도커

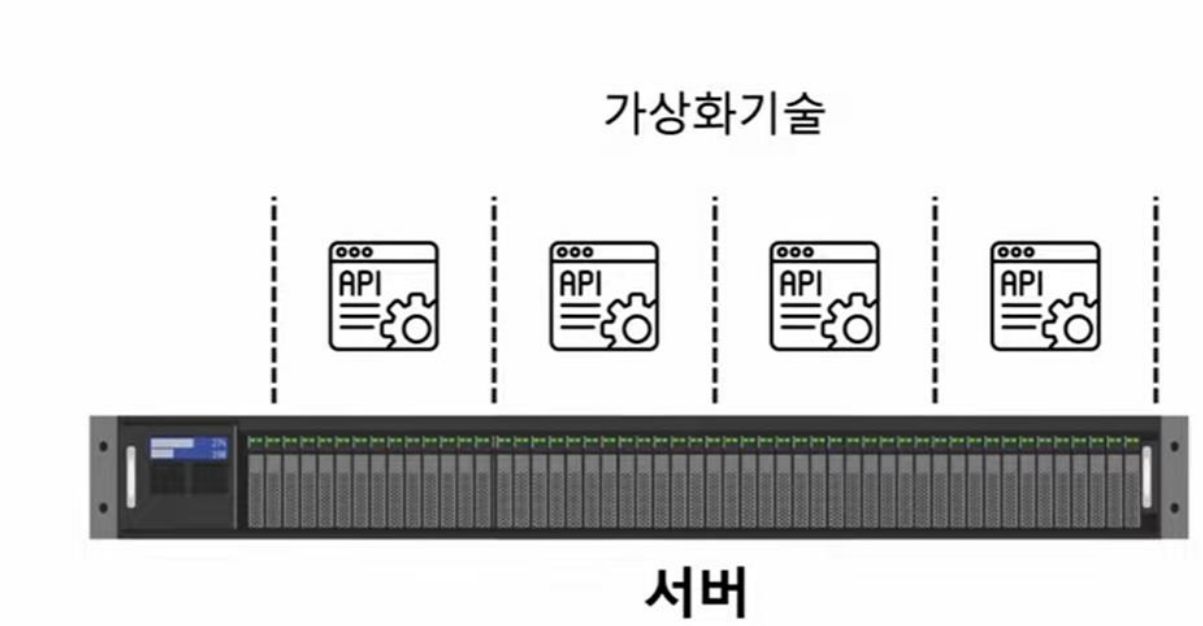
DOCKER

CONTAINER

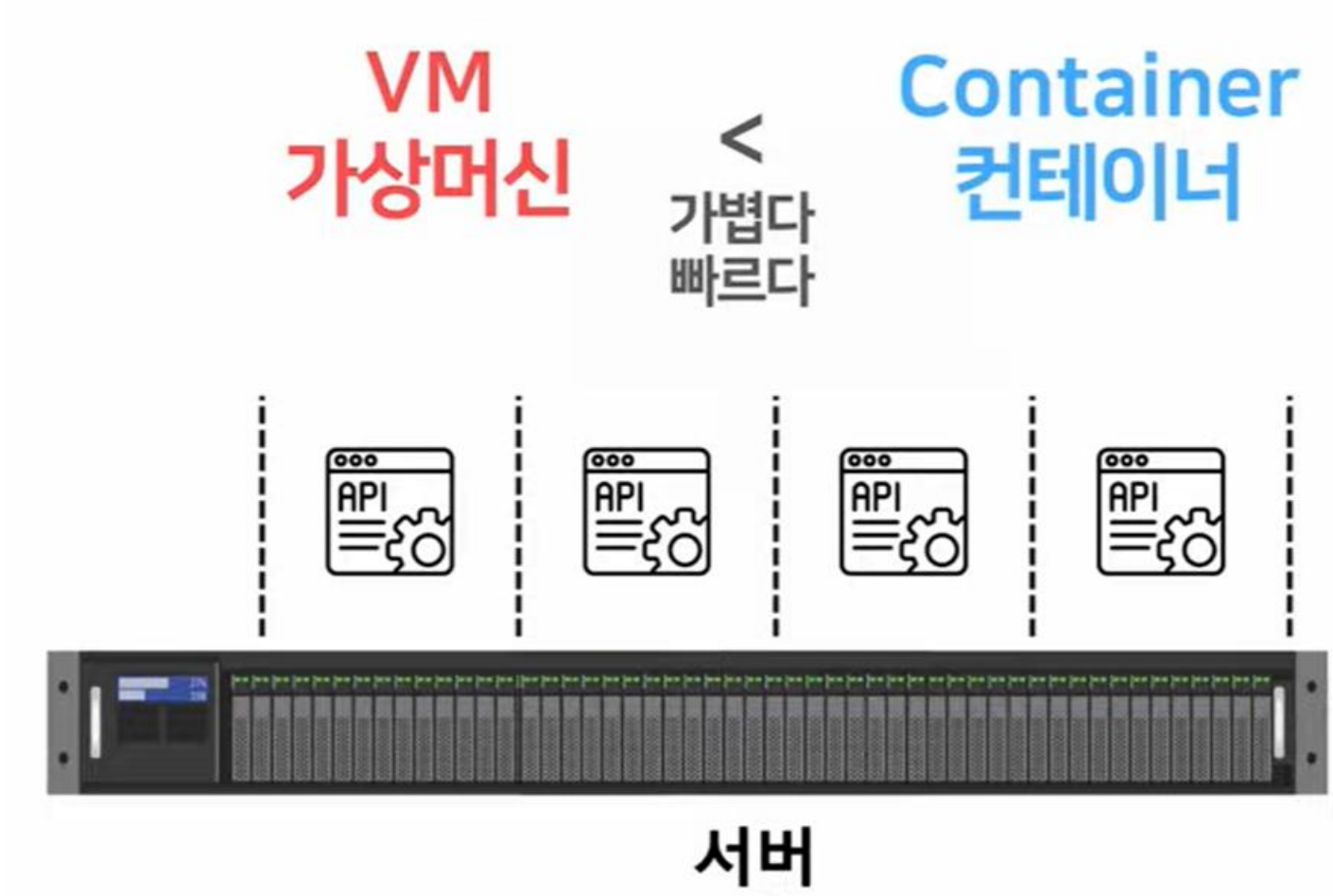
KUBERNETES

컨테이너 = 서버

비효율 해결



하나의 소프트웨어는 하나의 서버에서 실행되어야 안전하다


VM
가상머신
<
가볍다
빠르다
Container
컨테이너
API
API
API
API
서버

DOCKER

컨테이너 관리를 위한 일종의 프로그램

- 도커가 실행 중인 여러 대의 서버를 관리하기 위한 도구
- 오케스트레이션 도구

정리

컨테이너를 사용하면 한 대의 서버에서 여러 개의
소프트웨어를 안전하고 효율적으로 운영할 수 있습니다.

도커는 컨테이너를 관리하기 위한 도구로 일종의 프로그램입니다.

쿠버네티스는 서버가 여러 대 있는 환경에서
각각의 서버의 도커에게 대신 지시해주는 오케스트레이션 도구입니다.

도커(Docker)란?

- **프로그램을 컨테이너(Container)** 라는 독립된 공간에 넣어 어디서든 동일하게 실행되도록 도와주는 기술
- **응용 프로그램을 표준화된 컨테이너 이미지로 패키징하고, 리눅스·윈도우 어디서든 동일한 환경에서 실행할 수 있게 해주는 플랫폼**

도커의 핵심 개념

개념	설명	예시
이미지(Image)	프로그램과 실행 환경을 묶은 '설계도'	Ubuntu, Nginx, Python3 이미지 등
컨테이너(Container)	이미지를 실제로 실행한 **실행 인스턴스**	nginx 컨테이너 실행 중
Docker Hub	이미지 저장소 (GitHub의 도커 버전)	https://hub.docker.com
Dockerfile	이미지를 만드는 '설정 스크립트'	Python 웹 서버 빌드용 Dockerfile 등

도커 설치

단계	명령	설명
1	sudo apt-get remove docker*	기존 버전 제거
2	sudo apt-get update	패키지 정보 갱신
3	curl ... docker.gpg	GPG 키 등록
4	sudo tee ... docker.list	저장소 추가
5	sudo apt-get install docker-ce ...	도커 설치
6	docker --version	설치 확인
7	sudo usermod -aG docker $USER	일반 사용자 권한 추가
8	docker run hello-world	동작 테스트

기본 명령어

기본 정보 확인

명령어	설명
docker version	Docker 클라이언트와 서버의 버전 정보 표시
docker info	Docker 시스템 정보 출력 (엔진 상태, 플러그인, 리소스 등)
docker --help	Docker 명령어 도움말 표시
docker <command> --help	특정 명령어의 상세 도움말 표시 (예: docker container run --help)

Docker 명령어 기본 구조

docker [Management Command] Command [옵션] [인자]

- Management Command는 종종 생략됩니다.
- 예 : docker container run → docker run

컨테이너 관리

실행

명령어	설명
docker run [옵션] <이미지명>	새 컨테이너 생성 및 실행
docker run -d --name <컨테이너명> <이미지명>	백그라운드에서 이름을 지정하여 컨테이너 실행
docker start <컨테이너명>	중지된 컨테이너 시작
docker run --env KEY=VALUE <이미지명>	환경 변수를 설정하여 컨테이너 실행
docker run <이미지명> <CMD>	기본 CMD를 오버라이드하여 컨테이너 실행

조회

명령어	설명
docker ps	실행 중인 컨테이너 목록 표시
docker ps -a	모든 컨테이너 목록 표시 (중지된 컨테이너 포함)

중지 및 삭제

명령어	설명
docker stop <컨테이너명>	실행 중인 컨테이너 중지
docker rm <컨테이너명>	중지된 컨테이너 삭제
docker rm -f <컨테이너명>	실행 중인 컨테이너 강제 삭제

예 :

여러 컨테이너 동시 삭제: docker rm -f container1 container2 container3

모든 Docker 컨테이너 삭제: docker rm -f $(docker ps -aq)

이미지 관리

명령어	설명
docker image ls	모든 이미지 목록 표시
docker image ls <이미지명>	특정 이미지 정보 표시
docker pull <이미지명>	Docker Hub에서 이미지 다운로드
docker rmi <이미지명>	이미지 삭제

네트워크 관리

명령어	설명
docker network ls	도커 네트워크 목록 표시
docker network create <네트워크명>	새 도커 네트워크 생성
docker network connect <네트워크명> <컨테이너명>	컨테이너를 네트워크에 연결
docker network disconnect <네트워크명> <컨테이너명>	컨테이너를 네트워크에서 분리

볼륨관리

명령어	설명
docker volume ls	도커 볼륨 목록 표시
docker volume create <볼륨명>	새 도커 볼륨 생성
docker volume rm <볼륨명>	도커 볼륨 삭제

로그 확인 명령어

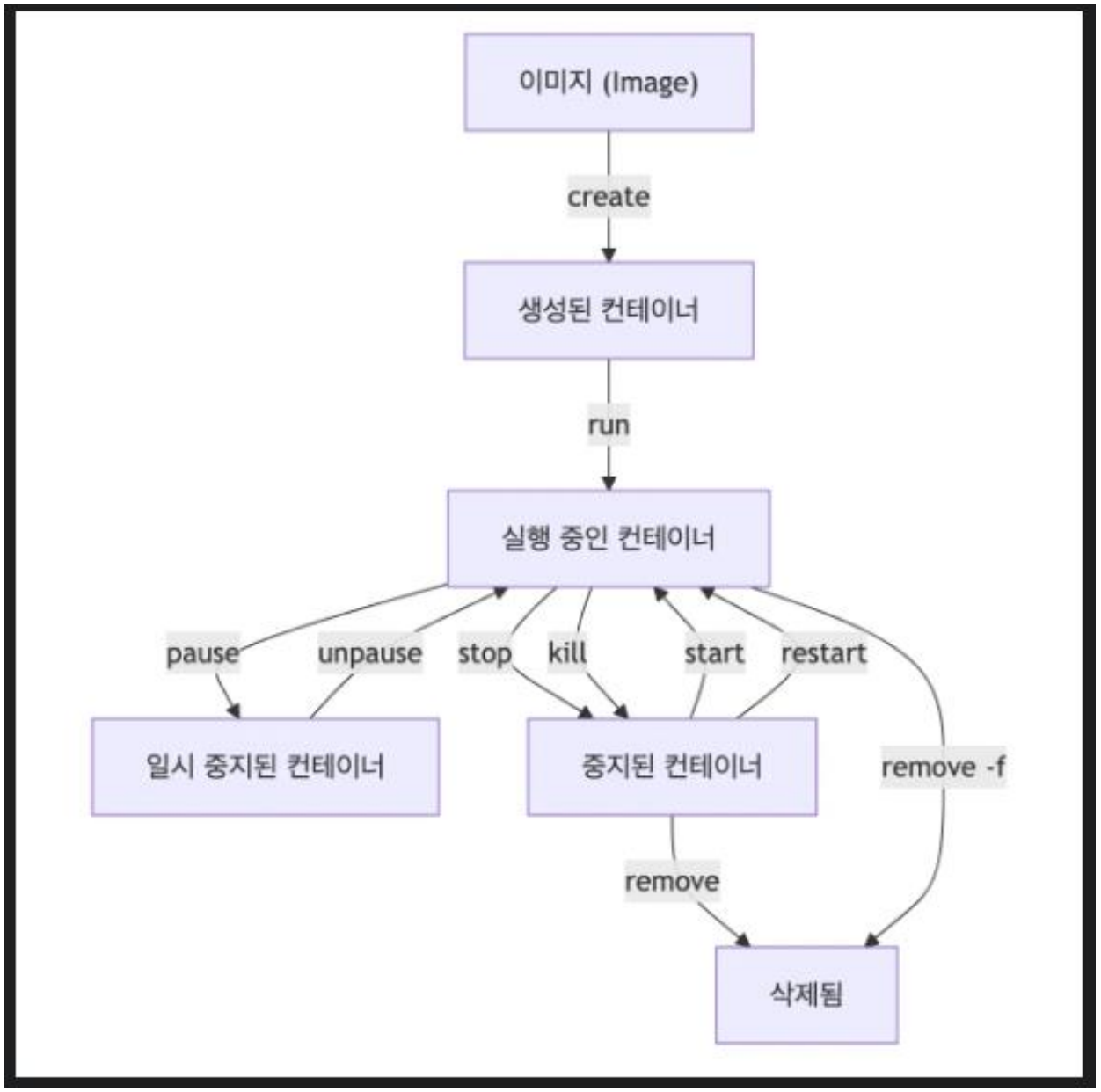
명령어	설명
docker logs <컨테이너명>	지정한 컨테이너의 로그 출력
docker logs -f <컨테이너명>	실시간 로그 표시
docker logs --tail <n> <컨테이너명>	마지막 n개의 로그 라인만 출력
docker logs --since <시간> <컨테이너명>	특정 시간 이후의 로그만 출력
docker logs --until <시간> <컨테이너명>	특정 시간 이전의 로그만 출력
docker logs --timestamps <컨테이너명>	로그와 타임스탬프 함께 표시

기타 유용한 명령어

명령어	설명
docker logs <컨테이너명>	컨테이너 로그 확인
docker exec -it <컨테이너명> <명령어>	실행 중인 컨테이너에 명령어 실행
docker cp <컨테이너명>:<컨테이너 내 경로> <호스트 경로>	컨테이너에서 호스트로 파일 복사
docker inspect <컨테이너명>	컨테이너의 상세 정보 확인

대부분의 Docker 명령어는 컨테이너 ID 대신 컨테이너 이름을 사용할 수 있습니다.
docker run 명령어는 이미지가 로컬에 없으면 자동으로 pull 합니다.
컨테이너 이름을 지정하지 않으면 Docker가 자동으로 임의의 이름을 할당합니다.

컨테이너 라이프 사이클


이미지 (Image)
create
생성된 컨테이너
run
실행 중인 컨테이너
pause
unpause
stop
kill
start
restart
일시 중지된 컨테이너
중지된 컨테이너
remove -f
remove
삭제됨

생성 : 도커 이미지를 기반으로 새로운 컨테이너를 생성합니다.

- 상태: 이 단계에서는 컨테이너가 생성되지만 아직 실행되지는 않습니다.
- 명령어: docker create

실행 : 생성된 컨테이너를 시작하고 실행합니다.

- 상태: 컨테이너 내부의 애플리케이션이 동작하기 시작합니다.
- 명령어: docker start 또는 docker run (docker run은 create와 start를 합친 명령)

일시중지 : 실행 중인 컨테이너를 일시적으로 중지시킵니다.

- 상태: 컨테이너의 상태는 유지되지만, 프로세스 실행은 멈춥니다.
- 명령어: docker pause

재개 : 일시 중지된 컨테이너를 다시 실행 상태로 되돌립니다.

- 명령어: docker unpause

중지 : 실행 중인 컨테이너를 완전히 중지시킵니다.

- 상태: 컨테이너의 상태는 저장되지만, 실행은 완전히 멈춥니다.
- 명령어: docker stop

재시작 : 중지된 컨테이너를 다시 시작합니다. 또는 실행 중인 컨테이너를 중지하고 다시 시작합니다.

- 명령어: docker restart

삭제 : 중지된 컨테이너를 완전히 시스템에서 제거합니다.

- 명령어: docker rm
- 참고: 실행 중인 컨테이너는 강제로 중지하고 삭제할 수 있습니다 (-f 옵션 사용).