### **GRK Workspace 배포 로드맵**

로컬 컴퓨터에 있는 프론트엔드와 백엔드 프로젝트를 네이버 클라우드에 옮겨서, 누구나 접속할 수 있는 실제 웹 서비스로 만드는 전체 과정입니다.

### **Phase 1: 서버 컴퓨터 준비하기 (가상 서버 생성)**

가장 먼저, 백엔드 코드를 실행하고 웹사이트 파일을 올려둘 가상 컴퓨터를 네이버 클라우드에 생성해야 합니다.

1. **NCP 콘솔 접속 및 서버 생성:**
   * 네이버 클라우드 플랫폼 콘솔에서 **Services > Compute > Server**로 이동합니다.
   * [+ 서버 생성] 버튼을 클릭합니다.
   * **OS 선택:** Ubuntu 최신 버전(예: 20.04)을 선택하는 것을 추천합니다.
   * **서버 설정:** 개발 초기 단계이므로 가장 낮은 사양으로 시작해도 충분합니다.
   * **인증키 설정:** 서버에 접속할 때 사용할 비밀번호 같은 역할을 합니다. 새로운 인증키 생성을 선택하고, 인증키 이름(예: grk-key)을 입력한 후 **인증키를 꼭 다운로드하여 안전한 곳에 보관하세요.** (잃어버리면 서버에 접속할 수 없습니다.)
   * **ACG(방화벽) 설정:** 신규 ACG 생성을 선택하고, 아래 규칙들을 **반드시 추가**해야 합니다.
     + TCP / 22 / 접근소스: 내 IP (SSH 원격 접속용)
     + TCP / 80 / 접근소스: 0.0.0.0/0 (HTTP 웹 접속용)
     + TCP / 443 / 접근소스: 0.0.0.0/0 (HTTPS 보안 웹 접속용)
     + TCP / 3000 / 접근소스: 0.0.0.0/0 (백엔드 API 테스트용)
   * 최종 확인 후 서버 생성을 완료합니다.
2. **공인 IP 할당:**
   * 서버가 생성되면 외부에서 접속할 수 있도록 공인 IP를 할당해야 합니다.
   * **Server > Public IP** 메뉴로 이동하여 [+ 공인 IP 신청]을 클릭합니다.
   * 방금 만든 서버를 선택하여 공인 IP를 할당합니다. 이제 이 IP 주소로 서버에 접속할 수 있습니다.

### **Phase 2: 서버에 개발 환경 구축하기**

이제 생성된 서버에 원격으로 접속해서, Nest.js와 React 앱을 실행하는 데 필요한 프로그램들을 설치합니다.

1. **서버 접속 (SSH):**
   * Mac의 터미널이나 Windows의 PowerShell을 엽니다.
   * 아래 명령어를 입력하여 서버에 접속합니다.  
     ssh -i [1단계에서 다운로드한 인증키 경로] root@[할당받은 공인 IP]
2. **필수 프로그램 설치:**
   * 서버에 접속된 상태에서 아래 명령어들을 순서대로 입력하여 Node.js, Nginx(웹 서버) 등을 설치합니다.  
     # 시스템 업데이트  
     sudo apt-get update  
       
     # Node.js 및 npm 설치  
     curl -sL https://deb.nodesource.com/setup\_18.x | sudo -E bash -  
     sudo apt-get install -y nodejs  
       
     # PM2 설치 (Node.js 앱을 안정적으로 실행시켜주는 도구)  
     sudo npm install -g pm2  
       
     # Nginx 설치 (웹 서버)  
     sudo apt-get install -y nginx  
       
     # Git 설치  
     sudo apt-get install -y git

### **Phase 3: 백엔드(Nest.js) 배포하기**

1. **DB 방화벽 설정:**
   * 이전에 Cloud DB for PostgreSQL을 만들 때 생성했던 ACG(방화벽)로 이동합니다.
   * Inbound 규칙에 **TCP / 5432 / 접근 소스: [1단계에서 만든 서버의 비공인 IP]** 를 추가합니다. (서버와 DB는 안전한 내부망으로 통신해야 합니다.)
2. **소스코드 복제 및 설정:**
   * SSH로 서버에 접속된 상태에서, Git에 올려둔 백엔드 프로젝트를 복제합니다.  
     git clone [백엔드 프로젝트 Git 주소]  
     cd [프로젝트 폴더명]
   * .env 파일을 생성하고, **네이버 클라우드의 PostgreSQL DB 접속 정보**를 입력합니다.  
     # .env 파일 내용  
     DB\_HOST='[NCP DB의 프라이빗 도메인]'  
     DB\_PORT=5432  
     ...
   * 필요한 라이브러리를 설치합니다.  
     npm install
3. **애플리케이션 빌드 및 실행:**
   * 코드를 실행 가능한 파일로 변환(빌드)합니다.  
     npm run build
   * pm2를 사용해 백엔드 서버를 실행합니다. 이렇게 하면 터미널을 종료해도 서버가 계속 실행됩니다.  
     pm2 start dist/main.js --name "grk-backend"

### **Phase 4: 프론트엔드(React) 배포 및 Nginx 연동**

1. **소스코드 빌드 및 업로드:**
   * **로컬 컴퓨터**에서 React 프로젝트를 빌드합니다.  
     npm run build
   * 빌드가 완료되면 build 폴더가 생성됩니다. 이 폴더의 모든 내용을 scp와 같은 명령어를 사용해 **서버의 특정 위치**(예: /var/www/grk-frontend)로 업로드합니다.
2. **Nginx 설정 (가장 중요):**
   * SSH로 서버에 접속된 상태에서, Nginx의 설정 파일을 수정합니다.  
     sudo nano /etc/nginx/sites-available/default
   * 파일 내용을 아래와 같이 수정합니다. 이 설정은 사용자가 웹사이트에 접속하면 React 화면을 보여주고, 특정 API 경로로 요청하면 백엔드로 연결해주는 역할을 합니다.  
     server {  
      listen 80;  
      server\_name [할당받은 공인 IP]; # 나중에 도메인을 사면 도메인 주소로 변경  
       
      # React 앱 파일 위치  
      root /var/www/grk-frontend;   
      index index.html index.htm;  
       
      # API 요청은 백엔드로 전달 (Reverse Proxy)  
      location /api {  
      proxy\_pass http://localhost:3000; # Nest.js가 실행중인 포트  
      proxy\_http\_version 1.1;  
      proxy\_set\_header Upgrade $http\_upgrade;  
      proxy\_set\_header Connection 'upgrade';  
      proxy\_set\_header Host $host;  
      proxy\_cache\_bypass $http\_upgrade;  
      }  
       
      # 그 외 모든 요청은 React 앱이 처리하도록 설정  
      location / {  
      try\_files $uri $uri/ /index.html;  
      }  
     }
   * 설정 저장 후 Nginx를 재시작합니다.  
     sudo systemctl restart nginx

이제 웹 브라우저에서 **http://[할당받은 공인 IP]** 주소로 접속하면 GRK Workspace의 로그인 화면이 나타나고, 모든 기능이 정상적으로 동작할 것입니다.