

AI 전화 발신 및 음성 합성 시스템 가이드 (AI Phone Call & TTS Control System)

시스템 개요 (System Overview)

이 시스템은 두 가지 주요 구성 요소로 이루어져 있습니다:

구성 요소	위치	역할
1. NIPA 클라우드 서버	NIPA 클라우드 상	핵심 AI 서비스(텍스트-음성 변환 FishSpeech, 챗봇, 전화 발신 API)를 실행합니다. AWS Connect, DynamoDB, S3 오디오 업로드를 처리합니다.
2. 로컬 제어 패널(Web UI)	로컬 PC	간단한 한/영 웹 인터페이스로 전화 발신, 인사말 생성, 참고 음성 업로드를 통한 TTS 테스트를 수행합니다.

로컬 인터페이스는 공용 도메인을 통해 NIPA 서버에 연결됩니다.

1. 서버 접속 (NIPA Cloud Access)

항목	세부 정보
링크 (Link)	https://nipa.nhncloud.com/
아이디 (ID)	nipa-gpu2025-668@ktcloud.com
비밀번호 (Password)	_insunetfc6276
서버 URL (Server URL)	https://honest-trivially-buffalo.ngrok-free.app

2. NIPA 서버 실행 (Starting the NIPA Server)

Step 1 — 서버 시작 (Starting the server)

The screenshot shows the N H N CLOUD interface. On the left sidebar, under the 'Sessions' section, there is a 'START' button. In the main area, the 'Resource Statistics' section displays resource usage for a 'H100E' group. Below it, the 'Sessions' list shows a running session named 'IgmDjhqQ-session' with a status of 'RUNNING'. A red box highlights the 'START' button for this session.

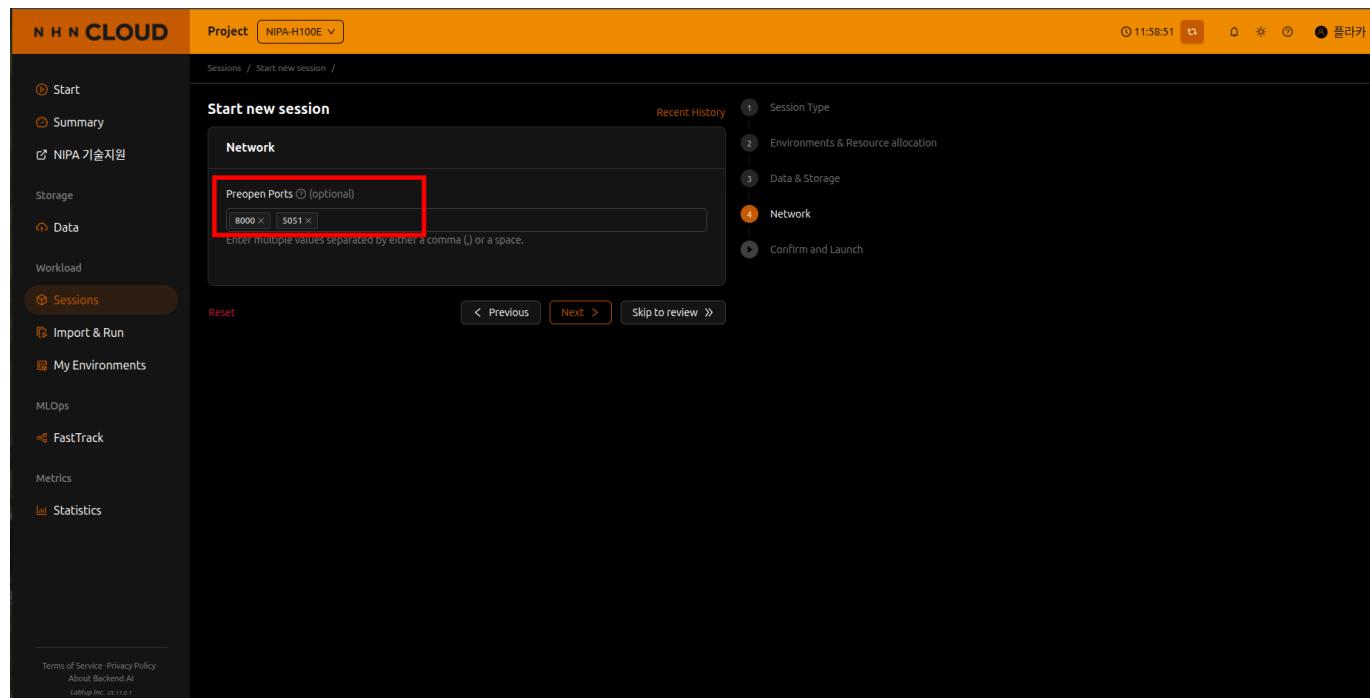
Step 2 — VALL-E 디렉터리 마운트 (Mount VALL-E directory)

VALL-E 디렉터리를 선택하여 마운트합니다.

The screenshot shows the 'Start new session' wizard in the N H N CLOUD interface. The current step is 'Data & Storage'. In the 'Data & Storage' section, there is a list of storage entries. One entry, 'VALL-E', has a checked checkbox next to it, indicated by a red box. Other entries include 'test' and another 'VALL-E' entry. The right side of the screen shows a 'Recent History' sidebar with steps 1 through 4: Session Type, Environments & Resource allocation, Data & Storage (which is highlighted), and Network.

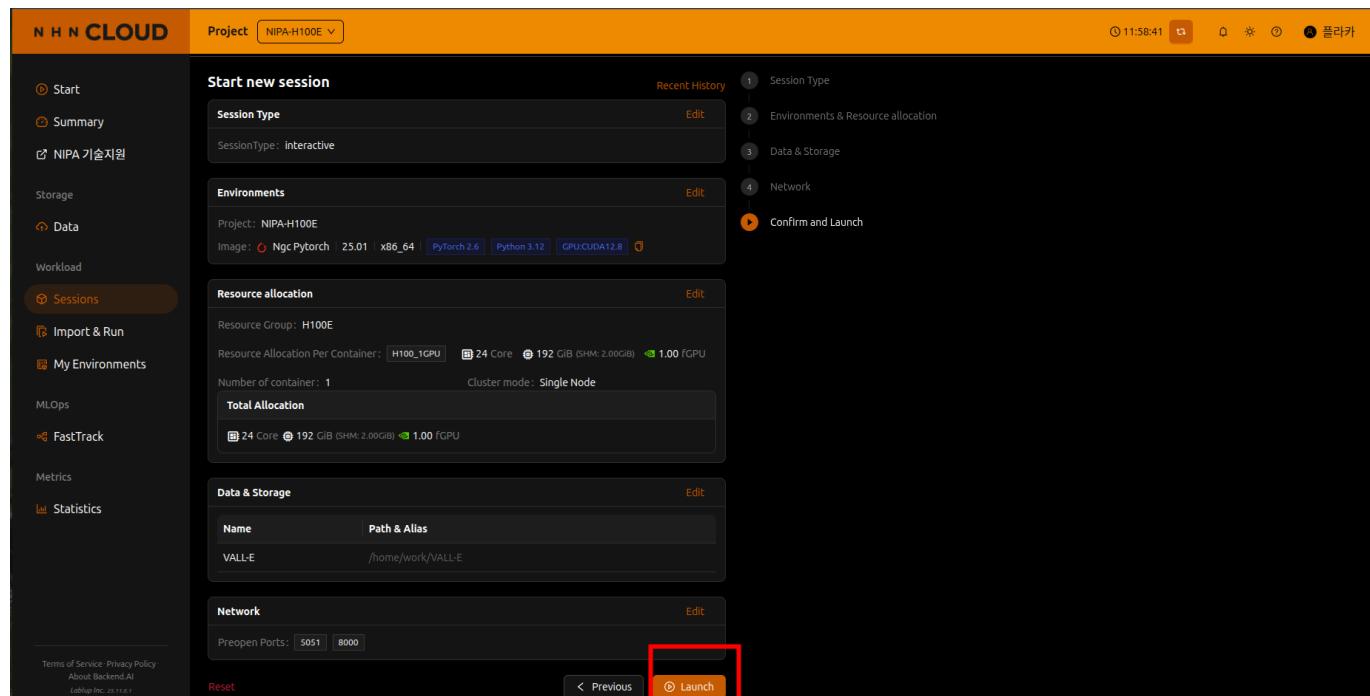
Step 3 — 사전 오픈 포트 선택 (Select pre-open ports)

포트 번호 8000과 5051을 입력합니다.



Step 4 — 서버 실행 (Launch the server)

Launch 버튼을 클릭하여 서버를 실행합니다.



Step 5 — 서버 설정을 위한 App Dialog 선택 (Selecting app dialog)

App Dialog 버튼을 클릭합니다.

The screenshot shows the NHN CLOUD interface with the 'Sessions' tab selected. A session named 'IgmDjhqQ-session' is listed under the 'RUNNING' tab. The 'See App Dialog' button next to it is highlighted with a red box.

Step 6 — JupyterLab 선택 및 실행 (Select and open JupyterLab)

JupyterLab을 선택합니다.

The screenshot shows the NHN CLOUD interface with the 'Sessions' tab selected. In the 'App' dialog, the 'Development' section is open, showing icons for Visual Studio Code, Visual Studio Code (Desktop), JupyterLab, and Jupyter Notebook. The 'JupyterLab' icon is highlighted with a red box.

Step 7 — NIPA 클라우드 서버 시작 (Start the NIPA cloud server)

다음 명령어를 실행합니다:

```
./VALL-E/run_server.sh
```

The screenshot shows a Jupyter Notebook interface with a sidebar containing a file browser and a main area with a terminal window.

File Browser:

- Path: /VALL-E /
- Items:
 - asr_bridge (1d ago)
 - audio_samples (23h ago)
 - chatbot (yesterday)
 - data (last mo.)
 - fishspeech (2d ago)
 - installations (2d ago)
 - miniconda3 (2mo ago)
 - miniconda3.bak (2mo ago)
 - nltk_data (2mo ago)
 - phone_call (43m ago)
 - app.py** (50m ago)
 - KsonSpeech_da... (5mo ago)
 - Miniconda3-lates... (3mo ago)
 - ngrok.log (22m ago)
 - requirements.txt (2mo ago)
 - run_server.sh** (2h ago)
 - stable-ngr... (2mo ago)

Terminal Window:

```
work@main1:~/gobjhq-session:~/VALL-E/run_server.sh
```

Step 8 — 서버 실행 완료 (Server launch complete)

서버 실행이 완료되면 다음과 같은 화면이 표시됩니다.

```
File Edit View Run Kernel Tabs Settings Help
+ Terminal 1 app.py +
work@win11:[~/VALL-E]$ ./VALL-E/run_server.sh
[!] direnv already installed.
[!] Allowing .envrc in /home/work/VALL-E
direnv: loading ./VALL-E/.envrc
Registering ngrok authtokens
[!] Allowing ./fishspeech | AWS ready
direnv: export +AWS_ACCESS_KEY_ID +AWS_REGION +AWS_SECRET_ACCESS_KEY +CONDA_DEFAULT_ENV +CONDA_EXE +CONDA_PREFIX +CONDA_PROMPT_MODIFIER +CONDA_PYTHON_EXE +CONDA_SHLVL +GSETTINGS_SCHEMA_DIR +GSETTINGS_SCHEMA_DIR_CC
DOS BACKUP +NGROK_AUTHTOKEN +NGROK_FIXED_DOMAIN +NGROK_LOG +NGROK_PORT +NGROK_TGZ +PROJECT_ROOT +UVICORN_APP +UVICORN_HOST +UVICORN_PORT +_CE_CONDA +_CE_M -PATH +PYTHONPATH
[!] Starting ngrok...
[!] ngrok log (last 20 lines):
[!] Starting ngrok...
/home/work/VALL-E/miniconda3/uvens/fishspeech/lib/python3.10/site-packages/torch/_init_.py:1617: UserWarning: Please use the new API settings to control TF32 behavior, such as torch.backends.cudnn.conv.fp32_precision = 'tf32' or torch.backends.cuda.matmul.fp32_precision = 'tf32'. Old settings, e.g., torch.backends.cuda.matmul.allow_tf32 = True, torch.backends.cudnn.allow_tf32 = True, allowTF32CuDNN() and allowTF32CuBLAS() will be deprecated after PyTorch 2.9. Please see https://pytorch.org/docs/main/notebooks/cuda.html#float32-tf32-on-amere-and-later-devices (Triggered internally at /pytorch/aten/src/Aten/Context.cpp:80.)
    set_fp32_precision(fp32_precision)
2025-11-07 02:23:33.549 [INFO] fishspeech.fish_speech.app:modules>:254 - Patched TTISInferenceEngine.inference_to use REF_AUDIO
2025-11-07 02:23:33.549 [INFO] fishspeech.fish_speech.app:modules>:254 - Patched TTISInferenceEngine.inference_to use REF_AUDIO
2025-11-07 02:23:33.519 [INFO] fishspeech.fish_speech.app: warmup:fishspeech:455 - [FishSpeech] Starting warmup...
2025-11-07 02:23:33.521 [INFO] fishspeech.fish_speech.app: init :1244 - Launching LLM queue.
2025-11-07 02:23:33.713 [INFO] fishspeech.models.text2semantic.llama:from_pretrained:432 - Loading model from /home/work/VALL-E/fishspeech/fish-speech/checkpoints/openaudio-s1-mini.config: dualLMModelArgs(
    vocab_size=155776, n_layer=28, n_head=8, dim=1024, intermediate_size=3072, n_local_heads=8, head_dim=128, rope_base=100000, norm_eps=0.06, max_seq_length=302, dropout=0.03, tie_word_embedding=False, fast_n_local_heads=8, fast_head_dim=64, fast_n_local_heads=8, fast_head_dim=64, fast_intermediate_size=3072, fast_n_local_heads=8, fast_head_dim=64, fast_attention_qkv_bias=False, fast_attention_qkv_norm=False, fast_atten...
ebook_embeddings=True, no_fast_layer4=True, fast_dim=1024, fast_n_local_heads=8, fast_head_dim=64, fast_intermediate_size=3072, fast_n_local_heads=8, fast_head_dim=64, fast_attention_qkv_bias=False, fast_attention_qkv_norm=False, fast_attent...
n_o_bias=False)
2025-11-07 02:23:40.032 [INFO] fishspeech.models.text2semantic.llama:from_pretrained:498 - Loaded weights with error: All keys matched successfully!
2025-11-07 02:23:40.032 [INFO] fishspeech.fish_speech.app: load_model:103 - Loading model from checkpoint
2025-11-07 02:23:40.032 [INFO] fishspeech.fish_speech.app: token_embedding:103 - Using D4UTrTransformer
2025-11-07 02:23:40.569 [INFO] fishspeech.fish_speech.app: init :1248 - Loading decoder model (D4UTr)
/home/work/VALL-E/miniconda3/uvens/fishspeech/lib/python3.10/site-packages/torch/mn/utils/weight_norm.py:144: FutureWarning: 'torch.nn.utils.weight_norm' is deprecated in favor of `torch.nn.utils.parametrizations.v...
eight_norm...
WeightedNorm.apply(proxy.module, name, dim)
2025-11-07 02:23:43.761 [INFO] fishspeech.models.dac.inference.load_model:46 - Loaded model: All keys matched successfully!
2025-11-07 02:23:43.761 [INFO] fishspeech.fish_speech.app: init :1251 - Constructing TTISInferenceEngine...
2025-11-07 02:23:44.838 [INFO] fishspeech.fish_speech.app:engine:1347 - torch.compile not available; continuing
2025-11-07 02:23:44.839 [INFO] fishspeech.fish_speech.app: load_target_se_onces:1364 - [FishSpeech] Loaded target SE from /home/work/VALL-E/audio_samples/cached_ref.pt
2025-11-07 02:23:44.840 [INFO] fishspeech.fish_speech.app:load_target_se_onces:1364 - [FishSpeech] Loaded target SE from /home/work/VALL-E/audio_samples/cached_ref.pt
2025-11-07 02:23:44.840 [INFO] fishspeech.fish_speech.app:patched_inference:149 - [PATCHED] Loading default embedding from /home/work/VALL-E/audio_samples/cached_ref.pt
2025-11-07 02:23:44.841 [INFO] fishspeech.fish_speech.app:patched_inference:150 - [PATCHED] Done loading embeddings. Using 1 prompt token(s).
2025-11-07 02:23:44.842 [WARNING] fishspeech.fish_speech.app:patched_inference:165 - set seed: 42
2025-11-07 02:23:44.853 [INFO] fishspeech.models.text2semantic.inference:generate_long:457 - Encoded text: 안녕하세요. 테스트입니다.
2025-11-07 02:23:47.862 [INFO] fishspeech.models.text2semantic.inference:generate_long:491 - Generated 37 tokens in 3.01 seconds, 12.30 tokens/sec
2025-11-07 02:23:47.862 [INFO] fishspeech.models.text2semantic.inference:generate_long:494 - Bandwidth achieved: 10.58 GB/s
2025-11-07 02:23:47.862 [INFO] fishspeech.models.text2semantic.inference:generate_long:497 - GPU Memory used: 5.02 GB
    VO features: torch.Size([10, 36])
2025-11-07 02:23:47.862 [INFO] fishspeech.models.text2semantic.inference:generate_long:497 - [FishSpeech] Dummy audio length: 73728 samples at 44100 Hz
[2025-11-07 02:23:49.049] [INFO] real.accelerator.py:203:get_accelerator() Setting ds_accelerator to cuda (auto detect)
INFO: Started server process [580]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

■ 3. 로컬 제어 패널 실행 (Starting the Local Control Panel)

Step 1 — 터미널에서 프로젝트 폴더 열기 (Open terminal in project folder)

```
cd /path/to/local_app
```

Step 2 — 의존성 설치 (Install dependencies)

```
pip install fastapi uvicorn httpx
```

Step 3 — 웹 앱 실행 (Start the web app)

```
uvicorn local_app:app --host 0.0.0.0 --port 5051
```

브라우저에서 다음 주소로 접속합니다:

☞ <http://localhost:5051>

K[R] 4. 로컬 웹 인터페이스 사용법 (업데이트)

개요

로컬 웹 인터페이스는 **단일 페이지** 구조로 구성되며, 다음 3개의 영역을 포함합니다:

영역	이름	기능
①	전화 패널	전화 발신 및 개인화된 인사말 생성
②	채팅 인터페이스	차진사 챗봇 테스트 (Chajipsa / GPT 엔진 선택 가능)
③	TTS 유ти리티 (선택)	로컬 TTS 서버로 직접 음성 생성

1 — 전화 패널

페이지 **좌측 상단** 영역입니다.

기능

- 아웃바운드 전화 걸기
- 고객 이름 및 전화번호 입력
- 개인화된 인사말 및 “네, {이름}님” 음성 생성

사용 방법

1. 전화번호 입력
2. (선택) 고객 이름 입력
3. **인사말 생성** 클릭
4. **통화 시작** 클릭

자리표시 이미지

The screenshot shows the 'Call with GPT' section of the AI Phone Call System. It includes fields for '전화번호' (Phone Number) containing '+821012345678', '표시 이름 (인트로용)' (Display Name (Introductory)) containing '차집사', and 'Call Engine' set to 'GPT Voice'. Below these are two blue buttons: 'Call with GPT' with a phone icon and 'Generate Intro' with a microphone and checkmark icon.

2 — TTS 유ти리티

페이지 하단에 위치합니다.

기능

- 텍스트 기반 음성 생성
- (선택) 참고 음성 업로드
- 오디오 재생 및 다운로드

사용 방법

1. 텍스트를 입력합니다
2. (선택) 참고 음성 업로드
3. 음성 생성 클릭
4. 생성된 음성을 재생하거나 다운로드

자리표시 이미지

The screenshot shows a user interface for a Text-to-Speech application. At the top left is a microphone icon followed by the text "Text-to-Speech". Below this is a section labeled "엔진 선택" (Engine Selection) with a dropdown menu set to "FishSpeech". The next section is "텍스트" (Text) with a placeholder "여기에 텍스트를 입력하세요..." (Input text here...). Below that is "톤 선택 (Tone Selection)" with buttons for "[None]", "(happy)", "(sad)", "(angry)", "(excited)", "(friendly)", and "(fearful)". A "Temperature" slider is set to 0.70. Under "참조 음성 (선택)" (Reference voice selection), there is a "Choose File" button with "No file chosen" and a blue "Generate Audio" button with a speaker icon.

💬 3 — 채팅 인터페이스

페이지 우측에 있습니다.

기능

- 차집사 챗봇과 대화
- Chajipsa / GPT 엔진 선택
- 응답 메시지 옆에 엔진 아이콘 표시
- 시스템 로그 출력

사용 방법

1. 메시지 입력
2. 엔진 선택
3. 보내기 또는 Enter

자리표시 이미지

안녕하세요, 차집사 다이렉트입니다. 보험료 비교 안내 간단히 드려도 될까요? C

안녕하세요! 어떻게 도와드릴까요? G

Type your message here... Send

Clear Chat

△ 5. 문제 해결 (Troubleshooting)

문제	원인	해결 방법
"Server is not running" 메시지	NIPA 서버가 중지되었거나 Ngrok 연결이 끊김	Ngrok 재연결 후 <code>uvicorn app:app</code> 재실행
음성이 비어 있음	참고 파일 누락 또는 잘못된 경로	<code>/tts/synthesize2</code> API를 직접 테스트
전화 발신이 작동하지 않음	AWS Connect 권한 문제	Lambda + Connect 통합 권한 확인
"Generating..." 상태가 멈춤	서버 지연 또는 메모리 문제	NIPA 로그 확인

🧠 6. 다음 단계 (Next Steps)

- 웹 API 및 제어 패널 완성
- 멀티태스크 챗봇 모델 학습 (의도 분류 + 응답 생성)
- 전화 통화 흐름에 챗봇 통합 (선택 사항 - 3번째 탭)