Author – *A.Kishore/Sachin*
http://appsdba.info

Rsync – An Intro

Rsync is a fast and extraordinarily versatile file-copying tool. It can copy locally, to/from another host over any remote shell, or to/from a remote rsync daemon. It provides a large number of options that control every aspect of its behavior and permit very flexible specification of the set of files to be copied. It is famous for its delta-transfer algorithm, which reduces the amount of data sent over the network by sending only the differences between the source files and the existing files in the destination. Rsync is widely used for backups and mirroring and as an improved copy command for everyday use.

Rsync finds files that need to be transferred using a lqquick checkrq algorithm (by default) that looks for files that have changed in size or in last-modified time. Any changes in the other preserved attributes (as requested by options) are made on the destination file directly when the quick check indicates that the file's data does not need to be updated.

The main difference between scp and rsync is, scp does not provide an option to "Not-Overwrite" a file if already exists on the destination server.

This can be used mainly when we want to take incremental backup of filesystems or keep servers in sync etc.

- **Syntax Of rsync :**

rsync [OPTION...] [USER@]HOST:SRC... [DEST]

Rsync Configuration:

Step 1 : Establish SSH with No Password.

On Source server generate ssh keys as below:
Lets take an Example:

```
oracle@source $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/export/oracle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /export/oracle/.ssh/id_rsa.
Your public key has been saved in /export/oracle/.ssh/id_rsa.pub.
```

Step 2 : Copy the id_rsa.pub file to the destination server.

Step 3: Append the contents of the id_rsa.pub to authorized_keys (or authorized_keys2-depending on version) file on target server. This file could be located under .ssh directory in the home directory of the user on target server.
For example:
cat id_rsa.pub >>/home/bkpuser/.ssh/authorized_keys

Please Note:
Please check the permissions of the auhorized_keys file and home directory of the users on both systems, public read/write permissions will fail this setup. We can have rwx- - permissions for this file or home directory.

- **How rsync works:**
  In our demonstration, we are taking the following assumption:

Source Server	: source
Source User	: oracle
Destination Server	: destination
Destination User	: user

  - We will create a directory "test" in source and synchronize with destination server.

```
oracle@source MYDB $ mkdir test
oracle@source MYDB $ pwd
/backup/oracle/MYDB/SKS
oracle@source  MYDB $ rsync -av /backup/oracle/MYDB/SKS/test user@destination:/backup/oracle/SKS
building file list ... done
test/
```

    Here is what the "-av" option does:
    a = archive - means it preserves permissions (owners, groups), times, symbolic links, and devices.
    v = verbose - means that it prints on the screen what is being copied.

  - Now we will create one file "test.txt" in test directory and synchronize with destination server.

```
oracle@source MYDB $ cd test
oracle@source MYDB $ touch test.txt
oracle@source MYDB $ rsync -av /backup/oracle/MYDB/SKS/test user@destination:/backup/oracle/SKS
```

```
        building file list ... done
        test/
        test/test.txt  <——————– Only one new file copied to destination.
```

- <u>We will create two files“test2.txt”,“test3.txt” in test directory and synchronize with destination server.</u>

```
        oracle@source MYDB $ touch test2.txt
        oracle@source MYDB $ touch test3.txt
oracle@source  MYDB $ rsync -av /backup/oracle/MYDB/SKS/test user@destination:/backup/oracle/SKS
        building file list ... done
        test/
        test/test2.txt  <——————–Only two new files copied to destination.
        test/test3.txt
```

- <u>Check the permission.</u>

```
        oracle@source MYDB $ ls -otr
        total 0
        -rw-r–r– 1 oracle 0 May 3 11:19 test.txt
        -rw-r–r– 1 oracle 0 May 3 11:20 test2.txt
        -rw-r–r– 1 oracle 0 May 3 11:20 test3.txt
```

- <u>Now we will update one file and try to synchronize with destination server:</u>

```
        oracle@source  MYDB $ echo "Hello Sir" >>test.txt
        oracle@source  MYDB $ echo "Hello Sir" >>test.txt
oracle@source  MYDB $ rsync -av /backup/oracle/MYDB/SKS/test user@destination:/backup/oracle/SKS
        building file list ... done
        test/test.txt  <—————–Only One file copied to destination.
```

<u>Additional features of rsync :</u>

- Support for copying links, devices, owners, groups, and permissions.
- Exclude and exclude-from options similar to GNU tar.
- A CVS exclude mode for ignoring the same files that CVS would ignore.
- Can use any transparent remote shell, including ssh or rsh.
- Does not require super-user privileges.
- Pipelining of file transfers to minimize latency costs.
- Support for anonymous or authenticated rsync daemons (ideal for mirroring).