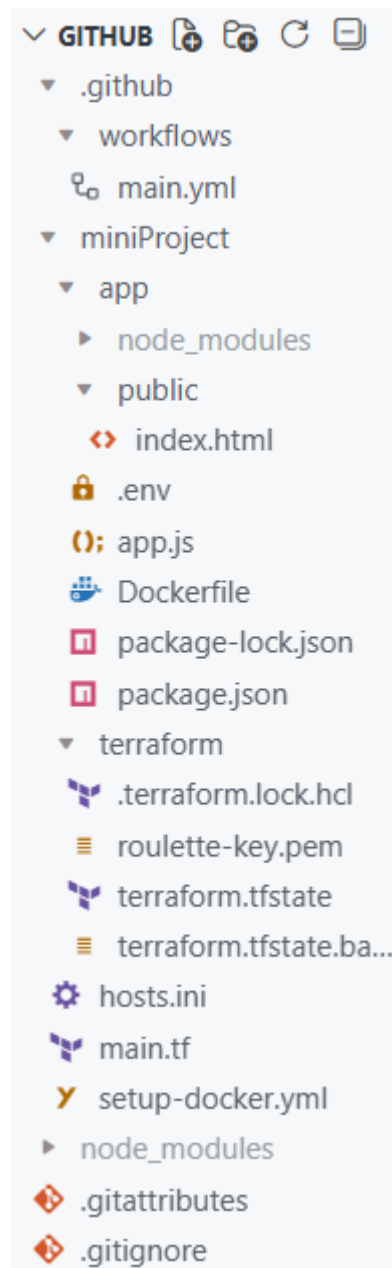


정리본(코드)

📁 과목	miniProject(정리)
📅 날짜	@2026년 1월 30일

디렉토리



파일

- **main.tf (인프라 정의서)**
 - **역할:** AWS EC2 인스턴스, VPC, 보안 그룹(Security Group)을 자동으로 생성합니다.
 - **필수 설정:**
 - `provider "aws"` : 리전 설정.
 - `resource "aws_security_group"` : 22번(SSH), 80번(HTTP) 포트 개방 설정.
 - `key_name` : 사용할 키 페어 이름 명시.
- **roulette-key.pem (개인 키)**
 - **역할:** 생성된 EC2에 SSH로 접속할 때 사용하는 복호화 열쇠입니다.
 - **필수 설정:** 파일 권한을 `400` 또는 `600` 으로 제한해야 안전하게 작동합니다.
- **package.json (의존성 관리자)**
 - **역할:** 프로젝트 정보와 필요한 라이브러리(`express` 등) 목록을 정의합니다.
 - **필수 설정:** `dependencies` 항목에 `"express": "^5.2.1"` 등이 포함되어야 합니다.
 - `npm install express`
 - `npm install mysql2`
 - (주의 !) 반드시 `miniProject/app` 경로에서 아래 명령어를 입력
→ 그래야 도커가 읽을 수 있음
- **app.js (실행 메인 코드)**
 - **역할:** Node.js 서버의 로직을 담당하며 브라우저 요청을 처리합니다.
 - 파일 생성 후 가장 먼저 `$ npm init` 하기
- **Dockerfile (이미지 빌드 설계도)**
 - **역할:** 소스 코드를 실행 가능한 이미지로 압축하는 순서를 정의합니다.
 - **필수 설정:** `WORKDIR`, `COPY package*.json ./`, `RUN npm install`, `CMD ["node", "app.js"]` 순서가 정확해야 합니다.
- **.github/workflows/main.yml (CI/CD 지시서)**

- **역할:** 코드가 push되면 이미지 빌드, 도커 허브 업로드, 앤서블 실행을 순차적으로 수행합니다.
- **필수 설정:**
 - `runs-on: ubuntu-latest` : 깃허브의 리눅스 가상 컴퓨터 사용.
 - `secrets` : 깃허브 설정에 등록한 `DOCKERHUB_TOKEN` , `SERVER_IP` , `SSH_PRIVATE_KEY` 호출.
 - `working-directory` : `./miniProject/app` 등 정확한 작업 경로 지정.

깃허브 액션의 명령을 받아 EC2 서버 내부를 직접 세팅합니다.

- **`hosts.ini` (인벤토리/주소록)**
 - **역할:** 앤서블이 접속할 대상 서버의 IP와 접속 계정(ubuntu)을 지정합니다.
 - **필수 설정:** `[web]` , `ansible_host=${{ secrets.SERVER_IP }}` , `ansible_user=ubuntu` .
- **`setup-docker.yml` (플레이북/현장 지시서)**
 - **역할:** 서버 내 도커 설치, 기존 컨테이너 중지/삭제, 새 이미지 실행을 담당합니다.
 - **필수 설정:**
 - `docker rm -f roulette-app || true` : 이름 충돌 방지를 위한 강제 삭제 구문.
 - `ports: "80:3000"` : 외부 80번 포트를 내부 3000번으로 연결.

추가)

.env 파일

- 현재 로컬의 mysql에 저장된 내 정보

테라폼 폴더 안의 설정 파일들

- 코드 작성 후 정리본(인프라) 설명을 따라하면 자동으로 생성됨