Tutorial Rsync

Ubuntu already has Rsync installed so we don't have to download it, if you want to check the version of Rsync that you're running use rsync --version.

Step 1: On the backup server generate RSA keys with the following command.

```
ssh-keygen -t rsa
```

We use the default directory to store the keys.

```
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
```

It's up to you if you want a passphrase, it's of course more secure to have one.

```
Enter passphrase (empty for no passphrase):
```

Now you have your RSA keys, the next step is to copy them to the clients so you can safely SSH and It's also required to remotely backup without using passwords.

Step 2: Copying RSA keys to the clients

Next we use the ssh-copy-id command followed with 'username'@'ipaddress'.

```
sync@portahost:~/backups/zabbix$ ssh-copy-id agent@192.168.1.6
```

```
rsync@portahost:~/backups/zabbix$ ssh-copy-id agent@192.168.1.6
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/rsync/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.6 (192.168.1.6)' can't be established.
ECDSA key fingerprint is SHA256:bwTdTMxW3UWYyXNUH1kF8QVI6A6TGr7ubvQeupxvaA0.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
agent@192.168.1.6's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'agent@192.168.1.6'"
and check to make sure that only the key(s) you wanted were added.
```

Step 3: Changing the ssh config file.

After that is done we will change the config file so we won't have to enter the password if the client has our RSA key. This is required so we get automatic backups from a remote client to the server.

Follow the steps:

```
$ sudo nano /etc/ssh/sshd_config
```

Within that file, find the line that includes `PermitRootLogin` and modify it to ensure that users can only connect with their SSH key:

/etc/ssh/sshd_config

```
PermitRootLogin without-password
```

Save and close the file when you are finished.

To put these changes into effect:

```
$ sudo systemctl reload sshd.service
```

Step 4: Testing the Rsync commands

First we run a command like this:

```
rsync -av --delete -e ssh puppetmain@192.168.1.3:/etc/puppet ~/backups
```

I will explain what the following options stand for.

-a = recursive
-v = verbose
--delete : this causes everything that isnt in the etc/puppet directory but is in the backups directory to be deleted.

The 'servername'@'ipaddress':/etc/puppet 'pulls' the content of the puppet folder on the client server and stores it in the backup folder on our server.

If this works and you see the content in the /backups folder then you know it works.

Step 5: Crontab

We use Cron to schedule jobs to run periodically at fixed times, dates, or intervals.

To open crontab we enter crontab -e in the command line.

Then we have to choose between different editors.

```
client@client-virtual-machine:~$ crontab -e
no crontab for client - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /bin/ed

Choose 1-3 [1]:
```

I recommend nano because it's the easiest to use.

Then you copy the rsync command in the cron file and add 5 asterisk.

The asterisks means that this command will be executed every minute, this is just for testing purposes. Later we will move on to a longer cycle.

```
* * * * * rsync -av --delete -e ssh puppetmain@192.168.1.3:/etc/puppet ~/backups/puppet
```

Wait a minute and check in your repository on your server to see if the files are there. If they aren't there check if the syntax is correct, try the command without crontab.

If this is successful the only thing remaining is setting crontab to backup in longer cycles. I will set it so it backups every day at 4pm.

```
0 16 * * * rsync -av --delete -e ssh puppetmain@192.168.1.3:/etc/puppet ~/backups/puppet
```

This is how it should look like.

Step 6(optional): Backing up MySQL databases

If you want to back up your database you can do it either manually or with software.

We will use the software because It's much simpler and keeps daily/weekly/monthly backups without having to do any changes to the configuration at all.

automysqlbackup is a utility program that is available in the Ubuntu repositories. This utility can be scheduled to automatically perform backups at regular intervals. To install, type the following into the terminal:

```
$ sudo apt-get install automysqlbackup
```

A following prompt will ask you which mail configuration you prefer. Select "internet site" if you're going to set up email notification. If not, just select "no configuration". I personally went for "no configuration" to leave things unchanged. Start *automysqlbackup*:

```
$ sudo automysqlbackup
```

A folder structure and initial backups will be performed immediately. The default location for backups is `/var/lib/automysqlbackup`. List contents of this directory to see the folder structure:

```
$ ls /var/lib/automysqlbackup
```

You should see three directories for the `daily`, `weekly` and `monthly` backups being performed automatically by *automysqlbackup* and a subdirectory for each database containing the respective gzipped SQL dump.

If you want to edit the configuration file you can do so with following command.

```
sudo nano /etc/default/automysqlbackup
```

Now that we got automysqlbackup installed we can add an crontab rsync command for the folder that automysqlbackup backups into.

```
0 16 * * * rsync -av --delete -e ssh zabbix@192.168.1.5:/var/lib/automysqlbackup ~/backups/zabbix
```

And we're done!

That's how you make automatic remote backups using rsync, crontab and automysqlbackup.

Links:

These links were used to make this tutorial.

https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2

https://www.howtogeek.com/135533/how-to-use-rsync-to-backup-your-data-on-linux/

https://medium.com/@mhagemann/how-to-backup-mysql-databases-automatically-on-ubuntu-17-10-a2b29fb47ac9