Linux Bridge Setup Guide

Bridge MVP Setup

1) Android SDK + environment
Use this SDK path:

```bash
export ANDROID_HOME="$HOME/Android/Sdk"
export ANDROID_SDK_ROOT="$ANDROID_HOME"
export PATH="$ANDROID_HOME/platform-tools:$ANDROID_HOME/cmdline-tools/latest/bin:$ANDROID_HOME/tools/bin:$PATH"
```

2) Linux agent install

```bash
cd /home/mykl/projects/linux_bridge/bridge_app
./scripts/install_agent.sh
```

Local run without systemd:

```bash
cd /home/mykl/projects/linux_bridge/bridge_app/linux_agent
python3 -m venv .venv
. .venv/bin/activate
pip install -r requirements.txt
export BRIDGE_PAIRING_SECRET='change-me'
export BRIDGE_JWT_SECRET='change-me'
export BRIDGE_RELAY_URL='wss://<relay-host>/ws'
export BRIDGE_RELAY_TOKEN='change-me-relay-token'
export BRIDGE_AGENT_ID='default-agent'
uvicorn bridge_agent.main:app --host 0.0.0.0 --port 8787
```

3) Persistent services (no open terminals required)

Create a user secrets file used by both systemd units:

```bash
mkdir -p ~/.config/bridge
chmod 700 ~/.config/bridge
cat > ~/.config/bridge/secrets.env <<EOF
BRIDGE_RELAY_TOKEN=<set-random-token>
BRIDGE_PAIRING_SECRET=<set-random-secret>
BRIDGE_JWT_SECRET=<set-random-secret>
BRIDGE_AGENT_ID=default-agent
EOF
chmod 600 ~/.config/bridge/secrets.env
```

Install and start user services:

```bash
cd /home/mykl/projects/linux_bridge/bridge_app
./scripts/install_relay.sh
./scripts/install_agent.sh
```

Manage services:

```bash
systemctl --user status bridge-relay.service
systemctl --user status bridge-agent.service
systemctl --user restart bridge-relay.service bridge-agent.service
```

Optional: keep user services alive after logout/reboot:

```bash
loginctl enable-linger "$USER"
```

4) Relay deploy (Option B)
Relay is in `relay/` and keeps only transient in-memory routing.

VPS (Ubuntu) quick deploy:

```bash
cd /opt
git clone <your-repo-url> bridge_app
cd bridge_app/relay
python3 -m venv .venv
. .venv/bin/activate
pip install -r requirements.txt
export BRIDGE_RELAY_TOKEN='change-me-relay-token'
uvicorn main:app --host 0.0.0.0 --port 8788
```

Use a reverse proxy (Caddy/Nginx) with TLS and route `/ws/*` to `127.0.0.1:8788`.

5) Pair once from Android
1. Call `POST /pair/init` on Linux agent with `X-Pairing-Secret`.
2. Put `pairing_token` into Android Pair screen (or QR payload parser).
3. Android calls `/pair/complete` once and stores JWT/device key.
4. Run allowlisted tasks from app; agent streams logs via websocket.

6) Dev convenience

```bash
cd /home/mykl/projects/linux_bridge/bridge_app
./scripts/dev.sh
```

This starts agent + relay and runs Android debug build.

7) One-tap phone testing

Pair ADB wireless once (Android 11+):

```bash
ADB="$HOME/Android/Sdk/platform-tools/adb"
$ADB pair <PHONE_IP:PAIR_PORT> <PAIRING_CODE>
$ADB connect <PHONE_IP:DEBUG_PORT>
$ADB devices -l
```

Stable debug signing (preserve app data across installs):

- This project uses `android_app/.keystore/debug.keystore` for debug signing.
- If you previously installed `com.bridge.app` signed with a different key, do a one-time uninstall:

```bash
ADB="$HOME/Android/Sdk/platform-tools/adb"
$ADB -s <PHONE_IP:DEBUG_PORT> uninstall com.bridge.app
```

- After that, repeated installs can use `adb install -r` and keep app data:

```bash
ADB="$HOME/Android/Sdk/platform-tools/adb"
$ADB -s <PHONE_IP:DEBUG_PORT> install -r android_app/app/build/outputs/apk/debug/app-debug.apk
```

Bridge task names now available:

- `android_build_debug`: builds `android_app` debug APK using `./gradlew :app:assembleDebug`.
- `android_install_debug`: installs `android_app/app/build/outputs/apk/debug/app-debug.apk` to connected phone.
- `android_launch`: launches package (`com.bridge.app` by default).
- `android_logcat`: streams filtered logcat for 45 seconds.
- `android_smoke`: build + install + launch + 45s logcat.

All scripts fail fast if no ADB device is connected.

Terminal tab (Android app):
- Full-screen command console runs Linux host commands via `POST /terminal/exec`.
- Requires pairing token/JWT (same app auth).
- Command timeout: 1-60 seconds.
- Output is capped/truncated to keep UI responsive.

EVA latency tuning (optional in `~/.config/bridge/secrets.env`):
- `BRIDGE_EVA_PRIMARY_TIMEOUT_SECONDS` (default `20`)
- `BRIDGE_EVA_COMPAT_TIMEOUT_SECONDS` (default `35`)
- `BRIDGE_EVA_COMPAT_TIMEOUT_DEV_SECONDS` (default `120`, used for `/dev` prompts)
- `BRIDGE_EVA_ENABLE_COMPAT_FALLBACK` (`true`/`false`, default `true`)

Sessions tab (Android app):
- Live tmux pane sharing + control over paired agent auth.

- Session inventory: `GET /terminal/sessions`
- Pane snapshot: `GET /terminal/session/{pane_id}/snapshot?lines=240`
- Live pane stream: `WS /terminal/session/{pane_id}/stream?token=...`
- Send input: `POST /terminal/session/{pane_id}/input`
- Create/reuse session: `POST /terminal/session/new`
- Control lock status: `GET /terminal/session/{pane_id}/control`
- Claim/release control: `POST /terminal/session/{pane_id}/control/{claim|release}`
- Agent host must have `tmux` installed and active sessions.

Dev helper to get a Bearer token for direct `/task/run` calls:

```bash
cd /home/mykl/projects/linux_bridge/bridge_app
set -a; source ~/.config/bridge/secrets.env; set +a

PAIR_JSON=$(curl -s -X POST http://127.0.0.1:8787/pair/init -H "X-Pairing-Secret: $BRIDGE_PAIRING_SECRET")
PAIR_TOKEN=$(python3 - <<PY
import json,sys
print(json.loads('''$PAIR_JSON''')['pairing_token'])
PY
)

COMPLETE_JSON=$(curl -s -X POST http://127.0.0.1:8787/pair/complete \
  -H 'Content-Type: application/json' \
  -d "{\"pairing_token\":\"$PAIR_TOKEN\",\"device_name\":\"dev-cli\",\"device_key\":\"dev-cli-key-1234567890abcdef\"}")

ACCESS_TOKEN=$(python3 - <<PY
import json
print(json.loads('''$COMPLETE_JSON''')['access_token'])
PY
)
echo "$ACCESS_TOKEN"
```

8) CI deploy gate (GitHub Actions)

Workflow file:
- `.github/workflows/deploy_internet_check.yml`

It runs `scripts/deploy_internet_check.sh` (including terminal session websocket upgrade validation).

Required repository secrets:
- `BRIDGE_AGENT_URL`
- `BRIDGE_RELAY_URL`
- `BRIDGE_PAIRING_SECRET`
- `BRIDGE_JWT_SECRET`
- `BRIDGE_RELAY_TOKEN`
- `BRIDGE_AGENT_ID`
- `BRIDGE_EVA_URL`