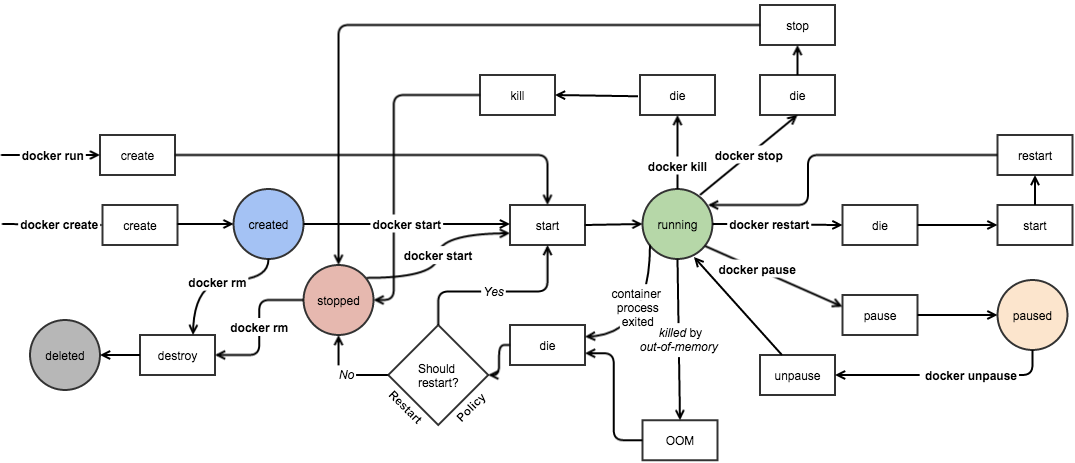
세가지 migration 기법의 동작 방식

내용

[<Full-Copy, FC> 2](#_Toc77713953)

[<Differential-Copy, DC, + user request duplicating> 3](#_Toc77713954)

[<Log-Replay, LR, + user request duplicating> 4](#_Toc77713955)



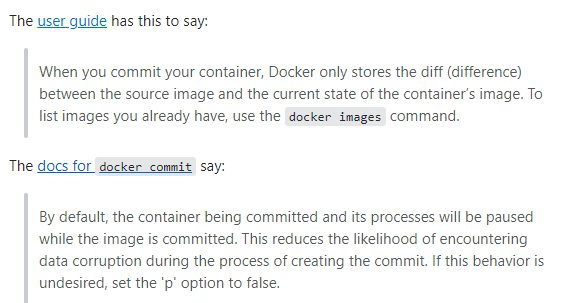
# <Full-Copy, FC>

1. EdgeServer1에서 현재 실행중인 컨테이너를 체크포인트 만들고 (no stop)
2. 컨테이너 전체 이미지를 만들어서 EdgeServer2 로 전송
3. 그 동안 AP2가 받는 user request는 duplicating 해서, 당분간은 AP1-EdgeServer1으로부터 서비스를 받도록 하고(AP2 => AP1 => EdgeServer1), EdgeServer2가 실행하면 그 동안의 request를 forwarding 해주고 migration 완료 (근데, 그냥 AP2에서 큐에다가 REQ를 쌓아 두고 ES2가 준비되면 한번에 전달해 주는 거랑 성능상 큰 차이가 없을 듯?)

|  |  |
| --- | --- |
| ES 1 | 새로운 IMAGE 생성 (no pause): IMG 생성 중에 안정성을 보장하기 위해 일시적으로 pause 한다. 따라서 commit 이후에 즉시 unpause 해 주어야 한다. Commit 할 때 pause를 안하게 하는 방법도 있지만, 안정성 문제가 있어서 그냥 pause하고 unpause 하는게 나을 듯...  $ docker commit CONTAINER\_NAME ANY\_NAME  $ docker unpause CONTAINER\_NAME |
| 체크포인트 생성 (no stop)  $ docker checkpoint create CONTAINER\_NAME CHECKPOINT\_NAME |
| 이미지를 tar archive로 덤프  $ docker save -o ANY\_NAME.tar ANY\_NAME |
| ES2에게 tar 파일을 전송  $ scp ANY\_NAME.tar 아이디@호스트IP:절대경로 |
| ES 2 | Tar archive 파일로부터 이미지 불러오기  $ docker load -i ANY\_NAME.tar |
| 컨테이너 생성(실행은 안함)  $ docker create ANY\_NAME |
| 체크포인트로 컨테이너 실행  $ docker |

\*\* CONTAINER\_NAME이랑 ANY\_NAME을 서로 같게 하자

\*\* 위의 시나리오 대로 실행 가능한지, 즉, FC에서도 live migration이 가능한지를 확인하자

<URL>

# <Differential-Copy, DC, + user request duplicating>

1. EdgeServer2에서 동일한 base image로 컨테이너 생성(create만 하고, run은 안함)
2. EdgeServer1은 Container layer의 파일을 EdgeServer2로 전송
3. EdgeServer1에서 체크포인트 생성 (no stop)하고, 체크포인트 전송
4. EdgeServer2에서 체크포인트로 컨테이너 실행
5. 그 동안 AP2가 받는 user request는 duplicating 해서, 당분간은 AP1-EdgeServer1으로부터 서비스를 받도록 하고(AP2 => AP1 => EdgeServer1), EdgeServer2가 실행하면 그 동안의 request를 forwarding 해주고 migration 완료 (근데, 그냥 AP2에서 큐에다가 REQ를 쌓아 두고 ES2가 준비되면 한번에 전달해 주는 거랑 성능상 큰 차이가 없을 듯?)

# <Log-Replay, LR, + user request duplicating>

1. EdgeServer2에서 동일한 base image로 컨테이너 실행(run을 함)
2. EdgeServer1에서 컨테이너에 적용된 모든 command log를 뽑아서 EdgeServer2로 전송
3. EdgeServer2는 log를 모두 순서대로 replay
4. 그 동안 AP2가 받는 user request는 duplicating 해서, 당분간은 AP1-EdgeServer1으로부터 서비스를 받도록 하고(AP2 => AP1 => EdgeServer1), EdgeServer2가 실행하면 그 동안의 request를 forwarding 해주고 migration 완료 (근데, 그냥 AP2에서 큐에다가 REQ를 쌓아 두고 ES2가 준비되면 한번에 전달해 주는 거랑 성능상 큰 차이가 없을 듯?)