Open source · runs locally

Podcast to Shorts for free, on your own computer

Cut a long podcast or interview into vertical clips with captions using only free, open-source tools: FFmpeg, Whisper and auto-editor. No watermark, no upload, no account.

VideoToShorts · https://videotoshortsconverter.com/workflows/podcast-to-shorts-free-local · checked November 19, 2026

COST
Free

TIME
About 15 minutes per clip once set up

LEVEL
Intermediate (command line)

HARDWARE
Any recent computer. Whisper runs on CPU; an NVIDIA GPU makes transcription faster.

What you need

FFmpeg: Cut, crop to 9:16 and burn in captions

Whisper: Speech to text, with word timings

auto-editor: Remove silences and dead air

Best for: Podcasters, Interview channels, Anyone who wants no watermark and full control.

Steps

1 Install the three tools

Install FFmpeg from ffmpeg.org (or your package manager), then install Whisper and auto-editor with pip. Whisper needs FFmpeg on your PATH.

```
pip install -U openai-whisper auto-editor
ffmpeg -version
```

2 Find the moments worth clipping

Transcribe the full episode once, then read the transcript and note the start and end times of 3 to 5 moments that stand on their own: a strong opinion, a surprising number, a clear how-to. Aim for 20 to 60 seconds each.

```
whisper episode.mp3 --model small --language en --output_format txt
```

3 Cut each moment out

Use the times you noted. Re-encoding (libx264) gives clean cuts at any timestamp.

```
ffmpeg -ss 00:12:30 -to 00:13:15 -i episode.mp4 -c:v libx264 -c:a aac clip.mp4
```

4 Remove pauses

auto-editor cuts silent sections and keeps a short margin around speech (0.2 seconds by default). It writes clip_ALTERED.mp4 next to the original.

```
auto-editor clip.mp4 --margin 0.2s
```

5 Crop to vertical 9:16

This takes a centered 9:16 crop of the frame and scales it to 1080x1920. If the speaker is off-center, shift the crop with the x offset, for example crop=ih*9/16:ih:200:0.

```
ffmpeg -i clip_ALTERED.mp4 -vf "crop=ih*9/16:ih,scale=1080:1920" -c:a copy vertical.mp4
```

6 Make short-form captions

Transcribe the finished clip (so timings match) with word timestamps and short lines, which suits vertical video.

```
whisper vertical.mp4 --model small --language en --word_timestamps True --
max_line_width 24 --max_line_count 1 --output_format srt
```

7 Burn the captions in

Style the captions with force_style. Alignment=2 puts them bottom-center; raise MarginV to keep them above the app's buttons.

```
ffmpeg -i vertical.mp4 -vf
"subtitles=vertical.srt:force_style='FontName=Arial,FontSize=14,Bold=1,Outline=2,Alignment=2,MarginV
-c:a copy final.mp4
```

8 Check before posting

Watch the clip once on a phone with sound off. Fix any misheard names in the .srt file and re-run the last step. Check that captions are not hidden behind the like and comment buttons.

Questions

Is this really free with no watermark?

Yes. FFmpeg, Whisper and auto-editor are open source and run on your machine, so nothing adds a watermark and nothing is uploaded.

Can it pick the best moments automatically?

Not on its own. You choose the moments from the transcript. AI clippers like Opus Clip or Vizard automate that step, with limits on their free plans.

Do I need a GPU?

No. Whisper runs on CPU; a GPU only makes transcription faster. The small model is a good balance of speed and accuracy.

Sources

- Whisper README and CLI options
- auto-editor options (--margin)
- FFmpeg filters (crop, scale, subtitles)

Commands checked against each tool's docs on November 19, 2026.