

Linux Server (for centos 7.X)

- Data Backup -

Copyright @ 2016 MajunSoft co.,Ltd

소 속	IDC실
이 름	신용우 매니저
E-mail	tech@tongkni.co.kr

통큰아이

INDEX

- 1. 개요 3
- 2. 데이터 백업 스크립트 작성. 4
- 3. **crontab**을 이용한 자동 백업 예약. 7
- 4. 활용 - 다른 서버에 백업하기. 10

1. 개요.

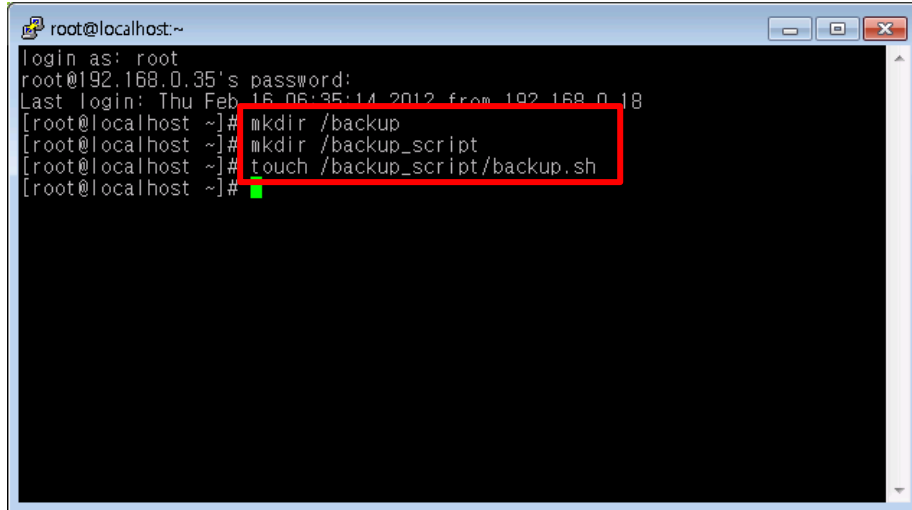
- 서버를 운영하는데 있어 백업은 아무리 강조해도 지나치지 않습니다.
만약 웹 서버를 운영하고 있는 도중에 하드디스크 장애가 생겼다고 가정할 때, 백업이 되어 있지 않았다면 그 동안 운영해왔던 데이터는 전부 혹은 일부를 잃어버리게 됩니다.
게다가 복구하는 동안 서비스를 할 수 없게 되므로 그 기간 동안 서비스를 운영하지 못한 금전적인 손실 또한 발생합니다.
그러므로 서버 운영에 있어 백업은 필수입니다.
- 백업을 하는 방법에는 여러 가지 있는데 본 매뉴얼에서는 리눅스 기본 명령어 중 복사(rsync) 명령어와 예약작업(cron)을 이용하여 정기적으로 백업하는 방법에 대해 안내해드리고자 합니다. 리눅스 서버에 기본적으로 내장되어 있기 때문에 별도의 소프트웨어를 구매할 필요가 없어 경제적이고 간단하게 백업을 할 수 있습니다.
- 백업의 절차
 1. 스크립트 작성 : rsync 명령어를 이용한 실제 백업을 실행 시키는 명령어들의 모음으로 sh 파일로 생성 (실제로 이 스크립트 파일을 실행하면 백업이 이루어진다.)
 2. crond의 설정파일인 crontab을 수정하여 백업 예약 : crond는 스크립트 파일을 반복적으로 실행 하기 위한 데몬. 반복적으로 실행 시킬 시간이나 날짜를 지정.

※ 참고 : 작업 예약과 스크립트를 이용하면 백업 이외에도 여러 가지 작업을 해당 시간에 실행 시켜서 효율적으로 서버를 관리할 수 있습니다.
- 백업은 아래와 같이 구성할 수 있습니다. (백업 안정성 : 1<2<3)
 1. 같은 하드디스크에 백업.
 2. 같은 서버 내에 다른 하드디스크를 추가하여 백업.
 3. 다른 서버의 하드디스크에 백업.본 매뉴얼에서 rsync와 crond를 이용하여 자동 백업이 되도록 설정하는 방법에 대해 안내해드리겠습니다.

2. 데이터 백업 스크립트 작성.

- 아래와 같이 backup폴더(데이터를 백업할 폴더), backup_script 폴더(백업스크립트 및 백업결과 파일을 저장할 폴더) 및 backup.sh(백업스크립트) 파일을 생성합니다.

- ➔ mkdir /backup
- ➔ mkdir /backup_script
- ➔ touch /backup_script/backup.sh

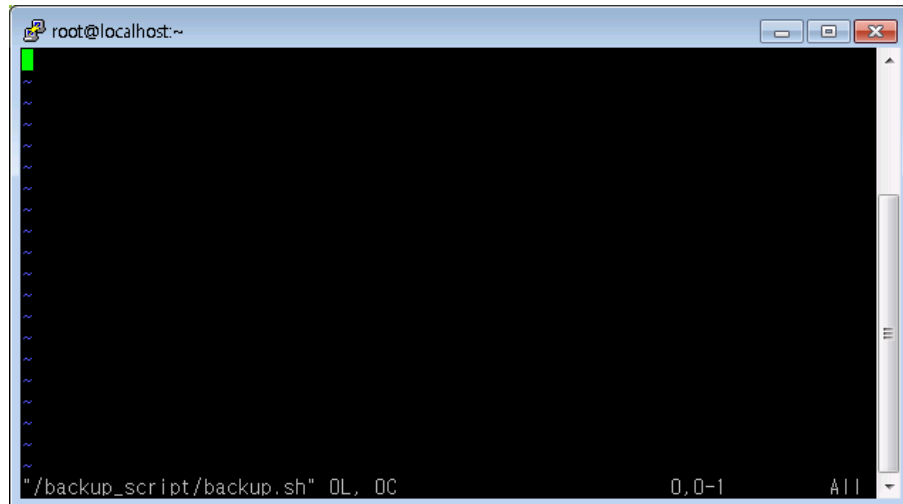


```

root@localhost:~
login as: root
root@192.168.0.35's password:
Last login: Thu Feb 16 06:35:14 2012 from 192.168.0.18
[root@localhost ~]# mkdir /backup
[root@localhost ~]# mkdir /backup_script
[root@localhost ~]# touch /backup_script/backup.sh
[root@localhost ~]#
  
```

- vim 편집기 명령어를 이용해 backup.sh 파일을 수정하기 위해 아래 명령어를 실행합니다.

- ➔ vim /backup_script/backup.sh



```

root@localhost:~
~/backup_script/backup.sh 0L, 0C          0, 0-1          All
  
```

- 편집기에서 “i”를 눌러 입력모드로 전환하고 아래 예시대로 백업스크립트를 작성 후 “:wq!”를 입력하여 저장합니다.

Ex) 같은 서버 내 /home 폴더 데이터를 /backup으로 백업 후 home_backup_result.txt 파일에 로그를 남기는 스크립트 예시.

- ➔ echo "\$(date +%Y년%m월%일) home폴더 증분 백업 결과" >> /backup/home_backup_result.txt
- ➔ rsync -avr /home /backup >> /backup/home_backup_result.txt
- ➔ :wq!

```

root@localhost:/backup
echo '$(date +%Y년%m월%d일) home폴더 증분 백업 결과 ' >> /backup/home_backup_result.txt
rsync -avrz /home /backup >> /backup/home_backup_result.txt
~
~
~
~
~
~
~
~
~
~
~
:wq!

```

- 4 해당 스크립트 파일에 실행권한을 부여합니다.
→ `chmod 700 /backup_script/backup.sh`

```

root@localhost:/backup
drwxr-xr-x  2 root root 12288 Feb 16 03:06 sbin
drwxr-xr-x  2 root root  4096 Feb 16 01:35 selinux
drwxr-xr-x  2 root root  4096 May 11  2011 srv
drwxr-xr-x 11 root root    0 Feb 16  2012 sys
drwxrwxrwt  4 root root  4096 Feb 16 04:05 tmp
drwxr-xr-x 14 root root  4096 Feb 16 01:37 usr
drwxr-xr-x 20 root root  4096 Feb 16 01:38 var
[root@localhost /]# cd backup
[root@localhost backup]# ll
total 4
drwxr-xr-x 5 root root 4096 Feb 16 06:58 home
[root@localhost backup]# rm -rf home
[root@localhost backup]# ll
total 0
[root@localhost backup]# vim /backup_script/backup.sh
[root@localhost backup]# vim /etc/sysconfig/i18n
[root@localhost backup]# source /etc/sysconfig/i18n
[root@localhost backup]# vim /backup_script/backup.sh
[root@localhost backup]# chmod 700 /backup_script/backup.sh
[root@localhost backup]#

```

- 5 해당 스크립트를 실행하고 데이터 백업이 잘 되었는지 로그를 확인합니다.
→ `/backup_script/backup.sh`
→ `cat /backup/home_backup_result.txt`

```
root@localhost:/
[root@localhost /]# ./backup_script/backup.sh
[root@localhost /]# cat /backup/home_backup_result.txt
2012년 02월 16일 home폴더 증분 백업 결과
sending incremental file list
home/
home/test1/
home/test1/testfile1
home/test1/testfile2
home/test1/testfile3
home/test2/
home/test2/testfile1
home/test2/testfile2
home/test2/testfile3
home/test3/
home/test3/testfile1
home/test3/testfile2
home/test3/testfile3

sent 544 bytes  received 199 bytes  1486.00 bytes/sec
total size is 0  speedup is 0.00
[root@localhost /]#
```

- 6 /home.폴더에 있던 데이터가 /backup 폴더에 실제로 존재하는지 확인합니다. 백업 스크립트 작성이 완료되었으므로 [chapter 3](#)을 참고하여 crontab에 등록하는 작업을 진행합니다.
- `ls -R /backup`

```
root@localhost:/backup
[root@localhost backup]# ls -R /backup
/backup:
home  home_backup_result.txt

/backup/home:
test1 test2 test3

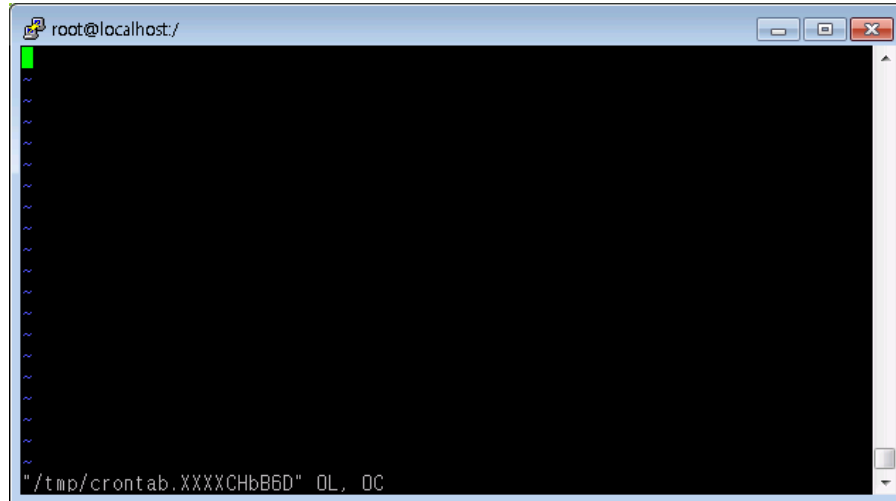
/backup/home/test1:
testfile1 testfile2 testfile3

/backup/home/test2:
testfile1 testfile2 testfile3

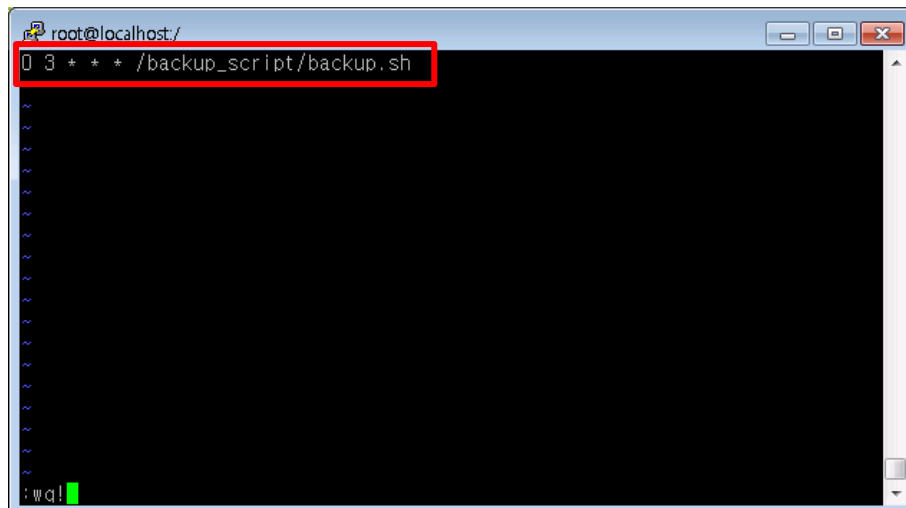
/backup/home/test3:
testfile1 testfile2 testfile3
[root@localhost backup]#
```

3. Crontab을 이용한 자동 백업 예약.

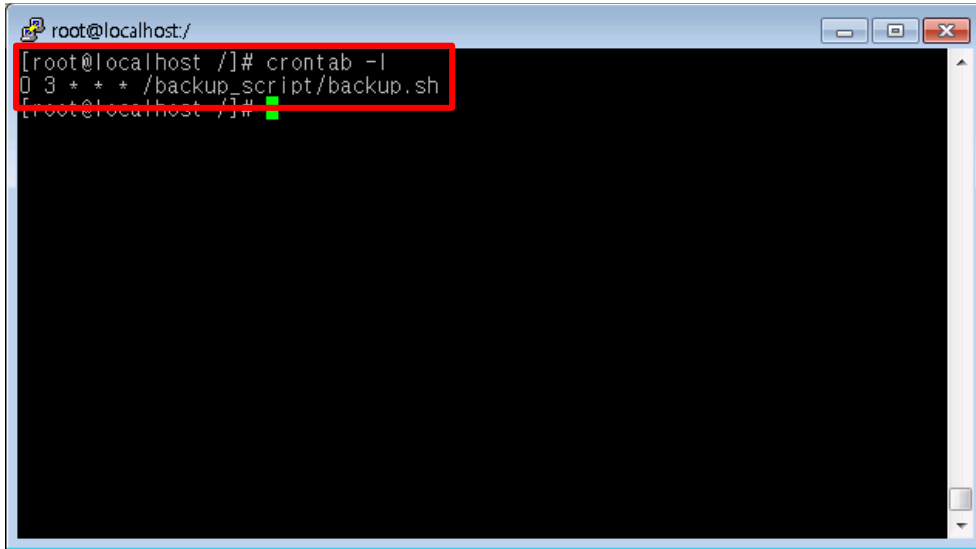
- 1 crond는 일정한 시간마다 명령을 자동 실행하는 기능을 가지고 있으며 crontab 파일을 참조합니다. 아래 명령어를 실행하면 crontab을 수정하기 위한 편집창이 실행됩니다..
→ crontab -e



- 2 편집기에 실행할 명령어와 시간을 아래와 같은 양식대로 등록해야 합니다.
<분> <시> <일> <주> <월> <실행할명령어>
Ex) 매일 새벽 3시 정각에 /backup_script/backup.sh 스크립트를 실행하도록 예약.
→ 0 3 * * * /backup_script/backup.sh
→ :wq!

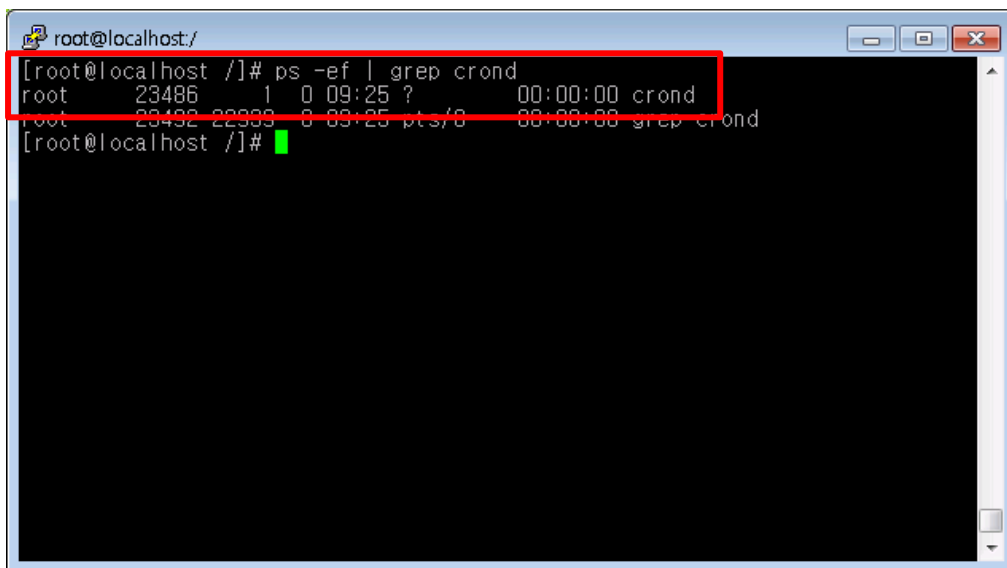


- 3 crontab에 설정한 내용을 보려면 "crontab -l", 설정된 내용을 지우려면 "crontab -e" 명령어를 사용합니다.



```
root@localhost /  
[root@localhost /]# crontab -l  
0 3 * * * /backup_script/backup.sh  
[root@localhost /]#
```

- 4 crontab에 설정한 내용이 실행되려면 서버에 crond 서비스가 구동되어 있어야 합니다. 아래 그림처럼 “crond”가 구동되고 있는지 확인합니다.
- ps -ef | grep crond
 - crond가 목록에 없을 경우 : service crond start



```
root@localhost /  
[root@localhost /]# ps -ef | grep crond  
root 23486 1 0 09:25 ? 00:00:00 crond  
root 23492 22993 0 09:25 pts/0 00:00:00 grep crond  
[root@localhost /]#
```

- 5 crontab에 등록된 스크립트가 해당 시간에 정상적으로 실행 되었는지 새로 생성된 백업데이터 및 로그를 확인합니다.


```
root@localhost:/backup
[root@localhost backup]# ls -al /backup/home/
합계 24
drwxr-xr-x 6 root root 4096 2월 16 02:59 .
drwxr-xr-x 3 root root 4096 2월 16 2012 ..
drwxr-xr-x 2 root root 4096 2월 16 2012 test1
drwxr-xr-x 2 root root 4096 2월 16 2012 test2
drwxr-xr-x 2 root root 4096 2월 16 2012 test3
drwxr-xr-x 2 root root 4096 2월 16 02:59 test4
[root@localhost backup]#
[root@localhost backup]#
```

```
root@localhost:/backup
[root@localhost backup]# tail -10 /backup/home_backup_result.txt
2012년 02월 16일 home폴더 증분 백업 결과
sending incremental file list
home/
home/test4/
home/test4/testfile_1
home/test4/testfile_2
home/test4/testfile_3

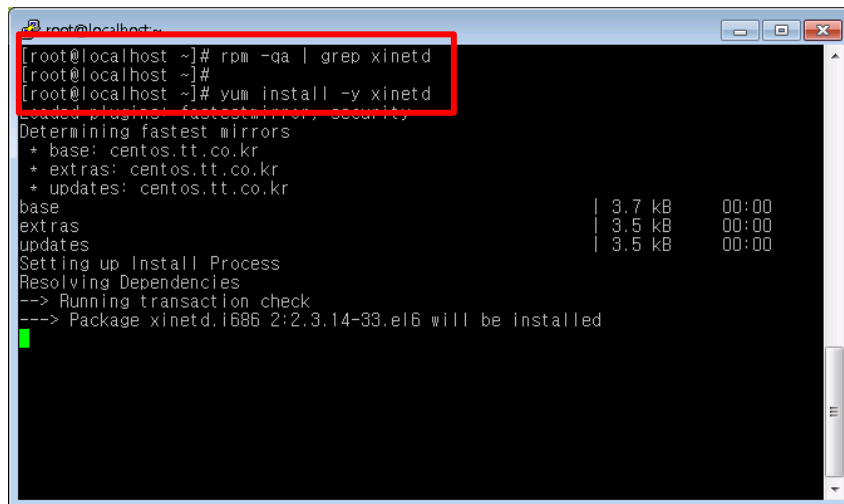
sent 377 bytes received 80 bytes 914.00 bytes/sec
total size is 0 speedup is 0.00
[root@localhost backup]#
```

4. 활용 - 다른 서버에 백업하기.

- 이 챕터는 원본 서버의 데이터를 다른 IP의 백업 서버로 백업하는 방법에 대해 작성하였습니다.
- 서로 다른 서버의 데이터를 복사할 때 보안 상 패스워드를 물어보게 되는데, 이런 경우 자동으로 예약 작업을 진행할 수 없습니다. 따라서, 본 서버의 /etc/rsyncd.conf 설정 파일 내용을 적절히 수정하여 백업 서버에서 rsync 요청이 들어올 경우 패스워드를 묻지 않고 처리되도록 한 후 백업 스크립트를 작성합니다.
- 작업 순서.
 1. 본 서버 - "rpm -qa | grep xinetd" 명령으로 xinetd가 설치되어있는지 확인.
-> 설치되어 있지 않은 경우 : "yum install -y xinetd" 입력하여 설치.
 2. 본 서버 - 방화벽에 rsync가 사용하는 포트인 873번 포트를 허용.
 3. 본 서버 - /etc/rsyncd.conf를 수정.
 4. 백업서버 - 백업 스크립트 생성.
 5. 백업서버 - 백업 스크립트를 crond 예약작업에 추가.

1 원본 서버에서 xinetd가 설치되어 있는지 확인하고, 설치되어 있지 않은 경우 설치를 진행합니다.

- ➔ rpm -qa | grep xinetd
- ➔ xinetd가 없을 경우 아래 명령어 실행.
yum install -y xinetd
- ➔ 설치 완료 후 아래 xinetd 서비스 시작.
service xinetd start



```
[root@localhost ~]# rpm -qa | grep xinetd
[root@localhost ~]#
[root@localhost ~]# yum install -y xinetd
Loaded plugins: fastestmirror, security
Determining fastest mirrors
 * base: centos.tt.co.kr
 * extras: centos.tt.co.kr
 * updates: centos.tt.co.kr
base                               | 3.7 kB    00:00
extras                             | 3.5 kB    00:00
updates                            | 3.5 kB    00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package xinetd.i686 2:2.3.14-33.el6 will be installed
```

2 본 서버에서 방화벽이 구성되어 있는 경우 873번 포트를 허용하도록 추가합니다.

- ➔ vim /etc/sysconfig/iptables
- ➔ 아래 구문 추가 후 저장.
-A INPUT -m state --state NEW -m tcp -p tcp --dport 873 -j ACCEPT
- ➔ 방화벽 설정을 적용하기 위해 서비스 재 시작.
service iptables restart

```

root@localhost:~
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
# 백업을 위한 rsync 포트 허용
-A INPUT -m state --state NEW -m tcp -p tcp --dport 873 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
~
~
~
~
~
~
10,64   모두

```

3. 아래는 백업서버의 IP: **192.168.0.35** 에서 별명이 **"home"**인 rsync 명령을 요청 시 **/home** 폴더를 **패스워드 요청 없이 복사**하도록 허용하는 예시입니다. 원본 서버에서 vi /etc/rsyncd.conf로 아래 내용을 추가합니다.
- vi /etc/rsyncd.conf 후 아래 내용 추가 후 저장.

```

[home]
path = /home
comment = home
uid = root
gid = root
use chroot = yes
read only = yes
hosts allow = 192.168.0.35

```

:wq!

```

root@localhost:~
[home]
path = /home
comment = home
uid = root
gid = root
use chroot = yes
read only = yes
hosts allow = 192.168.0.35
~
~
~
~
~
~
:wq!

```

※ rsyncd.conf 파일의 각 항목별 의미는 아래와 같습니다.
[home] : rsync 요청 때 사용할 별명을 home으로 명명.

path : 위에서 명명된 별명을 다른 서버에서 rsync 요청 시 path로 지정된 경로를 복사.
comment : 해당 rsync에 대한 설명.
uid : 실행할 유저 id
gid : 실행할 그룹 id
use chroot : yes로 설정하면 path로 설정된 디렉토리의 상위 디렉토리는 접근하지 못하도록 함.
read only : rsync 실행 때 원본 서버의 데이터를 읽기 전용으로 설정.
hosts allow : 설정된 IP는 패스워드 없이 허용함.

※ 백업서버에서 본 서버의 /home 폴더를 root 계정으로 rsync 요청하여 백업서버의 /backup에 복사하는 예시.

➔ `rsync -avrz root@192.168.0.35::home /backup`
 (주의) home 앞의 콜론(:)을 두 번 입력해야 함.

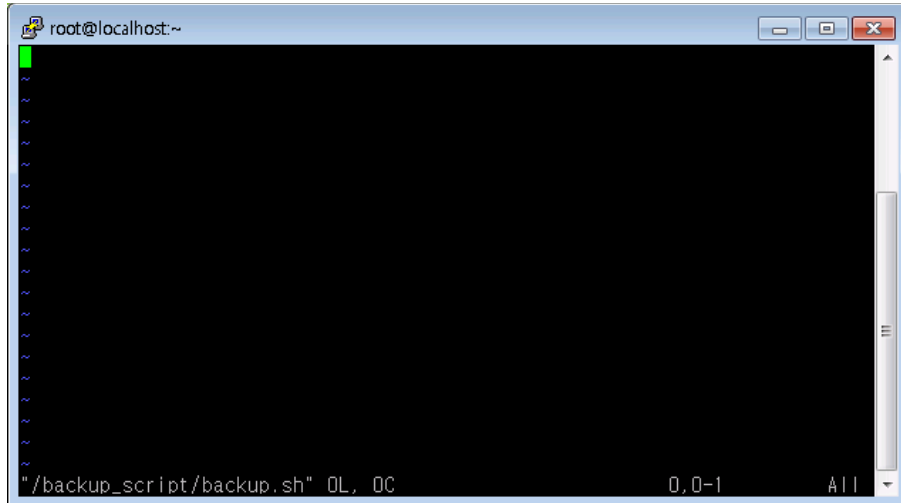
4 아래와 같이 backup폴더(데이터를 백업할 폴더), backup_script 폴더(백업스크립트 및 백업결과 파일을 저장할 폴더) 및 backup.sh(백업스크립트) 파일을 생성합니다.

- ➔ `mkdir /backup`
- ➔ `mkdir /backup_script`
- ➔ `touch /backup_script/backup.sh`

```

root@localhost ~
login as: root
root@192.168.0.35's password:
Last login: Thu Feb 16 06:35:14 2012 from 192.168.0.18
[root@localhost ~]# mkdir /backup
[root@localhost ~]# mkdir /backup_script
[root@localhost ~]# touch /backup_script/backup.sh
[root@localhost ~]#
  
```

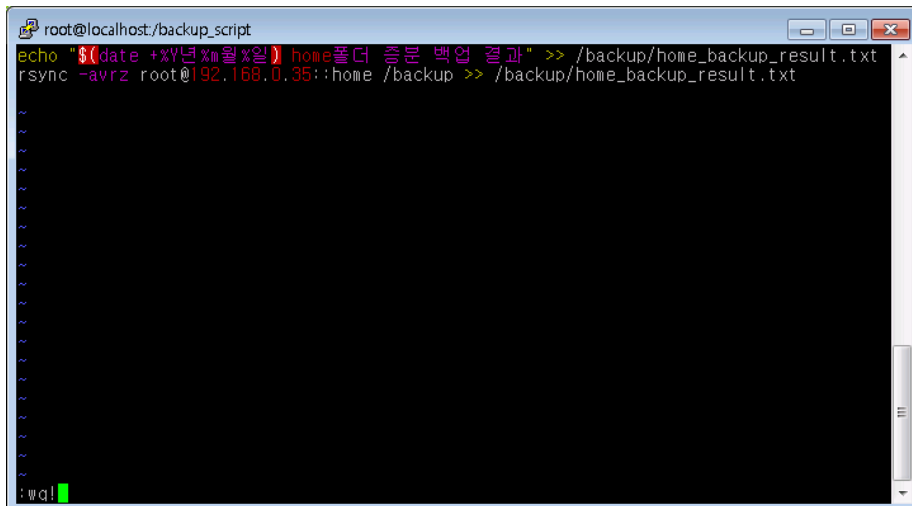
5 vim 편집기 명령어를 이용해 backup.sh 파일을 수정하기 위해 아래 명령어를 실행합니다.
 ➔ `vim /backup_script/backup.sh`



6 백업 서버의 편집기에서 “i”를 눌러 입력모드로 전환하고 아래 예시대로 백업스크립트를 작성 후 “:wq!”를 입력하여 저장합니다.

Ex) 본 서버의 /home 폴더 데이터를 백업 서버의 /backup으로 백업 후 로그를 남기는 스크립트 예시.

- ➔ echo "\$(date +%Y년%m월%일) home폴더 증분 백업 결과" >> /backup/home_backup_result.txt
- ➔ rsync -avrz root@192.168.0.35::home /backup >> /backup/home_backup_result.txt
- ➔ :wq!



7 해당 스크립트 파일에 실행권한을 부여합니다.

- ➔ chmod 700 /backup_script/backup.sh

```

root@localhost:/backup
drwxr-xr-x  2 root root 12288 Feb 16 03:06 sbin
drwxr-xr-x  2 root root  4096 Feb 16 01:35 selinux
drwxr-xr-x  2 root root  4096 May 11  2011 srv
drwxr-xr-x 11 root root    0 Feb 16  2012 sys
drwxrwxrwt  4 root root  4096 Feb 16 04:05 tmp
drwxr-xr-x 14 root root  4096 Feb 16 01:37 usr
drwxr-xr-x 20 root root  4096 Feb 16 01:38 var
[root@localhost /]# cd backup
[root@localhost backup]# ll
total 4
drwxr-xr-x 5 root root 4096 Feb 16 06:58 home
[root@localhost backup]# rm -rf home
[root@localhost backup]# ll
total 0
[root@localhost backup]# vim /backup_script/backup.sh
[root@localhost backup]# vim /etc/sysconfig/i18n
[root@localhost backup]# source /etc/sysconfig/i18n
[root@localhost backup]# vim /backup_script/backup.sh
[root@localhost backup]#
[root@localhost backup]# chmod 700 /backup_script/backup.sh
[root@localhost backup]#

```

- 8 해당 스크립트를 실행하고 데이터 백업이 잘 되었는지 로그를 확인합니다.
- ➔ /backup_script/backup.sh
 - ➔ cat /backup/home_backup_result.txt

```

root@localhost:/
[root@localhost /]# /backup_script/backup.sh
[root@localhost /]# cat /backup/home_backup_result.txt
2012년02월16일 home폴더 증분 백업 결과
sending incremental file list
home/
home/test1/
home/test1/testfile1
home/test1/testfile2
home/test1/testfile3
home/test2/
home/test2/testfile1
home/test2/testfile2
home/test2/testfile3
home/test3/
home/test3/testfile1
home/test3/testfile2
home/test3/testfile3

sent 544 bytes  received 199 bytes  1486.00 bytes/sec
total size is 0  speedup is 0.00
[root@localhost /]#

```

- 9 /home.폴더에 있던 데이터가 /backup 폴더에 실제로 존재하는지 확인합니다. 백업 스크립트 작성이 완료되었으므로 챕터 3을 참고하여 crontab에 등록하는 작업을 진행합니다.
- ➔ ls -R /backup

```
root@localhost/backup
[root@localhost backup]# ls -R /backup
/backup:
home  home_backup_result.txt

/backup/home:
test1 test2 test3

/backup/home/test1:
testfile1 testfile2 testfile3

/backup/home/test2:
testfile1 testfile2 testfile3

/backup/home/test3:
testfile1 testfile2 testfile3
[root@localhost backup]#
```

감사합니다.