

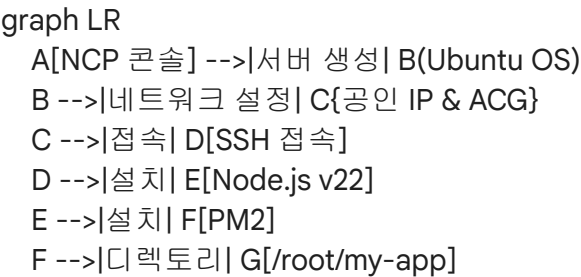
[문서] 서버 구축 및 CI/CD 배포 자동화 명세서

작성일: 2025.12.10
대상 시스템: Naver Cloud Platform (NCP) Ubuntu Server
목표: Node.js 애플리케이션의 무중단 배포 자동화

1. NCP 서버 구성 (Server Configuration)

[A] 구성 흐름도

서버 생성부터 서비스 구동까지의 흐름입니다.



[B] 작업 이력 및 설정 상세 (시계열 순)

순번	구분	상세 내용	명령어 / 비고
01	인스턴스 생성	NCP 콘솔에서 Ubuntu 20.04/22.04 기반 서버 생성	Micro/Standard 스펙 사용
02	네트워크	공인 IP 할당 및 포트 포워딩 설정	SSH(22), Web(3000) 포트 오픈
03	방화벽(ACG)	0.0.0.0/0 대상 Inbound 규칙 추가	[중요] 3000번 포트 허용 필수
04	Node.js 설치	[시행착오] v18 설치 시 EOL 경고 발생 -> v22 (LTS) 최신 버전으로 변경 설치	curl -fsSL ... setup_22.x
05	PM2 설치	무중단 프로세스 관리를 위한 PM2 전역	npm install -g pm2

		설치	
06	경로 생성	배포 파일이 위치할 디렉토리 생성	mkdir -p /root/my-app

2. CI/CD 파이프라인 구성 (GitHub Actions)

[A] 배포 시퀀스 (Architecture)

sequenceDiagram

participant Local as [Local PC]
 participant Git as [GitHub Repo]
 participant Action as [GitHub Actions]
 participant Server as [NCP Server]

Local->>Git: Push (Code + package-lock.json)

Git->>Action: Trigger 'deploy.yml'

rect rgb(240, 240, 240)

Note over Action: [BUILD]

Action->>Action: Node.js v22 Setup

Action->>Action: npm ci (의존성 설치)

end

rect rgb(230, 230, 230)

Note over Action: [DEPLOY]

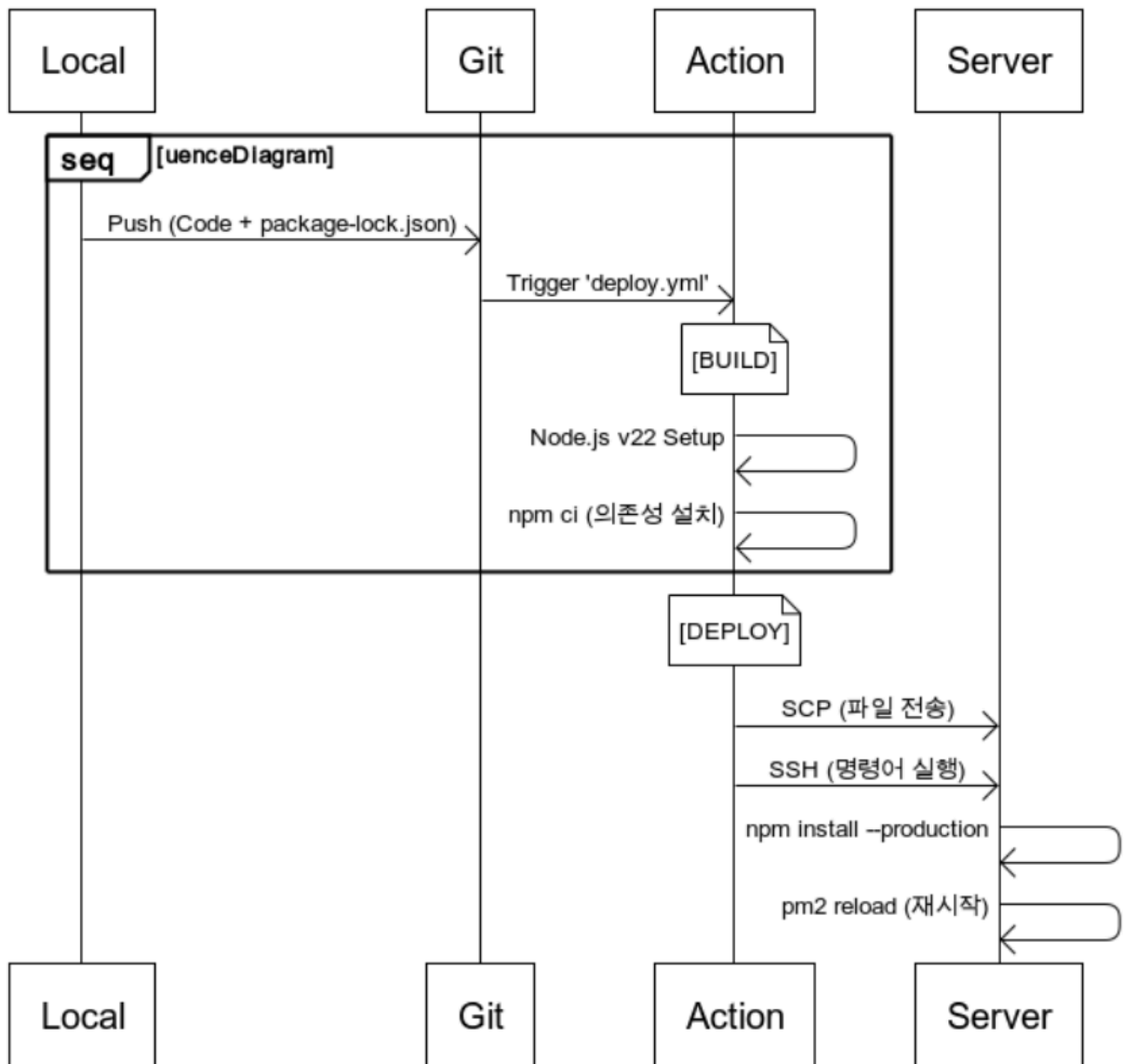
Action->>Server: SCP (파일 전송)

Action->>Server: SSH (명령어 실행)

Server->>Server: npm install --production

Server->>Server: pm2 reload (재시작)

end



[B] 단계별 구성 및 핵심 체크포인트

단계	작업 항목	[CRITICAL] 중요 체크포인트
Step 1	로컬 준비	<ul style="list-style-type: none"> - package.json 및 package-lock.json 필수 포함 - .gitignore에 node_modules 반드시 추가
Step 2	키 생성	<ul style="list-style-type: none"> - 호환성을 위해 PEM 형식 (-m PEM) 사용 - Passphrase(비밀번호) 없이 생성 (엔터 입력)

Step 3	서버 등록	- server_key.pub 내용을 ~/.ssh/authorized_keys에 등록 - 권한 설정: <code>chmod 600 authorized_keys</code>
Step 4	GitHub Secrets	- [Repository secrets] 메뉴에 등록 (Variables 아님) - REMOTE_SSH_KEY: -----BEGIN... 부터 끝까지 전체 복사
Step 5	Workflow	- .github/workflows/deploy.yml 작성 - Node 버전(22) 및 포트 등 환경변수 일치 확인

3. 트러블슈팅 (Troubleshooting History)

개발 과정에서 발생한 주요 에러와 해결 방법입니다.

[ISSUE 1] Dependencies lock file is not found

- 증상: GitHub Actions 빌드 단계에서 실패.
- 원인: Git 저장소에 package-lock.json 파일이 업로드되지 않음.
- 해결: 로컬에서 `npm install` 수행 후 생성된 락 파일을 Commit & Push 함.

[ISSUE 2] ssh: handshake failed / no key found

- 증상: SCP 전송 단계에서 인증 실패.
- 원인:
 1. 최신 OpenSSH 키 형식을 라이브러리가 인식하지 못함.
 2. GitHub Secrets에 Private Key 대신 Public Key를 잘못 등록함.
- 해결:
 1. `ssh-keygen -m PEM ...` 옵션으로 키 재생성.
 2. GitHub Secrets에 확장자가 없는 Private Key 내용 등록.

[ISSUE 3] ssh: passphrase protected

- 증상: SSH 접속 시도 중 비밀번호 에러 발생.
- 원인: SSH 키 생성 시 비밀번호(Passphrase)를 설정함.
- 해결: 키 재생성 시 비밀번호 입력 단계에서 아무것도 입력하지 않고 **Enter** 입력.