

Date: 2025-09-01 (Mon) 
Prepared by: Tiong-Sik Ng 

English 

1. Summary of Accomplishments

- Built a bulk **filler speech generator** using `/synthesize`; uploads to S3 under category folders (확인/설명/공감/시간별기형) with fixed names (`01.wav`–`05.wav`) and a CSV index.
- Confirmed server returns **regional URLs** by default (presign optional).
- Drafted Lambda paths for **single-shot synthesis** (`synthesize_full`) and a **filler-now, full-later** option (orchestrator + worker + poller).

2. Issues & Risks

- The end-to-end glue is **not finalized** yet (app.py, Lambda, Contact Flow).
- **Amazon Connect minimum waits** (2s/1s) limit perceived snappiness unless we use full-file or fillers.
- Very long texts may require **higher HTTP/Lambda timeouts** for full synthesis.

3. Next Steps

- Finish the **one-shot path** end-to-end: app.py `/synthesize` → Lambda `synthesize_full` → Contact Flow play-once branch.
- (Optional) Add **filler-now/full-later** polling loop for better perceived latency.
- Then focus on **FishSpeech speedups** (warmup/AMP/compile, and evaluate a faster model if needed).

날짜: 2025-09-01 (월) 

작성자: Tiong-Sik Ng 

1. 금일 수행 사항

- `/synthesize` 기반 대량 보이스 필러 생성 스크립트 구현: S3에 카테고리별(확인/설명/공감/시간별기형) 폴더와 고정 파일명(`01.wav`–`05.wav`)으로 업로드, 인덱스 CSV 생성.
- 서버가 기본적으로 리저널 URL을 반환하도록 확인(프리사인 선택적).
- Lambda 단일 일괄 합성 경로(`synthesize_full`)와 필러 우선+완성 오디오 후속 옵션(오케스트레이터/워커/풀러) 설계 초안 정리.

2. 이슈 & 리스크

- app.py, Lambda, 콘택트 플로우의 엔드 투 엔드 연계가 **미완성**.
- Amazon Connect의 **최소 대기(2초)·호출(1초)** 제약으로 체감 지연 발생 → 단일 파일 재생 또는 필러 활용이 유효.
- 긴 문장 합성 시 **HTTP/Lambda 타임아웃 상향** 필요 가능.

3. 다음 단계

- **단일 합성 경로**(app.py `/synthesize` → Lambda `synthesize_full` → 플로우 단일 재생) 마무리.

- (선택) 필러 우선/완성 후 재생 폴링 루프 연결.
- 이후 **FishSpeech** 속도 개선(워밍업/AMP/컴파일, 필요 시 신규 모델 평가) 집중.