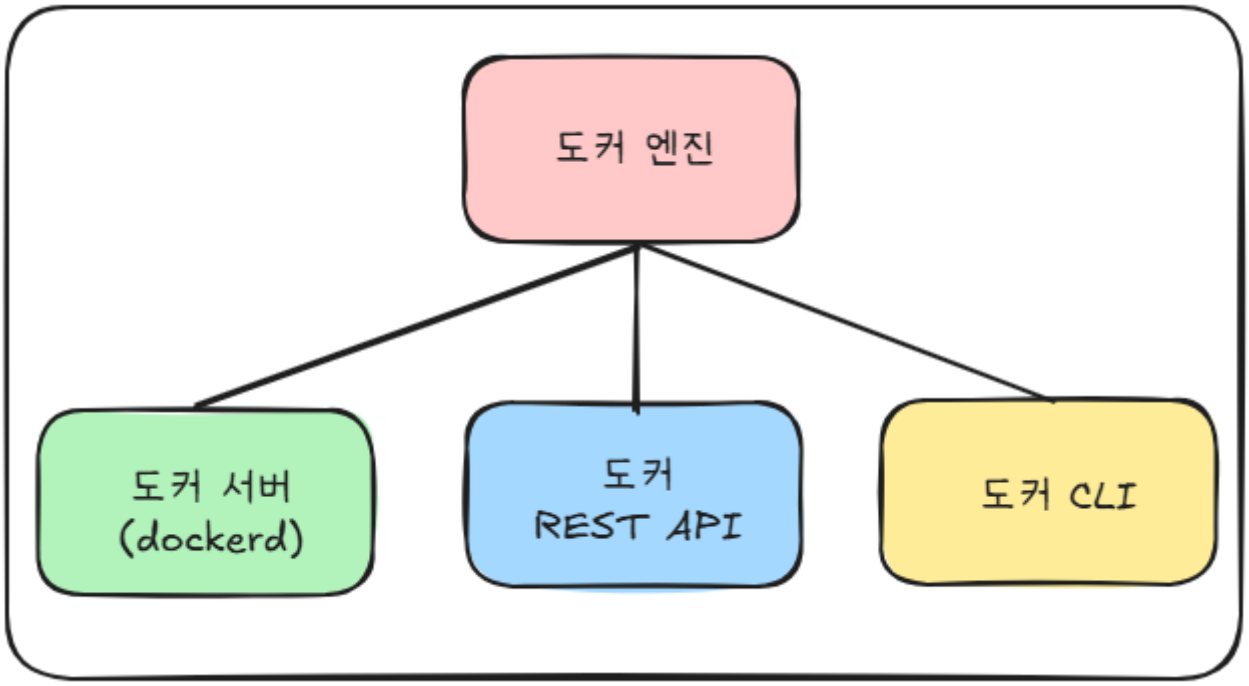
1. ????? ?? ??

- 도커란____?
- 컨테이너를_____사용하는_____이유___



????

????

- 컨테이너 가상화를 지원하는 대표 컨테이너 기술
- 애플리케이션 배포에 특화되어 있으며 컨테이너를 중심으로 개발 및 운영이 가능

??? ????? ??

- 도커 서버 : 컨테이너형 기술 구현을 위한 기본 애플리케이션으로 컨테이너 실행 및 관리 등 핵심을 담당
- 도커 **REST API**: 도커 서버를 조작하기 위한 인터페이스
- 도커 **CLI**: 도커 커맨드를 구현하는 애플리케이션으로 도커 **REST API**와 통신하며 도커 서버 를 조작함

????? ???? ??

????? ???? ??

- 불변하는 애플리케이션과 실행 환경에 대한 재현성 확보
- 애플리케이션 구성 관리의 용이성
- 환경과 상관없이 실행 가능한 높은 이식성
- 컨테이너 기반 개발의 필요성

???? ??????? ?? ??? ?? ???

> B 서버에도 같은 애플리케이션을 배포했는데 A와 다르게 동작해요

- 배포 서버의 상태가 달라 기대처럼 동작하지 않는 상황의 근본적인 원인은 가변적인 인프라 를 허용해서 발생함
  - OS, CPU, Memory, 라이브러리 , 언어 런타임 등 다양한 요소에 의존함
  - 각 서버에 배포하는 요소를 모두 하나로 통일하는 것이 해결 방법
- **IaC(Infrastructure as Code)**: 위 문제를 해결하기 위해 코드를 사용해서 인프라 환경을 구축하는 방법
  - 어떤 서버를 구성할 지 , 어떤 라이브러리나 애플리케이션을 사용할 지 코드로 정의하고 관리함
- IaC를 사용해도 항상 같은 결과를 보장하는 것은 아님
  - 예시 : `nvm install node`
    - 이 경우 항상 최신 버전이 릴리스될 때마다 업데이트되기에 같은 결과 보장이 어려움
- 언제나 실행해도 동일한 환경 유지가 가능하도록 모든 애플리케이션 런타임과 라이브러리는 특정 버전을 설치하도록 코드 를 작성

?????? ?? ??? ???

- 초기부터 도커는 도커 컴포즈 라는 간단한 컨테이너 오케스트레이션 시스템을 제공함
- `yaml` 형식의 설정 파일에 컨테이너 스펙을 정의하고 의존 관계 및 동작 순서를 제어해 컨테이너의 구성을 정의하고 실행
- 간편하다는 장점으로 인해 로컬 개발 환경 구성 에 널리 사용됨

??? ???? ?? ??? ?? ???

- [illegible]
- [illegible]
- [illegible]

???? ?? ??? ???

- [illegible], [illegible]
- **MSA** [illegible]
- **CI** [illegible] **API** [illegible] **(mock)** [illegible]