Pyannote Production Setup Checklist

Speaker diarization — who spoke when — on your own infrastructure

1 · The three-stage pipeline

#	Stage	Model	What it does
1	Segmentation	segmentation-3.0	Find speech + overlap in a 10 s sliding window (powerset)
2	Embedding	ResNet / WeSpeaker	Turn each solo voice into a numeric fingerprint
3	Clustering	agglomerative	Group matching fingerprints; speaker count discovered
–	Aggregation	—	Assemble a who-spoke-when timeline (RTTM)

2 · Self-host pyannote vs a hosted service

Criterion	Self-hosted pyannote	Hosted service
Software cost	Free (MIT license)	Per-hour / per-minute fee
Accuracy (DER)	Strong; tunable on your data	Often higher out of the box
Speed	~40x real time on one GPU	Vendor-managed, often faster
Data location	Stays on your servers	Sent to vendor cloud
Setup	GPU, models, gated token, tuning	An API key
Best when	On-prem; steady volume; can tune	Ship fast; top accuracy, no ops

3 · Pre-flight checklist before you ship

- ☐ Confirmed you actually need speaker labels — if you only need the words, use ASR, not diarization.
- ☐ Created a Hugging Face token and accepted user conditions for both segmentation-3.0 and speaker-diarization-3.1.
- ☐ Extracted clean mono audio at 16 kHz — the input the pipeline expects.
- ☐ Passed the speaker count (num_speakers) or a range (min/max_speakers) wherever the product knows it.
- ☐ Checked data residency: can the audio legally leave your servers, or must it stay on-prem?
- ☐ Reported DER honestly: stated the collar and whether overlapped speech was scored; compared like for like.
- ☐ Set expectations that overlap and short backchannels are the hardest cases and may be mislabelled.
- ☐ Built a human-correction path for records where a wrong speaker label has real cost (medical, legal).
- ☐ Considered tuning (clustering threshold δ) or fine-tuning the segmentation model on domain audio.
- ☐ Modelled cost at real batch volume (GPU-hours = audio-hours / realtime-factor) vs a hosted service.