파일시스템 동기화 환경 구축(rsyncd, lsyncd)

구축 목표

: 서로 다른 두 개의 서버의 파일시스템 동기화

Port : 873 / rsync

서비스 의미

rsyncd : 효율적으로 파일시스템을 동기화해주는 역할 -> 파일의 timestamp와 size를 검사하여 변경된 파일만 동기화 시행 ★실질적인 동기화 담당, 원격 동기화

lsyncd : 실시간 동기화를 해주는 역할 -> 파일 시스템 이벤트 모니터링 inotify를 통해 SRC 서버의 파일시스템이 바뀌었을때 rsync를 이용하여 동기화 진행 ★실시간 모니터링을 통해 변화에 따른 자동 동기화

서버 구축 진행

rsync 동기화 환경 구축

```
# rpm -qa | grep rsync
( 없으면 설치 )

# mkdir rsytest
# cd rsytest
# touch test1 a.txt b.txt c.txt a.java a.sql

# systemctl start rsyncd
# systemctl enable rsyncd

# rsync -avz /root/rsytest/ 175.106.99.250:/root/rsytest
( /root/rsytest/의 파일들을 175.106.99.250 DEC서버의 /root/rsytest로 파일 동기화 )
```

- 옵션 -a : archive mode (= -rlptgoD) -r : 하위 디렉토리 -l : symbolic link -p : permission(권한) -t : timestamp(최종 수정일) -g : 그룹 -o : 소유자, super user only -v :동기화 내역 상세 출력 -X : xattr -A : ACL -H : hard link --exclude : 제외할 파일확장자 설정

---

lsync 실시간 동기화 환경 구축

```
# yum -y install lsyncd
( 설치 안될시 epel-release 설치 후 진행 )

# vi /etc/lsyncd.conf
-----------------------------------------------------------
settings {
```

```
        logfile = "/var/log/lsyncd/lsyncd.log",
        statusFile = "/var/log/lsyncd/lsyncd-status.log",
        insist = 1
}

sync {
        default.rsyncssh,
        source="/root/rsytest",
        host="175.106.99.250",
        targetdir="/root/rsytest",
        rsync = {
                archive=true,
                compress=true,
                verbose=true
        }
}
-----------------------------------------------------------
- 로그파일 위치 및 출발지와 도착지 디렉토리 설정 등 진행

# vi /etc/ssh/sshd_config
-----------------------------------------------------------
38 PermitRootLogin yes  ( root 로그인 시 yes 설정 )
43 PubkeyAuthentication yes  ( 키를 통한 접속을 위해 공개키인증 yes )
80 #GSSAPIAuthentication yes
81 #GSSAPICleanupCredentials no
-----------------------------------------------------------

# cd ~/.ssh/
# ssh-keygen -t rsa
( 키 생성 - Master 서버에서 진행 )
# ssh-copy-id 175.106.99.250
( 생성한 키 복사해서 Backup 서버로 전달 )

  ▶ id_rsa: 개인키(private key)이며 접속하고자 하는 클라이언트(client)가 가지고 있는
다.

  ▶ id_rsa.pub: 공개키(public key)이며 서버(server)가 가지고 있는다.

  ▶ authorized_keys: 서버가 접속을 허용할 공개키(public key) 리스트다.
```

★ 키 접속 구조 ★

1. 먼저 접속할 Master 서버에서 개인키(id_rsa)와 공개키(id_rsa.pub)를 생성한다.
2. 접속을 허용할 Backup 서버에서 공개키 리스트(authorized_keys)를 생성하고 Master서버의 공개키(id_rsa.pub) 내용을 넣어준다.
3. 이제 Master 서버에서 개인키(id_rsa)를 통하여 Backup서버에 접속이 가능하다.

-> Master 서버에서 개인키로 데이터를 암호화하고 Backup 서버에서 공개키로 복호화

- 디렉토리 및 파일 권한 변경!

```
   /root 디렉토리 권한 : 700
   /.ssh/authorized_keys 권한 : 600, 644
   /.ssh 권한 : 700
( 확인 및 변경 진행 )
```

```
# systemctl start lsyncd
# systemctl enable lsyncd

# systemctl start sshd
# systemctl enable sshd

# ssh 175.106.99.250
( Master 서버에서 ssh접속을 통해 공개키로 접근이 되는지 확인 )

# cd /root/rsytest
( Backup 서버로 들어가서 Master 서버에 파일이 잘 동기화 되었는지 확인 )


* 오류 발생 시 /var/log/lsyncd/lsyncd.log
         /var/log/lsyncd/lsyncd-status.log
                /var/log/secure
                /var/log/messages
                systemctl status lsyncd
                등으로 확인 가능


* 양방향 동기화 설정 시에는 같은 방법으로 Backup서버에도 lsyncd 설정을 진행하면 됨.
```

TEST

정상적으로 동기화 됐는지 확인! -> 필요에의해 /etc/rsyncd.conf 설정을 통하여 서비스 대상 디렉토리나, chroot기능, 호스트별 접속 허용, 거부 설정 가능 -> 맨 하단에 사이트 참고

★ 정상적으로 작동된다면 파일시스템 동기화 환경 구축 완료 ! ★

참조

- https://blog.jiniworld.me/112
- https://min-nine.tistory.com/82
- https://fendys.tistory.com/211