

AI Phone Call & TTS Control System

System Overview

This system consists of **two connected parts**:

Component	Location	Purpose
1. NIPA Cloud Server	Hosted on NIPA Cloud	Runs the core AI services — Text-to-Speech (FishSpeech), Chatbot, and Phone Call APIs. Handles AWS Connect integration, DynamoDB, and S3 audio uploads.
2. Local Control Panel (Web UI)	Run on local PC	A simple bilingual (KR/EN) interface used to initiate phone calls, generate greetings, and test TTS with uploaded reference audio.

The local interface connects to the NIPA server via public domain.

1. Server Access (NIPA Cloud)

Item	Details
Link	https://nipa.nhncloud.com/
ID	nipa-gpu2025-668@ktcloud.com
Password	_insunetfc6276
Server URL	https://honest-trivially-buffalo.ngrok-free.app

2. Starting the NIPA Server

Step 1 — Starting the server

The screenshot shows the N H N CLOUD interface for managing sessions. On the left sidebar, under the 'Sessions' section, there is a red box around the 'Import & Run' button. The main area displays 'Resource Statistics' for the 'H100E' resource group, showing CPU, RAM, and GPU usage. Below this is a table of sessions:

Session Info	Status	Control	Configuration	Usage	Reservation	Idle Checks	Session Type
IgmDjhqQ-session PyTorch 2.0.1 NVIDIA GPU Cloud	RUNNING		VALL-E H100EIG 24GB 192.00GB (SHM 2.00GB) 1.00GPU	CPU 0.6% RAM 3.8 / 192 GB GPU Util 0.0 % GPU mem 5.21 / 79.65 GB I/O R: 0.0 MB / W: 2805.7 MB	11/7/2025, 9:07:58 AM Elapsed Time: 02:12:55	Utilization Checker 03:47:25 Grace Period	INTERACTIVE

A red box highlights the 'START' button in the top right corner of the session list.

Step 2 — Mount VALL-E directory

Select VALL-E directory for mounting

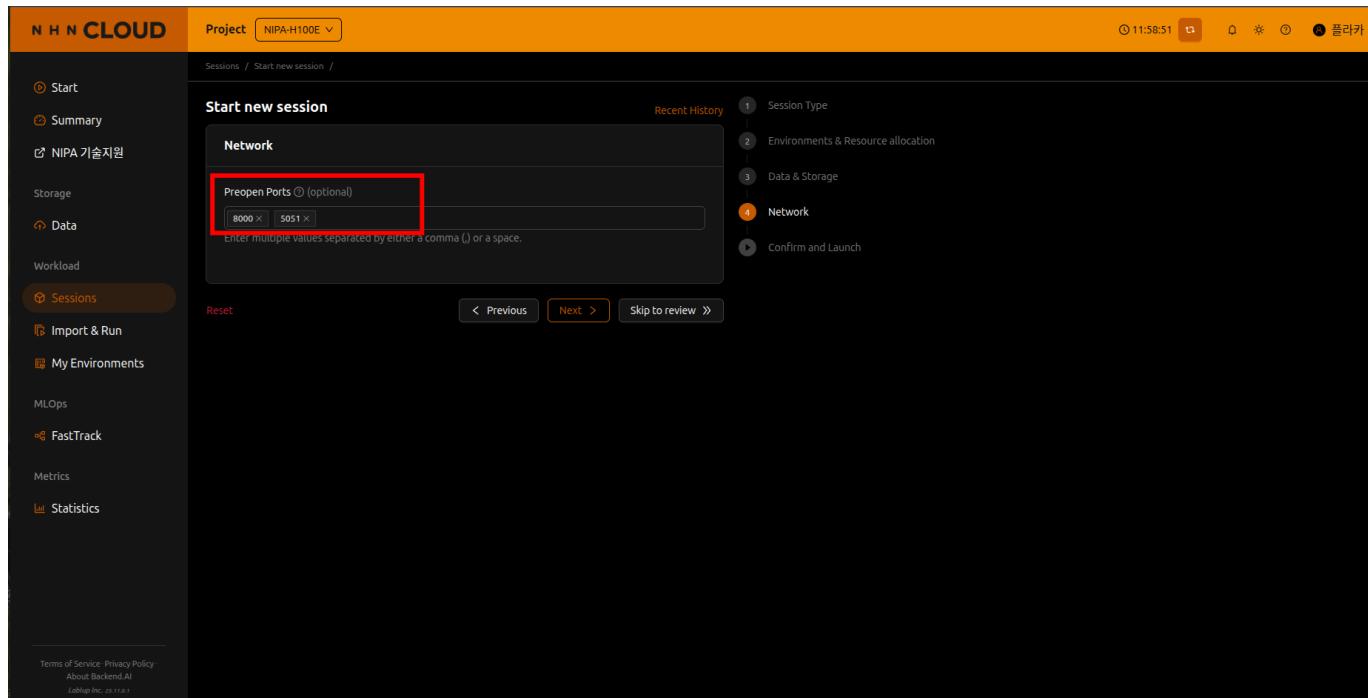
The screenshot shows the 'Start new session' wizard, step 3: Data & Storage. A red box highlights the checkbox next to the 'VALL-E' entry in the list, indicating it is selected for mounting. The list shows:

- test (general, sp1:nas_volo2, User)
- VALL-E (general, sp1:nas_volo2, User)

Below the list are buttons for 'Reset', '< Previous' and 'Next >', and 'Skip to review >'.

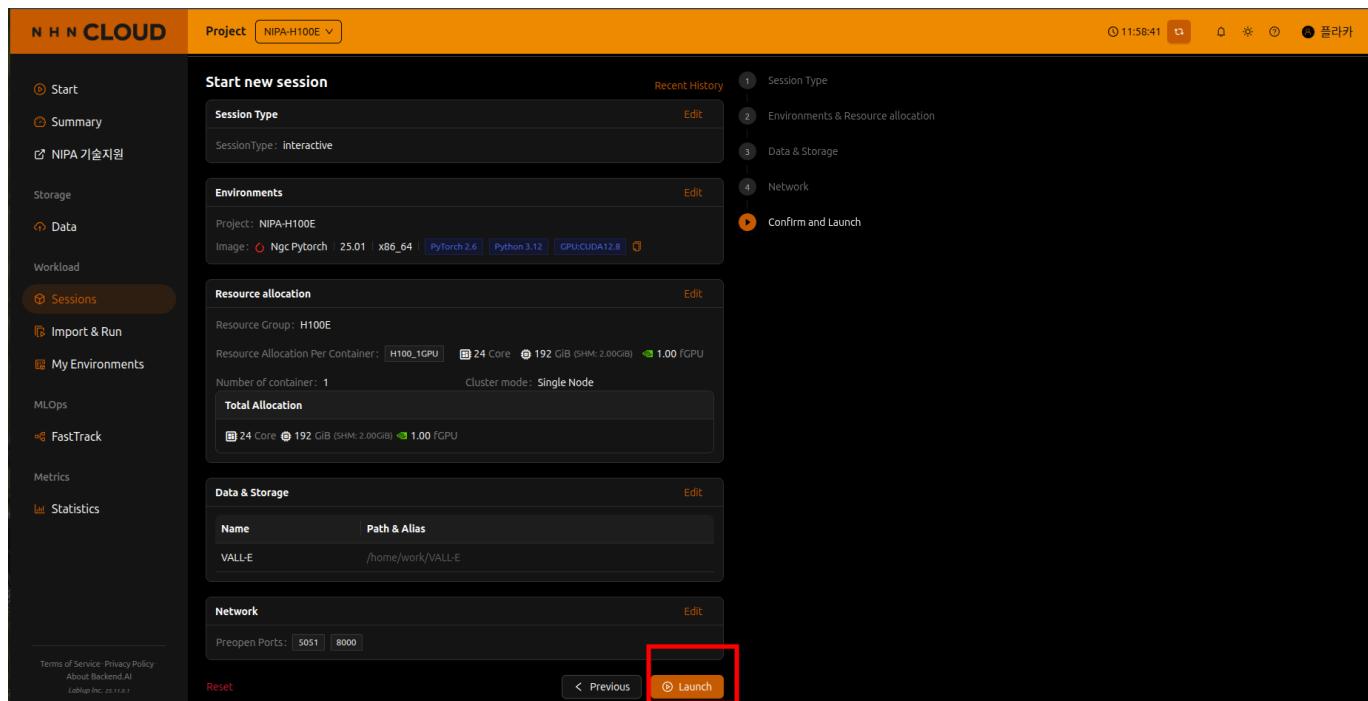
Step 3 — Select pre-open ports

Input ports 8000 and 5051



Step 4 — Launch the server

Click on the launch button



Step 5 — Selecting app dialog for server configuration

Click on app dialog button

The screenshot shows the NHN CLOUD interface. On the left sidebar, under the 'Sessions' section, there is a 'See App Dialog' button. This button is highlighted with a red box in the center of the screenshot. The main area displays 'Resource Statistics' for the 'H100E' resource group, showing CPU, RAM, and GPU usage. Below this is a table of sessions, with one row for 'IgmDjhQ-session' marked as 'RUNNING'. The 'See App Dialog' button is located to the right of the session status for this row.

Step 6 — Select and open JupyterLab

Select JupyterLab

The screenshot shows the NHN CLOUD interface with the 'App' dialog box open. The 'Development' section of the dialog is visible, containing icons for Visual Studio Code, Visual Studio Code (Desktop), JupyterLab, and Jupyter Notebook. The 'JupyterLab' icon is highlighted with a red box. The background shows the main session list, with one session named 'IgmDjhQ-session' running on PyTorch 25.01 and CUDA 12.8.

Step 7 — Start the NIPA cloud server

Run ./VALL-E/run_server.sh

```
workmain[1gn0]#gp0-session]: $ ./VALL-E/run_server.sh
```

The terminal output shows the server starting up, including environment loading, AWS access key export, and the start of the uwsgi server.

Step 8 — NIPA cloud server launch complete

This is the preview if the server launch is completed

```
workmain[1gn0]#gp0-session]: $ ./VALL-E/run_server.sh
```

The terminal output shows the server starting up, including environment loading, AWS access key export, and the start of the uwsgi server.

3. Starting the Local Control Panel (Client PC)

Step 1 — Open terminal in project folder

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

Step 2 — Install dependencies

```
pip install fastapi uvicorn httpx
```

Step 3 — Start the web app

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

Then open your browser and go to:

☞ <http://localhost:5051>

4. Using the Local Web Interface (Updated)

Overview

The local web interface is now a **single unified page** with three main functional areas:

Area	Name	Function
①	Phone Call Panel	Start calls and generate personalized greetings
②	Chat Interface	Test and interact with the 차집사 chatbot (Chajipsa / GPT engines)
③	TTS Utility (Optional)	Generate voice audio via the local TTS server

 Placeholder images shown below. Replace them with real screenshots later.

1 — Phone Call Panel

The **top-left** section of the interface handles phone call operations.

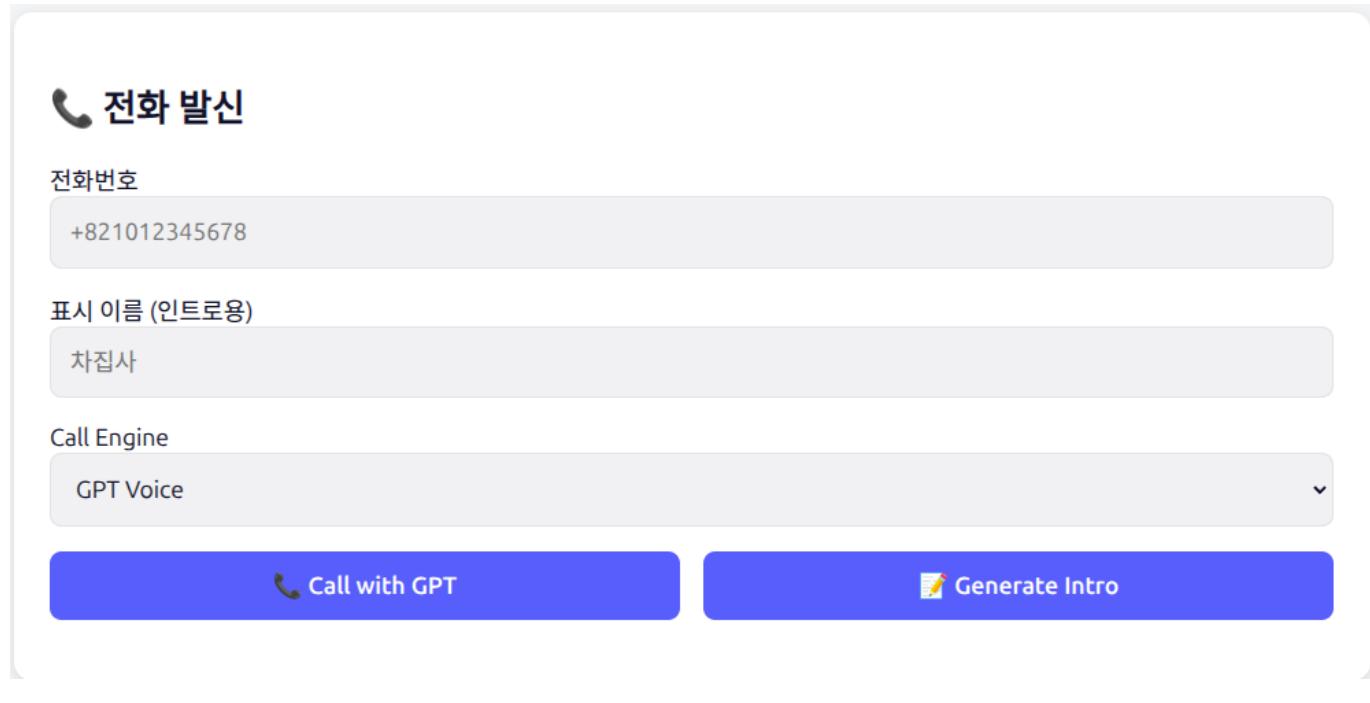
Functions

- Start outbound phone calls
- Enter customer name & phone number
- Generate personalized greetings:
 - Intro greeting (e.g., “안녕하세요, 홍길동 고객님...”)
 - Personalized affirmative clip (“네, 홍길동님”)

How to Use

1. Enter the phone number
2. (Optional) Enter the customer's name
3. Click **Generate Greeting**
4. Click **Start Call** to begin the outbound call

Screenshot Placeholder



2 — TTS Utility

Located at the **bottom** of the page.

Functions

- Generate voice audio
- (Optional) Upload reference audio (`.wav`, `.m4a`)
- Download or play generated audio

How to Use

1. Enter text into the TTS input
2. (Optional) upload a reference voice file
3. Click **Generate**
4. The result will appear as an audio player with a download option

Screenshot Placeholder

The screenshot shows the Text-to-Speech configuration page. At the top left is a microphone icon followed by the title "Text-to-Speech". Below it is a dropdown menu labeled "엔진 선택" (Engine Selection) with "FishSpeech" selected. A text input field labeled "텍스트" (Text) contains placeholder text "여기에 텍스트를 입력하세요...". Under "톤 선택 (Tone Selection)", there are seven buttons: [None], (happy), (sad), (angry), (excited), (friendly), and (fearful). A "Temperature" slider is set to 0.70. A "참조 음성 (선택)" (Reference Voice Selection) section includes a "Choose File" button with "No file chosen" and a blue "Generate Audio" button with a speaker icon.

💬 3 — Chat Interface

Located on the **right side** of the page.

Functions

- Interactive chat with the 차집사 assistant
- Switch between:
 - Chajipsa engine
 - GPT engine
- Engine icons appear next to each assistant message
- System log output shown underneath messages

How to Use

1. Type your message in the chat box
2. Choose the engine (Chajipsa or GPT)
3. Press **Enter** or click **Send**
4. Chat messages will appear continuously like a messenger UI

Screenshot Placeholder

The screenshot shows a chat application window. At the top left is a 'Chat' icon with three dots. Below it is a dropdown menu labeled '엔진:' with 'GPT' selected. On the right side of the header is the date '2025-11-17'. The main area consists of two message bubbles. The first bubble from the user says '안녕하세요, 차집사 다이렉트입니다. 보험료 비교 안내 간단히 드려도 될까요?' (Hello, this is DirectAgent. Can I briefly provide you with information about insurance premium comparison?). The second bubble from the AI says '안녕하세요' (Hello). Below the messages is a text input field with placeholder text 'Type your message here...' and a blue 'Send' button. At the bottom of the window is a dark grey bar with a 'Clear Chat' button.

⚠ 5. Troubleshooting

Issue	Cause	Solution
"Server is not running" message	NIPA server is offline or Ngrok disconnected	Reconnect Ngrok and restart <code>uvicorn app:app</code>
No sound / empty audio	Missing reference file or wrong prefix	Verify <code>/tts/synthesize2</code> works
Call not triggered	AWS Connect permission issue	Check Lambda + Connect integration
Browser stuck at "Generating..."	Timeout or memory issue	Check NIPA logs

🧠 6. Next Steps

- Web API and Control Panel completed
- Train new multitask chatbot model (intent + response)
- Integrate chatbot into phone call flow (optional Tab 3)