist 8080.
NODE_KEY	Nein	Base64 libp2p-Schlüssel fuer stabile Peer-Identitaet. Auto-generiert wenn nicht gesetzt, aendert sich dann bei jedem Neustart. Beim ersten Start in stderr ausgegeben: "SAVE THIS AS NODE_KEY: ". Kopieren und hier setzen.
IS_PRIMARY_NODE	Nein	Nicht setzen oder false lassen. Die Ausschuetzung nutzt jetzt einen DB-Lock — jeder Node kann sie ohne diese Variable ausfuehren.
RESET_STATE	Nein	GEFAEHRlich: True loescht die gesamte DB bei jedem Neustart. Nur fuer Entwicklung. Niemals in Produktion.

Schritt 2 — Deployment auf Railway (Empfohlen)

Railway ist der einfachste Weg deinen Node zu betreiben — kein Server-Setup, kein Terminal erforderlich. Der kostenlose Tarif deckt alle Anforderungen. Gesamtzeit: ca. 10–15 Minuten.

- 1 github.com/hanoi96international-gif/Aequitas forken (eigenes GitHub-Konto, Fork → Create fork)
- 2 Auf railway.app mit GitHub anmelden, dann New Project → + New → Database → Add PostgreSQL
- 3 Im selben Railway-Projekt: + New → GitHub Repo klicken und deinen Aequitas-Fork auswählen — Railway erkennt das Dockerfile automatisch
- 4 Deploy Now klicken — ein erster Build startet (kann ohne Env Vars fehlschlagen, das ist normal)
- 5 Aequitas-Service → Variables → Variablen hinzufügen (siehe Tabelle oben). Mindest-Anforderung: RELAYER_PRIVATE_KEY, RELAYER_ADDRESS, NODE_OPERATOR_WALLET, SELF_URL, PRIMARY_NODE_URL=<https://aequitas.digital> (PEER_SECRET nicht mehr erforderlich)
- 6 Deploy klicken (oder Variablen speichern für Auto-Redeploy). Build dauert ~3 Minuten für Go-Kompilierung.
- 7 Deploy-Logs beobachten. Erfolg sieht so aus: [Aequitas Node Running](#) und [\[NODE\] Registered node operator wallet: 0x...](#)
- 8 Settings → Networking → Generate Domain für deine öffentliche URL
- 9 <https://DEINE-URL/api/status> aufrufen — du siehst JSON mit height der alle ~6 Sekunden steigt

```
# Railway setzt DATABASE_URL automatisch wenn PostgreSQL im gleichen Projekt
RELAYER_PRIVATE_KEY = 0xDEIN_PRIVATER_SCHLUESSEL
RELAYER_ADDRESS     = 0xDEINE_NODE_WALLET_ADRESSE
NODE_OPERATOR_WALLET = 0xDEINE_MENSCH_WALLET
# PEER_SECRET ist nicht mehr erforderlich — Authentifizierung ist automatisch
SELF_URL            = https://DEIN-RAILWAY-DOMAIN.up.railway.app
PRIMARY_NODE_URL    = https://aequitas.digital
AUTO_HEAL_ON_DIVERGENCE = true
```

Schritt 2b — Alternative: Docker-Deployment (Fortgeschritten)

Nutze dies wenn du einen eigenen Server hast (VPS, Heimserver, Cloud-VM). Erfordert Docker und eine PostgreSQL-Datenbank.

```
# 1. Code herunterladen
git clone https://github.com/hanoi96international-gif/Aequitas && cd Aequitas

# 2. Node-Image erstellen (~3 Min für Go-Kompilierung)
docker build -t aequitas-node .

# 3. Node starten — alle Platzhalter ersetzen
docker run -d --name aequitas-node --restart unless-stopped \
-e DATABASE_URL="postgres://user:pass@host:5432/aequitas" \
-e RELAYER_PRIVATE_KEY="0xDEIN_PRIVATER_SCHLUESSEL" \
-e RELAYER_ADDRESS="0xDEINE_NODE_WALLET_ADRESSE" \
-e NODE_OPERATOR_WALLET="0xDEINE_MENSCH_WALLET" \
# -e PEER_SECRET="..." (optional/legacy, nicht erforderlich) \
-e SELF_URL="https://DEINE-OEFFENTLICHE-URL" \
-e PRIMARY_NODE_URL="https://aequitas.digital" \
-e AUTO_HEAL_ON_DIVERGENCE="true" \
-p 8080:8080 aequitas-node

# 4. Live-Logs beobachten
docker logs -f aequitas-node
```

Schritt 3 — Node-Betrieb pruefen

Oeffne diese URLs im Browser. Ersetze DEINE-NODE-URL durch deine Railway-Domain oder Server-Adresse.

```
https://DEINE-NODE-URL/api/status
→ Erwartet: {"height": 1234, "total_humans": N, "aequitas_index": N}

https://DEINE-NODE-URL/rpc
→ Erwartet: {"jsonrpc": "2.0", "error": "method not specified"} — RPC laeuft
```

Die Blockhoehe sollte innerhalb von Sekunden mit dem Primaer-Node uebereinstimmen (1–2 Bloecke). Bleibt sie bei 0: PRIMARY_NODE_URL=https://aequitas.digital pruefen.

Schritt 3b — Validator-Schluesel registrieren (Dezentrale Auth)

Statt eines gemeinsamen PEER_SECRET kannst du deinen Node-Signing-Key mit deiner Mensch-Wallet registrieren. Fuhre diesen Befehl auf deinem Server aus (SSH/Railway Shell):

```
curl "http://localhost:8080/api/sign-validator-challenge?wallet=0xDEINE_MENSCH_WALLET"
```

Dann auf der Website unter Network → Run a Node den Button "Sign with MetaMask & Register" nutzen um die Registrierung abzuschliessen.

Schritt 4 — MetaMask mit deinem Node verbinden (Optional)

In MetaMask: Netzwerk-Dropdown → Netzwerk hinzufuegen → Netzwerk manuell hinzufuegen:

Network Name	Aequitas Chain
RPC URL	https://DEINE-NODE-URL/rpc
Chain ID	1926
Currency Symbol	AEQ
Decimals	18
Block Explorer	https://aequitas.digital

Schritt 5 — Validator-Belohnungen erhalten

Der Validators-Pool sammelt 40% aller Protokollgebuehren (Swap-Gebuehren, Demurrage, Wealth-Cap-Ueberschuss). Jeden Tag um 20:00 Uhr Berliner Zeit (CEST/CET, DST automatisch) verteilt der Node den Pool-Saldo proportional nach produzierten Bloecken an alle registrierten Node-Betreiber. Je laenger dein Node laeuft, desto groesser dein Anteil.

- 1 Stelle sicher, dass du als Mensch auf Aequitas registriert bist. Falls nicht: Android-App installieren und biometrische Registrierung abschliessen. Du erhaeltst eine Wallet-Adresse und 1.000 AEQ.
- 2 NODE_OPERATOR_WALLET = deine Aequitas-Mensch-Wallet-Adresse in Railway Variables setzen
- 3 Speichern — Railway redeployt automatisch. Mit Docker: docker restart aequitas-node
- 4 In den Node-Logs bestaetigen: [NODE] Registered node operator wallet: 0x...
- 5 Belohnungen werden automatisch jeden Tag um 20:00 Uhr Berliner Zeit (CEST/CET) verteilt. Node laufen lassen — kein weiterer Eingriff noetig.

Fehlerbehebung

Symptom	Wahrscheinliche Ursache	Loesung
Blockhoehe bleibt bei 0	PRIMARY_NODE_URL nicht gesetzt	PRIMARY_NODE_URL=https://aequitas.digital setzen und neu deployen. SELF_URL auf Node-URL setzen.
DATABASE_URL-Fehler beim Start	Falscher Connection-String	Format pruefen: postgres://user:pass@host:5432/dbname — PostgreSQL muss erreichbar sein.
"no code at address" in Logs	V7-Contract noch nicht deployed	Normal beim ersten Start — Node deployed V7 automatisch. Kurz warten.
"NODE_OPERATOR_WALLET not set"	Fehlende Umgebungsvariable	NODE_OPERATOR_WALLET=0xDEINE_MENSCH_WALLE T hinzufuegen. Node laeuft ohne, aber keine Belohnungen.
Railway "Application error"	Build- oder Startfehler	Deploy-Logs pruefen. Haeufigste Ursache: fehlende DATABASE_URL oder falsches Schluessel-Format.
Port 8080 nicht erreichbar (Docker)	Firewall oder Cloud-Konfiguration	TCP-Port 8080 eingehend in Firewall oder Cloud-Security-Gruppe oeffnen.
Docker Build scheitert (Module)	Kein Internet beim Build	Docker Build benoetigt ausgehenden Internetzugang. Railway erledigt das automatisch.