

## 1. 사전 준비 (Prerequisites)

봇을 실행하기 전에 두 가지 중요한 키(Key)가 필요합니다.

### 가. 디스코드 봇 토큰 (Discord Bot Token)

1. **Discord Developer Portal** 로 이동하여 로그인합니다.
2. **New Application** 버튼을 클릭하고 봇의 이름을 정합니다 (예: TTS 버섯 봇).
3. 생성된 애플리케이션의 **Bot** 탭으로 이동합니다.
4. **Add Bot** 버튼을 클릭하여 봇을 생성합니다.
5. **Reset Token** 버튼을 클릭하여 봇의 토큰을 복사합니다. 이 토큰은 매우 중요하므로 안전하게 보관해야 합니다.
6. **Privileged Gateway Intents** 섹션에서 모든 INTENT(3개) 를 활성화(enable)해야 합니다. 이는 봇이 멤버 목록을 보고 메시지를 읽으며 음성 채널 상태를 감지하는 데 필요합니다.

### 나. Google Cloud TTS API 키 (Google Cloud Text-to-Speech API Key)

1. **Google Cloud Console** 로 이동하여 로그인합니다.
2. 새 프로젝트를 만들거나 기존 프로젝트를 선택합니다.
3. "Cloud Text-to-Speech API"를 활성화합니다.
4. 다음 참고: <https://www.youtube.com/watch?v=mVL4N0VBPIY>
5. 서비스 계정을 생성하고 키를 생성합니다. JSON 형식을 선택하여 다운로드합니다.
6. 다운로드받은 JSON 파일을 프로젝트 디렉토리에 배치합니다.

---

## 2. 로컬 설정 및 실행

### 설정 방법

1. 원본 소스코드의 압축을 풉니다.
2. 의존성 설치

```
npm install
```

### 3. 환경 변수 구성

- 프로젝트 루트 디렉토리에 **.env.example** 파일을 복사하여 **.env** 파일을 생성합니다:

```
cp .env.example .env
```

- **.env** 파일을 열고 필요한 값을 입력합니다:

```
DISCORD_TOKEN=여기에_디스코드_봇_토큰을_입력하세요  
GOOGLE_APPLICATION_CREDENTIALS=./google-credentials.json
```

```
TTS_TEXT_CHANNEL_ID=텍스트_채널_ID(선택사항)  
TTS_VOICE_CHANNEL_ID=음성_채널_ID(선택사항)
```

- **DISCORD\_TOKEN**: Discord 개발자 포털에서 받은 봇의 토큰
- **GOOGLE\_APPLICATION\_CREDENTIALS**: Google Cloud 서비스 계정 JSON 키 파일의 경로. 예: `./google-credentials.json`. JSON 파일을 프로젝트 루트 디렉토리에 배치하세요.
- **TTS\_TEXT\_CHANNEL\_ID**, **TTS\_VOICE\_CHANNEL\_ID**: 채널 ID는 봇 명령어로도 설정 가능하므로 선택 사항입니다.

#### 4. 봇 실행

```
npm start
```

---

### 3. 서버 배포 가이드 (Ubuntu VPS 기준)

이 가이드는 봇을 24시간 안정적으로 운영하기 위해 리눅스(Ubuntu) 가상 서버(VPS)에 배포하는 과정을 안내합니다.

#### 3.1. 준비물

- 리눅스(Ubuntu) VPS: [iwinv](#), AWS, GCP 등 클라우드 서비스에서 제공하는 가상 서버.
- 소스코드 압축 파일: 이 프로젝트 폴더 전체를 **bot.zip**과 같이 압축한 파일.
- SSH 접속 프로그램: Windows의 **PuTTY** 또는 macOS의 기본 터미널.

#### 3.2. 서버 접속 및 기본 프로그램 설치

1. **SSH로 서버 접속** 터미널을 열고 아래 명령어로 VPS에 접속합니다.

```
ssh [사용자이름]@[서버_IP_주소]
```

2. **시스템 업데이트** 패키지 목록을 최신 상태로 업데이트합니다.

```
sudo apt update && sudo apt upgrade -y
```

3. **Node.js, npm, Unzip 설치** 봇 실행에 필요한 Node.js 환경과 압축 해제 도구를 설치합니다.

```
sudo apt install -y nodejs npm unzip
```

4. **PM2 설치** Node.js 애플리케이션(봇)을 24시간 실행하고 관리해주는 프로세스 매니저입니다.

```
sudo npm install -g pm2
```

### 3.3. 소스코드 업로드 및 설정

1. 소스코드 업로드 (SCP) 로컬 컴퓨터의 터미널에서 아래 명령어를 실행하여 소스코드 압축 파일을 서버로 전송합니다.

```
# scp [로컬 파일 경로] [서버 사용자이름]@[서버 IP주소]:[서버에 저장될 경로]
# 예시: 바탕화면의 bot.zip을 서버의 홈 디렉토리(/home/ubuntu)에 업로드
scp ~/Desktop/bot.zip ubuntu@123.45.67.89:/home/ubuntu/
```

2. 서버에서 압축 풀기 다시 서버에 접속한 SSH 터미널로 돌아와서 압축을 풉니다.

```
# 파일명이 다르다면 실제 파일명으로 변경하세요.
unzip bot.zip
# 생성된 프로젝트 폴더로 이동합니다.
cd 250827-tts-bot # 폴더명은 압축 파일에 따라 다를 수 있습니다.
```

### 3. 의존성 설치 및 환경 설정

- 봇 실행에 필요한 라이브러리를 설치합니다.

```
npm install
```

- .env 파일을 서버 환경에 맞게 생성하고 내용을 채웁니다. 가이드를 참고하여 nano 편집기 등으로 작성하세요.

```
nano .env
```

- Google Cloud 서비스 계정 JSON 파일도 서버에 업로드하고 적절한 경로로 설정해야 합니다.

### 3.4. PM2로 봇 실행하기

#### 1. PM2로 봇 시작

```
pm2 start ecosystem.config.js
```

또는 직접 실행:

```
pm2 start npm --name "mushroom-tts-bot" -- start
```

#### 2. 실행 상태 확인

```
pm2 list
```

`mushroom-tts-bot` 프로세스가 **online** 상태로 표시되면 성공입니다.

3. 서버 재부팅 시 자동 실행 설정 아래 명령어를 실행해두면, 서버가 재부팅되어도 PM2가 봇을 자동으로 다시 실행시켜 줍니다.

```
pm2 startup
# 화면에 나오는 `sudo env ...`로 시작하는 명령어를 복사하여 한 번 더 실행해주세요.
pm2 save
```

### 3.5. 봇 관리 명령어 (PM2)

- 로그 확인: `pm2 logs mushroom-tts-bot`
- 봇 재시작: `pm2 restart mushroom-tts-bot`
- 봇 중지: `pm2 stop mushroom-tts-bot`
- 프로세스 목록에서 제거: `pm2 delete mushroom-tts-bot`

---

## 4. 환경 변수 설정

프로젝트의 최상위 폴더에 `.env`라는 이름의 파일을 생성하고 아래 내용을 붙여넣으세요.

```
# 디스코드 봇 토큰
DISCORD_TOKEN=여기에_디스코드_봇_토큰을_붙여넣으세요

# Google Cloud TTS 설정
GOOGLE_APPLICATION_CREDENTIALS=./google-credentials.json

# 선택사항: 기본 채널 설정 (봇 명령어로도 설정 가능)
TTS_TEXT_CHANNEL_ID=텍스트_채널_ID
TTS_VOICE_CHANNEL_ID=음성_채널_ID
```

#### 주의사항:

- `DISCORD_TOKEN`은 Discord Developer Portal의 Bot 탭에서 생성한 토큰입니다.
- `GOOGLE_APPLICATION_CREDENTIALS`는 Google Cloud 서비스 계정 JSON 키 파일의 상대경로 또는 절대경로입니다.
- 채널 ID들은 디스코드에서 개발자 모드를 활성화한 후 채널을 우클릭하여 복사할 수 있습니다.

---

## 5. 봇 기능

### 주요 기능

- 🔊 한국어 텍스트를 음성으로 변환 (Google Cloud TTS 사용)

- 🎙 개인별 목소리 설정 가능 (Neural2, Wavenet 음성 지원)
- 🚫 도배/스팸 자동 감지 및 타임아웃 처리
- 🗋 음성 채널 입장/퇴장 인사 기능
- 🗑 대량 메시지 삭제 기능
- ⚙ 관리자용 설정 명령어

## 사용 가능한 명령어

- **/설정**: 봇의 각종 설정을 관리합니다 (관리자 전용)
- **/삭제**: 메시지를 일괄 삭제합니다 (메시지 관리 권한 필요)
- **/상태**: 봇의 현재 설정 상태를 확인합니다

## 지원하는 TTS 목소리

- Neural2-A, B, C (자연스러운 음성)
- Wavenet-A, B, C, D (고품질 음성)

---

## 6. 문제 해결

### 봇이 음성을 재생하지 않는 경우

1. Google Cloud TTS API 키가 올바르게 설정되었는지 확인
2. 봇이 음성 채널에 연결되어 있는지 확인
3. TTS 텍스트 채널이 설정되어 있는지 확인

### PM2 로그 확인

```
pm2 logs mushroom-tts-bot
```

### 서비스 재시작

```
pm2 restart mushroom-tts-bot
```