

< 멀티호스트 구성 >

2018. 12. 13(목)

1. 설명

- docker에서 분산 네트워크를 관리해주는 플랫폼(overlay network 통해 관리) = docker swarm
 - docker swarm 초기화(초기화 시킨 해당 PC가 대장이 되고 나머지는 manager, worker가 됨)
 - 모든 host들이 공유하기 위한 overlay 네트워크 생성이 필요(swarm에서 사용하기 위한 넷월)
 - ssh -R 52698:localhost:52698 username@hostname
- guru

guru, guru2

: username은 접속하는 내 이름일 뿐이고 hostname이 네트워크의 이름
(alias로 주소, 이름 등 다 가능)

- bmhn(Build-Multi-Host-Network-Hyperledger)
- parent에서 생성한 bufn의 결과(CA, MSP 등)와 child가 같아야 함.
- USB를 이용해 ubuntu에서 마운트 시켜 받아 올 수도 있지만 어려움.
- 가장 쉬운 방법으로 git repository를 이용!!

git에 새로운 repository를 생성 → PC1에서 crypto-config, channel-artifacts 파일을 복사해서
git에 올리기 → PC2에서 git에 접속해 clone으로 다운받기(이후 새로 고친 내용이 있다면
\$ git pull 이용)

- PC1에서 올리고 PC2에서 git-clone하면 remote repository에 있는 것을 local에 가져오게 됨!
 local에서 remote repository에 파일을 올리는 것이 commit
 근데 git은 바로 안되고, add를 통해 마킹을 하고 commit을 해줘야 함
 add(1단계 local repository) : 인터넷이 안 되는 경우 서버에 가지 않고 working dir에서 local로만 올림
 commit(local에서 remote repository로 올리는 과정) : ID, 이메일주소 필요(누가 올리는지 알아야하니까)
 push(2단계 remote repository) : working dir에 있는 놈을 한 번에 remote로 올리는 것
 : 이 때는 ID / PW가 다 필요함(remote에 접속하는 것이므로)

fetch(remote repository에서 local repository로 땡겨오는 것)

merge(local repository에 있는 걸 working directory로 합쳐주는 것!)

(github는 굉장히 단순하게 라인별 비교함(같은 라인이 수정되거나 하면 충돌 발생))

(이럴 경우 내가 가지고 있는 파일을 어딘가에 백업해두고 git에서 다운받아 일일이

수작업으로 비교해보는 방법 뿐)

pull(fatch하고 머지하기 귀찮으니까 한큐에 처리해주는 것)

- docker run하면 없는 애도 설치해서 up 시켜줌
exec하면 있는 애들만 up 시켜줌