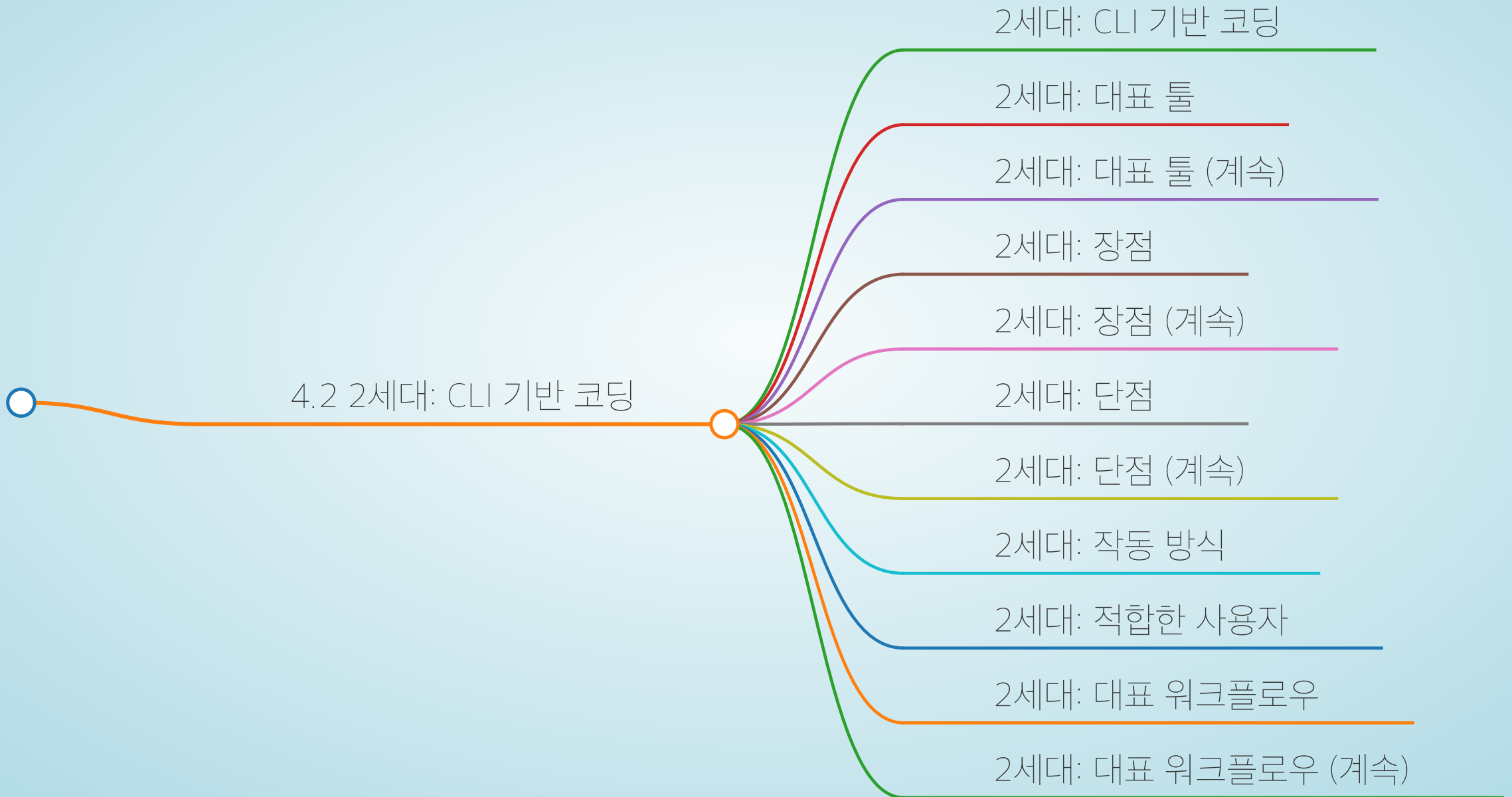


Examples for MarkdownGraph



2세대: CLI 기반 코딩

터미널(CLI)에서 AI와 대화하며 작업

- IDE 없이도 프로젝트를 전체를 자동화



Claude Code

- Anthropic 공식 CLI 도구
- 터미널에서 Claude와 직접 대화
- 프로젝트 전체 컨텍스트 이해
- 자동 파일 수정 및 명령 실행



2세대: 대표 툴 (계속)

Codex CLI

- 0전작 AI 기반 CLI 도구
- 터미널 기반 코드 생성
- Git 통합 및 자동화

서버/클라우드 기반

- 로컬 리소스 거의 사용 안 함
- 대트북 배터리 절약
- 원격 서버에서 작업 가동

무제한 병렬 처리

- 여러 터미널에서 동시 작업
- 멀티 프로젝트 동시 관리
- 백그라운드 실행 가동

2세대: 장점 (계속)

스크립트 자동화

- 셸 스크립트와 결합
- CI/CD 파이프라인 통합
- DevOps 워크플로우 자동화

빠른 실행 속도

- GUI 오버헤드 없음
- 텍스트 기반으로 빠른 응답
- 네트워크 대역폭 효율적

학습 곡선

- 터미널 명령실 U 지 프업
- G생 없이 작업해야 함
- 코드 변경 사항 플거화 제한적

시각적 피드백 부족

- d작f 확인이 텍스트 통반
- 그래픽 생 프리뷰 불가

2세대: 단점 (계속)

4.2 2세대: CLI 기반 코딩

초기 설정

- 환경 변수 설정 프요
- API 키 설정 프수

2세대: 작동 방식

```
# Claude Code 예시
```

```
$ claude
```

```
> 프로젝트 분석해서 README.md 업데이트해줘
```

```
[Claude가 프로젝트 파일들을 읽고 분석]
```

```
[README.md 자동 수정]
```

```
[변경사항 표시]
```

```
> 테스트 코드도 작성해줘
```

```
[테스트 파일 자동 생성]
```

```
[npm test 실행하여 검증]
```

2세대: 적합한 사용자

- 시니어 개발자: 터미널에 익숙한 터용자
- 백엔드/DevOps 개발자: 서버 관리 및 자동화 작업
- 멀티 프로젝트 관리: 여러 프로젝트를 동시에 진행
- 리눅스 사용자: C립 환경 선호
- 원격 개발: SSH로 원격 서버 작업

2세대: 대표 워크플로우

프로젝트 디렉토리에서

Claude Code 예시

```
# 프로젝트 디렉토리에서  
$ claude
```

```
> AWS Lambda 함수로 이미지 리사이징 서비스 만들어줘.  
   S3에서 이미지 받아서 처리하고 다시 S3에 저장.
```

[Claude가 작업 시작]

- ✓ lambda_function.py 생성
- ✓ requirements.txt 생성
- ✓ 환경변수 설정 확인
- ✓ SAM template 생성
- ✓ README.md 업데이트

2세대: 대표 워크플로우 (계속)

프로젝트 디렉토리에서

> 로컬에서 테스트해줘

[sam local invoke 실행]
[테스트 결과 확인]

> 문제없으면 배포해줘

[sam deploy 실행]
[배포 완료]