

앱 개발자를 위한 K8S (1)

2023.11.26 민경민



K8S는

1. 구글에서 만든 오픈소스 프로그램
2. 컨테이너화 된 워크로드와 서비스를 관리
3. 인프라를 선언적으로 세팅
4. 분산 시스템의 스케일링과 장애극복

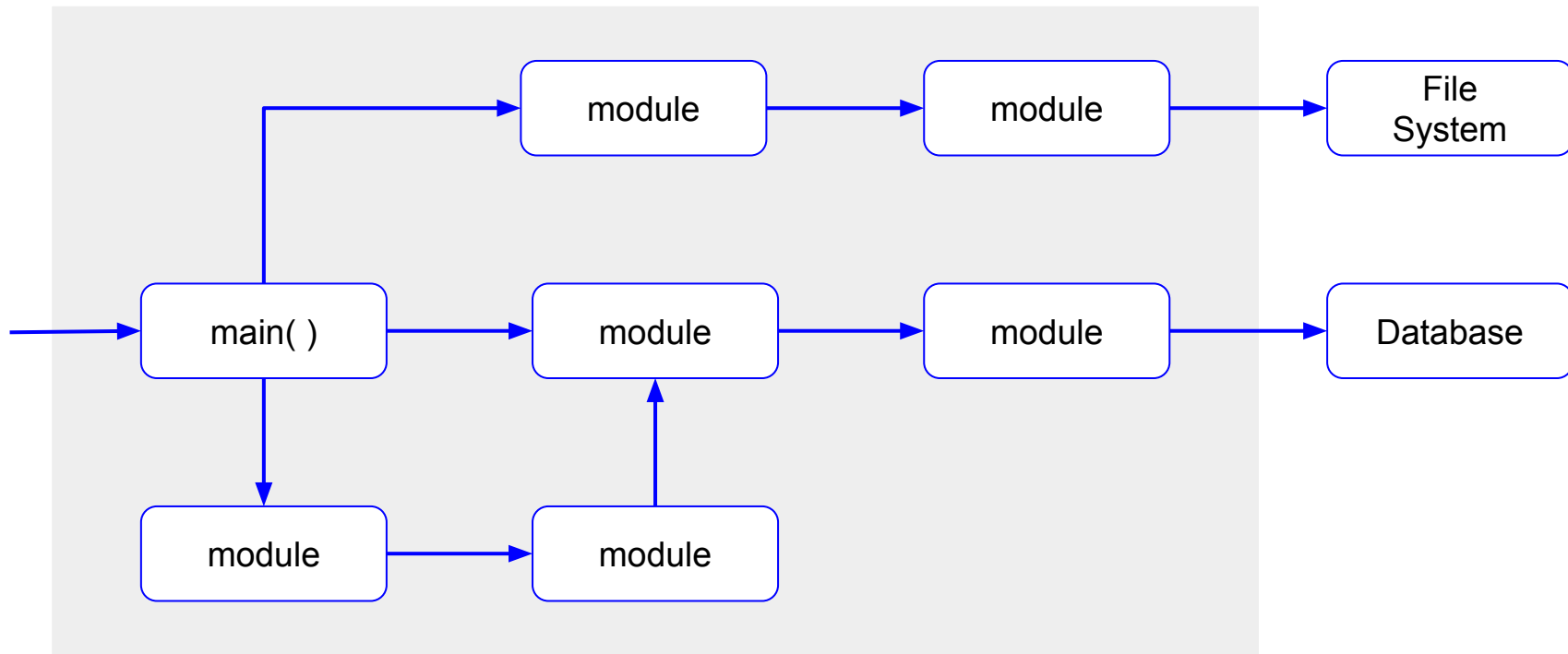


Kubernetes == K8S

구글에서 2014년에 오픈소스로 전환

Kubernetes K와 S 사이에 8자가 있어서 K8S

모놀리식 아키텍처



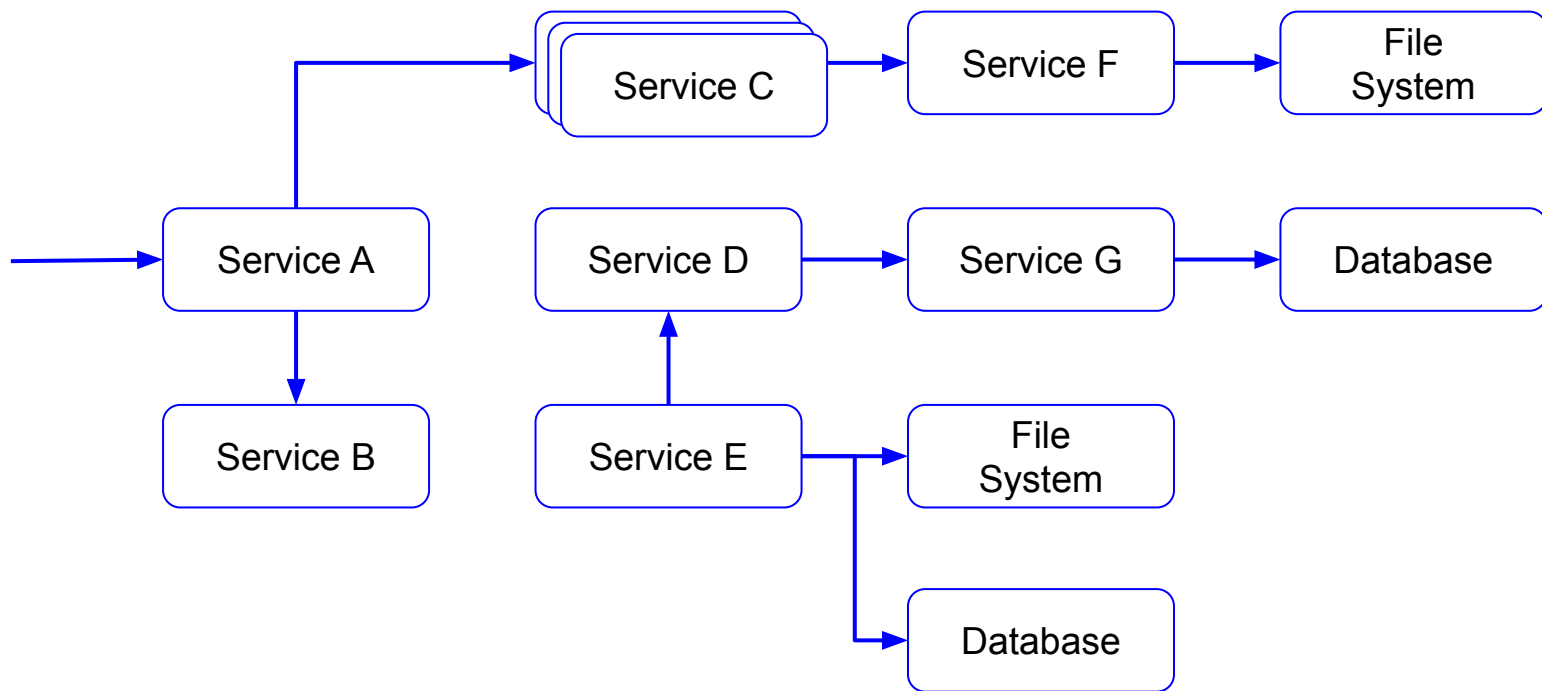
모놀리식 아키텍처

단일 빌드 결과물에 비즈니스 로직 / 데이터 접근 레이어가 들어가서 단순하다.

하지만 특정 모듈에서 병목이 일어나면 배포 환경 전체를 스케일 업 해야한다.

모듈 하나에 변경 사항이 생기면 전체 서비스 단위로 배포가 필요하다.

마이크로 서비스 아키텍처



마이크로 서비스 아키텍처

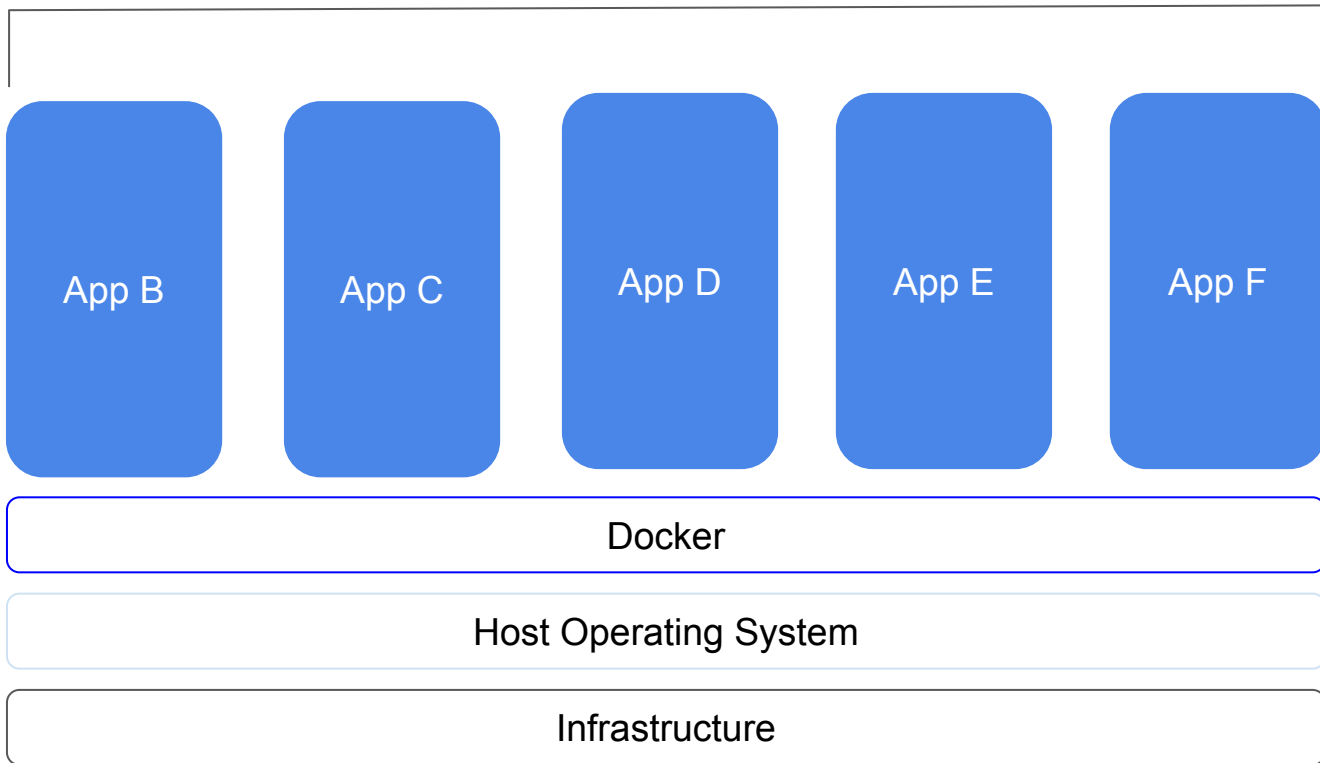
비즈니스 로직이 서비스 단위로 분리되어 있어서 스케일링에 유리하다.

서비스가 격리되어 있어, 서비스 단위로 배포 가능하고, 장애에 강하다.

하지만 초기 세팅과 관리가 복잡하다.

컨테이너화

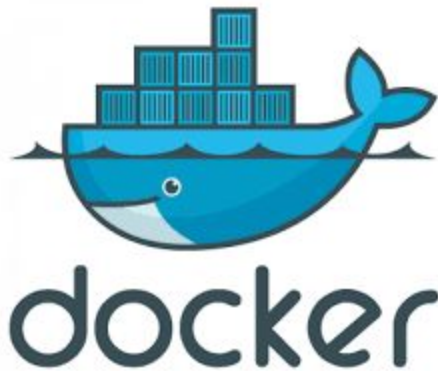
Containerized Applications



컨테이너

애플리케이션 코드와 디펜던시를 패키징하여, 응용 프로그램이 **다른 컴퓨팅 환경에서도 빠르고 안정적으로 실행**되도록 하는 표준 소프트웨어 단위.

가장 대표적인 기술로 **Docker**가 있다.





IT WORKS ON MY MACHINE

THEN WE'LL SHIP YOUR MACHINE

AND THAT IS HOW DOCKER WAS BORN

Namespace & Cgroup

Namespace (프로세스가 볼 수 있는 범위를 제한)

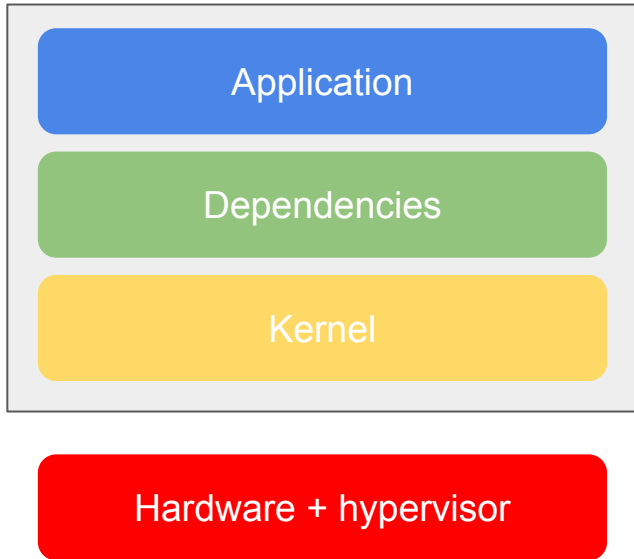
- Process ID (PID)
- Network
- Mount (file system)

Cgroups (프로세스가 사용할 수 있는 자원을 제한)

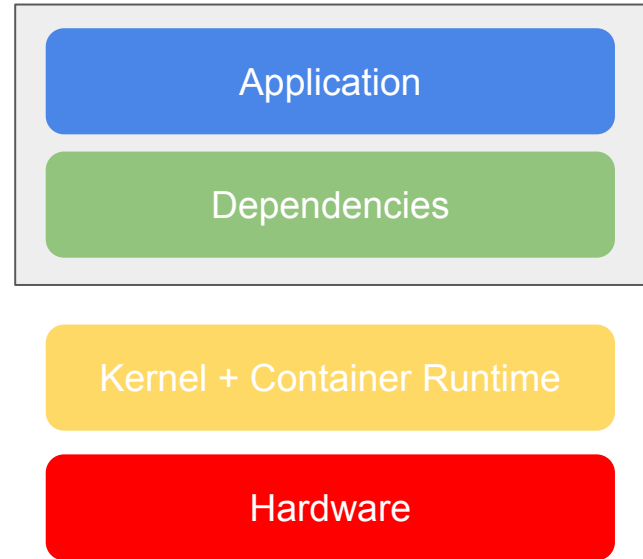
- Memory
- CPU

Virtual machine VS Container

Virtual Machine



Container



컨테이너

리눅스 커널에 존재하는 기능을 사용해서 격리된 실행 환경을 만들고
앱개발자가 선언한 애플리케이션 코드 + 종속성들을 묶어 관리해서
로컬환경, 테스트환경, 배포환경을 일치시키는데 사용 됨.

하나의 하드웨어 또는 가상머신에 여러 서비스를 실행 시킬 수 있다.

Dockerfile 예시

```
FROM ubuntu:latest
```

```
RUN apt-get update && apt-get install -y nodejs npm
```

```
WORKDIR /usr/src/app
```

```
COPY . .
```

```
RUN npm install
```

```
EXPOSE 3000
```

```
CMD ["node", "app.js"]
```

우분투를 베이스 이미지로 삼아서

Node.js와 npm을 설치하고

컨테이너의 **working directory**를 설정하고

파일들을 컨테이너로 복사한 뒤

package.json에 명시 된 의존성을 설치하고

3000 번 포트를 컨테이너 밖에서 접근 가능하게 하고

애플리케이션을 실행한다

컨테이너는 어떻게 관리하죠?

컨테이너를 어떻게 배포 할것인가?

어떻게 관리할 것인가?

컨테이너가 다운 되면 어떻게 할 것인가?

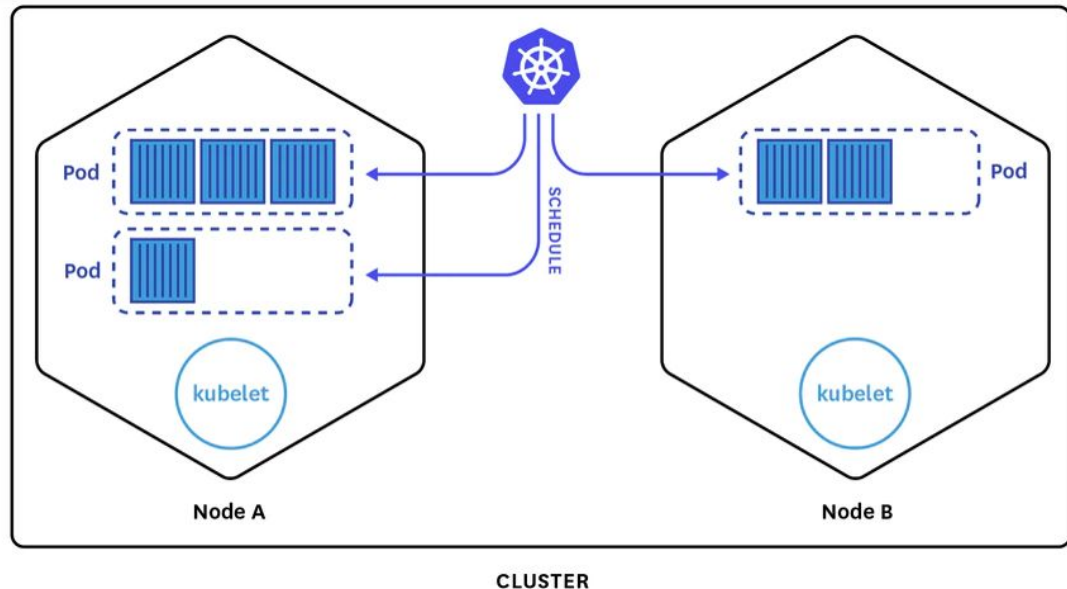
무중단 배포하려면 어떻게 할 것인가?

컨테이너 오케스트레이션

Dockerfile에 컨테이너가 어떻게 동작할지 선언 하는 것 처럼

Kubernetes Manifest 파일에 “바람직한 상태(**Desired State**)”를 정의 해놓으면
배포 환경을 바람직한 상태로 유지한다.

K8S Overview



노드(Nodes): 물리적 또는 가상 머신으로 구성된 쿠버네티스 클러스터에서, 컨테이너 및 필수 쿠버네티스 서비스를 실행합니다.

컨테이너(Containers): 애플리케이션과 그 종속성을 실행하기 위한 경량화되고 이동 가능한 환경입니다.

팟(Pods): 쿠버네티스에서 배포 가능한 가장 작은 단위로, 동일 노드 상의 컨테이너 그룹을 대표하며 공유 리소스를 가집니다.

클러스터(Cluster): 쿠버네티스에 의해 관리되는 노드들의 집합으로, 여러 머신에 걸쳐 컨테이너 배포 및 관리를 가능하게 합니다.