

TÀI LIỆU HƯỚNG DẪN

CẨM NANG CÀI ĐẶT & SỬ DỤNG

VOICESTUDIO BY DEBPALASH

ElevenLabs Thay Thế 100% Chạy Local: Nhân Bản Giọng Nói • Lồng Tiếng Video • Đọc Chính Tả • 646 Ngôn Ngữ

★ **GitHub Repository:** <https://github.com/debpalash/VoiceStudio> (~14.400 Stars)

👑 **Tác giả:** Palash Debnath ([debpalash](https://github.com/debpalash))

🌐 **Trang tải nhanh:** voicestudio.sh/install 📦 Trước đây tên là: *OmniVoice-Studio*

1. VOICESTUDIO LÀ GÌ & TẠI SAO ĐÁNG CHÚ Ý?

VoiceStudio là ứng dụng desktop mã nguồn mở, **chạy hoàn toàn local**, do **Palash Debnath** phát triển — được giới thiệu như một giải pháp thay thế ElevenLabs không cần gửi dữ liệu giọng nói lên cloud. Giọng nói, bản ghi và dự án của bạn không rời khỏi máy.

- **Nhân bản & thiết kế giọng nói:** voice cloning và voice design, hỗ trợ nhiều engine TTS khác nhau (IndexTTS 2.5, PocketTTS, CosyVoice, OmniVoice...).
- **Lồng tiếng video (dubbing):** giữ giọng người nói gốc khi dịch/lồng tiếng, có chế độ chỉ xử lý audio.
- **Đọc chính tả & phiên âm:** dictation và transcription qua WhisperX, Faster-Whisper, hoặc parakeet-mlx trên Apple Silicon.
- **646 ngôn ngữ:** phạm vi ngôn ngữ rộng cho cả TTS lẫn ASR (nhận diện giọng nói), tùy theo engine được chọn.
- **Nền tảng speech cục bộ:** ứng dụng khác có thể gọi vào tính năng dictation hoặc kết nối qua HTTP, WebSocket, JSON-RPC, CLI và MCP.
- **Kiến trúc:** vỏ desktop Tauri v2 (Rust) + giao diện React/Vite, giao tiếp với backend FastAPI cục bộ qua HTTP · SSE · WebSocket tại localhost:3900.

2. YÊU CẦU PHẦN CỨNG & CHUẨN BỊ

- **Dung lượng trống:** khoảng 10GB cho ứng dụng, môi trường Python và model weights (tải thêm ~2.4GB ở lần chạy đầu).
- **GPU (tùy chọn, khuyến nghị):** NVIDIA/CUDA tăng tốc trên Windows & Linux; Apple Silicon dùng MPS tự động; AMD/ROCm chỉ hỗ trợ trên Linux, trên Windows AMD chạy CPU-only.
- **macOS:** chỉ Apple Silicon (M1 trở lên) chạy được backend cục bộ — Intel Mac chỉ dùng được phần giao diện, phải trở về backend từ xa (xem mục 6).
- **Windows:** Windows 10 21H2 trở lên hoặc Windows 11, bản 64-bit.
- **(Tùy chọn) Tài khoản Hugging Face:** cần cho tính năng phân biệt người nói (diarization) và tải nhanh hơn một số model TTS lớn.

3. HƯỚNG DẪN CÀI ĐẶT CHI TIẾT THEO HỆ ĐIỀU HÀNH

Theo tài liệu chính thức trong docs/install/ của repo, có 2 cách cài cho mỗi nền tảng: dùng bản cài sẵn (MSI/DMG/ApplImage) hoặc build từ mã nguồn.

Windows 10 / 11 (x64)

Cách 1 — Một dòng lệnh (khuyến nghị): mở PowerShell thường (không cần Admin):

```
irm https://voicestudio.sh/install | iex
```

Lệnh trên tự cài Git, FFmpeg, uv và bun qua winget, rồi tự clone và build.

Cách 2 — Thủ công:

```
git clone https://github.com/debpalash/VoiceStudio.git
cd VoiceStudio
bun install
bun run desktop-prod
```

Cần cài trước: Git for Windows (winget install --id Git.Git -e), Python 3.11+ (winget install Python.Python.3.11), Microsoft C++ Build Tools (workload "Desktop development with C++"), Bun và Rust/Cargo (qua winget install Rust.Rustup). Lần chạy đầu tự tạo virtual env Python qua uv, đồng bộ dependency và tải model weights, hiển thị tiến trình ngay trên splash screen.

Cách 3 — Bản cài sẵn (MSI): tải file MSI mới nhất tại [trang Releases](#), chạy và làm theo trình cài đặt. GPU: chỉ tăng tốc bằng NVIDIA/CUDA (tự nhận driver, không cần cài riêng CUDA Toolkit); GPU AMD trên Windows luôn chạy CPU-only.

macOS (Apple Silicon)

Cách 1 — Một dòng lệnh:

```
curl -fsSL https://voicestudio.sh/install | sh
```

Cách 2 — Thủ công:

```
git clone https://github.com/debpalash/VoiceStudio.git
cd VoiceStudio
bun install
bun run desktop-prod
```

Cần cài trước: Xcode Command Line Tools (xcode-select --install), Python 3.11+ (brew install python@3.11), Bun và Rust/Cargo. Yêu cầu macOS 13.3 (Ventura) trở lên. GPU (Apple MPS) được kích hoạt tự động, không cần cấu hình gì thêm.

Cách 3 — Bản cài sẵn (DMG): tải DMG đúng kiến trúc máy (kiểm tra tại Apple menu → About This Mac): file **aarch64** cho Apple Silicon. **Lưu ý quan trọng:** Intel Mac chỉ chạy được phần giao diện — backend Python cục bộ không cài được vì PyTorch đã ngừng phát hành bản dựng cho Intel-Mac sau 2.2.x. Trên Intel Mac, chỉ dùng được VoiceStudio khi trở UI về một backend chạy từ xa trên máy khác (Settings → Sharing → Remote backend).

Nếu gặp lỗi Gatekeeper ("VoiceStudio cannot be opened because the developer cannot be verified") — do bản build chỉ ký ad-hoc, chưa được Apple notarise:

- Cách GUI: chuột phải (hoặc Control-click) vào VoiceStudio.app → Open → bấm Open lần nữa trong hộp thoại. (macOS 15: mở 1 lần rồi vào System Settings → Privacy & Security → Open Anyway.)
- Cách Terminal:

```
xattr -dr com.apple.quarantine "/Applications/VoiceStudio.app"
```

Linux

Cài từ mã nguồn (áp dụng chung cho các distro chính):

```
git clone https://github.com/debpalash/VoiceStudio.git
cd VoiceStudio
bun install
bun run desktop-prod
```

Lần chạy đầu tự tạo virtual env Python qua uv, đồng bộ dependency và tải model weights (~2.4GB); các lần chạy sau khởi động trong vài giây. Hoặc tải bản AppImage dựng sẵn từ trang Releases:

```
chmod +x VoiceStudio_*.AppImage
./VoiceStudio_*.AppImage
```

Lưu ý: một số distro mới dùng WebKitGTK 2.44/2.46 có lỗi compositing khiến cửa sổ Tauri hiển thị trắng toàn bộ, không lên giao diện — đây là vấn đề đã biết, theo dõi trong tài liệu troubleshooting của repo. GPU AMD (ROCm) chỉ được hỗ trợ trên Linux, không có trên Windows/macOS.

4. CÀI QUA DOCKER (KHÔNG CẦN GIAO DIỆN DESKTOP)

Phù hợp cho server không màn hình, máy có GPU rời, hoặc muốn triển khai bằng một lệnh duy nhất. Image chính thức: ghcr.io/debpalash/omnivoice-studio và palashdeb/omnivoice-studio trên Docker Hub (cùng image, cùng tag — tên image vẫn giữ omnivoice-studio dù repo đã đổi tên VoiceStudio).

Chạy trên GPU AMD (ROCm, Linux):

```
docker run -d --name omnivoice \
  --device /dev/kfd --device /dev/dri \
  -p 127.0.0.1:3900:3900 \
  -e OMNIVOICE_API_KEY="$OMNIVOICE_API_KEY" \
  -v omnivoice-data:/app/omnivoice_data \
  -v ~/.cache/huggingface:/root/.cache/huggingface \
  ghcr.io/debpalash/omnivoice-studio:rocm
```

Sau khi container chạy, mở trình duyệt tới địa chỉ được map ở cổng 3900 — backend được đóng gói sẵn trong image, giao diện phục vụ qua HTTP như một trang web thường. Có 2 kênh cập nhật: **:stable** (khuyến nghị dùng production) và **:preview** (bản build mới nhất từ nhánh main, luôn không thấp hơn :stable về phiên bản).

- Chạy trên WSL với AMD GPU: WSL lộ compute AMD qua /dev/dxg thay vì /dev/kfd và /dev/dri của Linux gốc — cần cài ROCm và librocdxg trong bản phân phối WSL, kiểm tra bằng rocminfo ở phía host trước.

5. CẤU HÌNH TOKEN HUGGING FACE (TUỖY CHỌN)

Cần thiết cho tính năng phân biệt người nói (pyannote/speaker-diarization-3.1, model bị khoá quyền truy cập) và giúp tải nhanh hơn một số engine TTS lớn.

- Cách khuyến nghị:** mở Settings → API Keys ngay trong ứng dụng — token được lưu vào SQLite mã hoá của VoiceStudio và cả vị trí chuẩn của huggingface_hub, mọi tiến trình con app khởi chạy đều tự nhận được.
- macOS/Linux (biến môi trường):** export HF_TOKEN=hf_... trong ~/.zshrc hoặc ~/.bashrc.
- Windows (PowerShell, không dùng setx):**
[Environment]::SetEnvironmentVariable("HF_TOKEN","hf_yourtokenhere","User") — rồi mở lại terminal. Không dùng lệnh setx vì cắt bớt giá trị dài hơn 1024 ký tự và không escape ký tự %.

6. LƯU Ý VẬN HÀNH & XỬ LÝ SỰ CỐ THƯỜNG GẶP

- **Chế độ Portable (Windows):** toàn bộ cài đặt — Python env, model, giọng nói, dự án — nằm gọn trong một thư mục cạnh file .exe, có thể copy sang máy khác kiểu USB-stick; chọn ở màn hình thiết lập lần đầu.
- **Backend từ xa (Remote backend):** dùng khi máy hiện tại không chạy được backend cục bộ (ví dụ Intel Mac) — cấu hình ở Settings → Sharing → Remote backend để trở UI sang một máy khác đang chạy VoiceStudio.
- **Lỗi Triton/torch.compile hết bộ nhớ (Windows, VRAM <16GB):** bật Settings → Performance → "Disable torch.compile (Windows)", hoặc set biến \$env:TORCH_COMPILE_DISABLE = "1" trước khi chạy bun run desktop-prod.
- **Tải wheel PyTorch CUDA bị đứt giữa chừng:** đây là lỗi mạng, không phải lỗi phần mềm — thử lại (VoiceStudio tự retry 5 lần), dùng VPN nếu mạng chặn CDN của PyTorch, hoặc tải sẵn 2 file wheel đúng máy rồi đặt vào thư mục wheel để cài offline.
- **Cẩn thận với bản sao/fork giả mạo:** chỉ nên tải và cài từ đúng repo chính thức github.com/debpalash/VoiceStudio hoặc trang tải nhanh voicestudio.sh — repo trước đây có tên OmniVoice-Studio, các link cũ được GitHub tự chuyển hướng sang tên mới.

Chúc bạn dựng xưởng xử lý giọng nói AI chạy hoàn toàn local thành công cùng VoiceStudio!

© Kênh AI Mỗi Ngày | Hướng dẫn AI & Linux Tự Động Hóa
Nội dung kỹ thuật tổng hợp từ tài liệu chính thức của repo github.com/debpalash/VoiceStudio.