

Docker 볼륨과 데이터 영속성

왜 데이터가 사라질까 ?

● 왜 사라질까

- 마운트 3 방식
- 볼륨 사용법
- 볼륨 백업
- 데이터 이전
- 흔한 실수

- Docker 컨테이너는 휘발성 — 삭제하면 안의 파일도 함께 소멸
- docker rm, docker compose down 모두 동일
- 컨테이너 레이어는 이미지 기반 임시 레이어
- 재시작은 남지만 , 삭제 후 재생성하면 깨끗한 상태에서 시작

Docker 마운트 3 가지 방식

- 왜 사라질까

- 마운트 3 방식

- 볼륨 사용법

- 볼륨 백업

- 데이터 이전

- 흔한 실수



Volume

Docker 관리

Docker 가 관리하는 영역에
저장, 이식성 좋음



Bind Mount

호스트 직접 연결

호스트 디렉토리를
컨테이너에 직접 마운트



tmpfs Mount

메모리 저장

휘발성이지만 빠름, 임시
데이터용

Volume 실전 사용법

- 왜 사라질까
- 마운트 3 방식
- 볼륨 사용법
 - 볼륨 백업
 - 데이터 이전
 - 흔한 실수

```
# docker run 에 -v 옵션 추가
docker run -v mydata:/var/lib/postgresql/data
postgres
```

```
# 볼륨 확인
docker volume ls
docker volume inspect mydata
```

```
# 컨테이너를 지워도 볼륨은 남는다!
```

Bind Mount 사용법

- 왜 사라질까
- 마운트 3 방식
- 볼륨 사용법
 - 볼륨 백업
 - 데이터 이전
 - 흔한 실수

```
# 호스트 디렉토리를 컨테이너에 직접 연결
docker run -v /home/user/html:/usr/share/nginx/html
nginx
```

```
# 읽기 전용 마운트 ( 설정 파일용 )
docker run -v
/config/nginx.conf:/etc/nginx/nginx.conf:ro nginx
```

```
# 호스트에서 수정하면 컨테이너에 즉시 반영
```

Named Volume (docker compose)

- 왜 사라질까
- 마운트 3 방식
- 볼륨 사용법
- 볼륨 백업
- 데이터 이전
- 흔한 실수

```
services:
  db:
    image: postgres
    volumes:
      - pgdata:/var/lib/postgresql/data
```

```
volumes:
  pgdata:
```

선언적이라 관리가 깔끔 !
down --v 를 쓰면 볼륨까지 삭제되니 주의

Docker 볼륨 백업 패턴

- 왜 사라질까
- 마운트 3 방식
- 볼륨 사용법
- 볼륨 백업
 - 데이터 이전
 - 흔한 실수

```
# 임시 컨테이너로 볼륨 백업
docker run --rm \
  -v mydata:/data \
  -v $(pwd):/backup \
  ubuntu tar czf /backup/backup.tar.gz /data

# 복원은 반대로 — tar 를 풀어서 볼륨에 넣기
# cron 에 등록하면 자동 백업 완료 !
```

데이터 이전 5 단계

- 왜 사라질까
- 마운트 3 방식
- 볼륨 사용법
- 볼륨 백업
- 데이터 이전
- 흔한 실수

1

기존 서버에서
볼륨을 tar 로
백업

2

백업 파일을
scp/rsync 로 새
서버에 전송

3

새 서버에서
컨테이너 한 번
실행해 빈 볼륨
생성

4

백업을 볼륨에
복원 (tar 해제)

5

컨테이너 재시작
— 데이터가
그대로 이전됨

Docker 데이터 영속성 흔한 실수

- 왜 사라질까
- 마운트 3 방식
- 볼륨 사용법
- 볼륨 백업
- 데이터 이전
- 흔한 실수

- `docker compose down --v` 를 무심코 치면 볼륨이 삭제됨 (치명적!)
- 익명 볼륨 사용 (`-v /data`) → 나중에 찾기 매우 어려움, 반드시 이름 명시
- 권한 문제 무시 → 컨테이너 `root` 가 호스트 파일을 `root` 소유로 만듦
- 볼륨 용량 미확인 → `docker system df` 로 반드시 확인

Docker 데이터 영속성 정리

 구독 & 좋아요

 댓글로 궁금증 남겨주세요

구독과 좋아요 부탁드립니다! 👍