

GonoPBX – Admin-Handbuch (kurz)

Stand: 13.02.2026

Ziel

Dieses Handbuch richtet sich an Administratoren und beschreibt Betrieb, Wartung und sichere Konfiguration von GonoPBX.

Überblick

GonoPBX besteht aus vier Docker-Services in einem internen Netzwerk:

- `pbx_frontend` (React/Nginx, Port 3000)
- `pbx_backend` (FastAPI, Port 8000)
- `pbx_asterisk` (Asterisk/PJSIP, Port 5060, AMI intern auf 5038)
- `pbx_postgres` (PostgreSQL, intern)

Die Asterisk-Konfigurationen werden automatisch aus der Datenbank generiert und nach Änderungen neu geladen.

Installation (Kurzfassung)

Voraussetzungen: Linux, Docker Engine 24+, Docker Compose v2.

1. Projekt nach `/root/asterisk-pbx-gui` legen.
2. Installation starten:

```
cd /root/asterisk-pbx-gui
./install.sh
```

Für ein schnelleres Server-Deployment ist auch `./deploy.sh` vorhanden.

Erstzugriff

Die Web-GUI läuft auf Port 3000. Beispiel:

`http://<SERVER-IP>:3000`

Der Admin-Benutzer wird beim Start automatisch angelegt oder aktualisiert:

- Benutzername: `admin`
- Passwort: aus `.env` (`ADMIN_PASSWORD`)

Wichtige Ports & Firewall

Freizugeben (Inbound) für ein Standard-Setup:

- SIP: 5060 UDP/TCP
- RTP: 10000-10100 UDP
- Web-GUI: 3000 TCP

- API: 8000 TCP

Optional (nur intern): AMI ist auf 127.0.0.1:5038 gebunden.

Betrieb: Start, Stop, Status

```
cd /root/asterisk-pbx-gui
```

```
# Status
```

```
docker compose ps
```

```
# Logs
```

```
docker compose logs -f
```

```
# Start/Stop/Restart
```

```
docker compose up -d
```

```
docker compose down
```

```
docker compose restart
```

Konfiguration & Pfade

Die wichtigsten Pfade auf dem Host:

- `.env` – zentrale Konfiguration und Secrets
- `asterisk/config/` – generierte Asterisk-Konfigs (Host-Mount)
- `backend/uploads/` – Benutzer-Uploads (Avatare)
- `asterisk/` – Asterisk Dockerfile und Assets

Wichtige Pfade im Asterisk-Container:

- `/etc/asterisk/custom/` – generierte Configs (Volume)
- `/var/spool/asterisk/voicemail` – Voicemails (Volume)

Benutzer & Rollen

- Rollen: `admin` und `user`
- Admin-Funktionen (nur `admin`): Benutzerverwaltung, Systemeinstellungen, SIP-Debug

Passwort-Reset für `admin`:

1. `ADMIN_PASSWORD` in `.env` anpassen
2. Backend neu starten

```
docker restart pbx_backend
```

Der Admin-Hash wird beim Start mit `ADMIN_PASSWORD` synchronisiert.

SIP-Nebenstellen (Extensions)

- Nebenstellen sind PJSIP-Peers in der Datenbank
- Änderungen regenerieren `pjsip.conf` und reloaden Asterisk
- Passwort-Stärkeprüfung und Passwort-Generator vorhanden

Trunks & Routing

- Trunks unterstützen Registrierung oder IP-Auth
- Inbound-DIDs werden Extensions zugeordnet (Dialplan-Generierung)
- Outbound-CID Auswahl pro Extension möglich
- Optional: P-Asserted-Identity (PAI)

Voicemail

- Mailboxen werden automatisch für Extensions angelegt
- Konfiguration wird aus der DB generiert
- Ringzeit pro Extension konfigurierbar

SMTP-Einstellungen

SMTP-Settings werden in der GUI gepflegt und erzeugen `msmtplib`-Konfiguration im Asterisk-Container. Test-E-Mail ist verfügbar.

Home Assistant & MQTT

- Optional: API-Key Authentifizierung
- MQTT-Publisher für Call-Events
- Konfiguration in den Systemeinstellungen

Sicherheit

Empfohlene Maßnahmen:

- `BIND_ADDRESS=127.0.0.1` bei Reverse-Proxy-Setup
- Starkes `ADMIN_PASSWORD`, regelmäßig rotieren
- SIP-Whitelist aktivieren (IP-Whitelist)
- Fail2Ban-Integration aktivieren (Host muss Fail2Ban bereitstellen)

Updates

In der GUI: **Settings** → **Server** → **Update**.

Manuell:

```
cd /root/asterisk-pbx-gui
git pull origin main
docker compose up -d --build
```

Backups

Wichtige Daten liegen in Docker-Volumes:

- `postgres_data` – Datenbank
- `asterisk_voicemail` – Voicemails
- `user_uploads` – Uploads/Avatare

Beispiel-Backup (lokales Verzeichnis `./backups`):

```
mkdir -p /root/asterisk-pbx-gui/backups
```

DB

```
docker run --rm -v postgres_data:/data -v /root/asterisk-pbx-gui/backups:/backup alpine \
  sh -c "cd /data && tar czf /backup/postgres_data.tgz ."
```

Voicemails

```
docker run --rm -v asterisk_voicemail:/data -v /root/asterisk-pbx-gui/backups:/backup alpine \
  sh -c "cd /data && tar czf /backup/asterisk_voicemail.tgz ."
```

Uploads

```
docker run --rm -v user_uploads:/data -v /root/asterisk-pbx-gui/backups:/backup alpine \
  sh -c "cd /data && tar czf /backup/user_uploads.tgz ."
```

Restore erfolgt analog durch Entpacken in das jeweilige Volume (Container gestoppt).

Troubleshooting

Backend-Health:

```
curl -f http://localhost:8000/api/health
```

Asterisk-CLI:

```
docker exec -it pbx_asterisk asterisk -rvvv
```

Häufige Probleme:

- Web-GUI leer: Backend-Logs prüfen, `pbx_backend` neu starten
- SIP-Registrierung schlägt fehl: Ports/Firewall prüfen, Asterisk-CLI nutzen
- Keine Audio-Streams: RTP-Ports (10000-10100/UDP) und `EXTERNAL_IP` prüfen

Datei- und Service-Referenz

- `docker-compose.yml` – Services, Volumes, Ports
- `install.sh` – Erzeugt `.env`, setzt Passwörter, startet Container

- `backend/` – API, Auth, Config-Generatoren
- `frontend/` – Web-UI
- `asterisk/` – Asterisk-Konfiguration und Assets