Author: Martin von Böhlen, ©2022

Save & Restore With bash

Disclaimer

The method demonstrated within this document is suitable to save a bunch of files to a remote and restore them afterwards. It is **not** suitable to backup & restore (hence the title) an exact bit copy of files. Use *rsync* for that purpose . It is also **not** suitable to backup a whole disk, or media, or some such. Use *dd* for that purpose.

Intro

Often you have to save some files to a remote computer. To be able to restore them sometimes later. To be able to do so you must be able to:

1. login to remote computer.
2. bundle & compress files
3. copy files to destination computer which may also be remote
4. copy files back to origin
5. extract file to former location

The traditional way to do so is quite comfortable under UN*X/Linux and uses four programs:

1. **ssh** (secure shell) to login to remotes
2. **scp** (secure copy) to copy to/from remotes
3. **tar** (tape archiver) to bundle multiple files to one large archive and extract them from such archive
4. **gzip** (GNU zip) to compress and decompress files very, very efficiently

Annotated example

Let us assume you have a directory */etc/foobar* and a file *~/foobar.data.* Since the computer they are on will be re-installed from scratch you would like to save both and restore them after install.

```
ssh user@mambo.jambo.local
sudo -s
tar -cf - /etc/foobar ~/foobar.data | gzip -c > ~/foobarwdata.tar.gz
scp ~/foobarwdata.tar.gz olduser@host.we.from:/home/olduser
```

```
#… back on our old host, time is fleeting …
```

```
scp ~/foobarwdata.tar.gz user@mambo.jambo.local:/home/user
ssh user@mambo.jambo.local
sudo -s
cd /
zcat ~/foobarwdata.tar.gz | tar -xvf -
```

1. **Login into remote**
   ssh user@192.168.0.200
   This logs you in as user ‘user’ on host with IP-address 192.168.0.200
   or
   ssh user@mambo.jambo.local
   This logs you in as user ‘user’ on host with name *mambo.jambo.local*
   To log out again press CTRL-d on an **empty** line. In this case you might want to use CTRL-c to abort whatever is running in foreground.
2. **Maybe become super user**
   In case the files in question are not owned by you you might have to become *root.* Do that by using either *sudo -s* or *su -*. Further explanations are beyond scope. If you need more help go looking for some nice tutorial like https://www.howtoforge.com/tutorial/sudo-beginners-guide/ !

3. **Bundle and compress files**
   One uses *tar -cf <file_name_of_bundle> <files_to_take>* to create new bundles (archives is the correct technical term). They are then labelled with the *.tar* extension.
   One uses *gzip -c <file_name>* to compress files to *stdout*. They are then labelled with the *.gz* extension. The file '-' (minus) is a special file. It means *stdout*.
   We weld programs together by using pipes '|' (vertical line). Thereby putting *stdout* of left file into *stdin* of right file.
   Now producing a compressed archive from a directory */etc/foobar* and a file *~/foobar.data* and storing it into your $HOME would look like this (underline denotes a blank!):
   tar_-cf_–_/etc/foobar_~/foobar.data_|_gzip_-c_>_~/foobarwdata.tar.gz

4. **Copy to save host**
   Copying to a save host works by using *scp <source> <target>*. Both can be designations for remote locations or local directory entires. To save our archive we would copy it to the host we came from by using:
   scp_*~/foobarwdata.tar.gz_olduser@host.we.from:/home/olduser*
   where *olduser* is the user name of our account on *host.we.from*.

5. **Copy back to original host**
   After many seconds the danger has passed. We now copy our archive back to its origins. By using scp from out save host
   scp_*~/foobarwdata.tar.gz_user@mambo.jambo.local:/home/user.*
   And following right behind our archive with:
   ssh_user@mambo.jambo.local

6. **Changing directory**
   Note that *tar* has removed the leading '/' from all pathes in our archive. So we have to do a *cd /* to restore our files. Afterwards we do a:
   zcat_~/foobarwdata.tar.gz | *tar -xvf -*.

7. **Done**
   All our files are restored. You may remove the compressed archive and leave via CTRL-d.