

SVN Server

- Export SVN repository
- Migrate configuration after export

Export SVN repository

At first it seems quite forward to do this, but I have ran into the issue of encoding problems. And trying to get a huge (40G) dumpfile into the container.

What did work for me:

1. Have a bind path in your container, to manage the large file

^ Storage		
Volume / Host Path	Container Path	Permission
/Container/DockerMedia/DockerStorage/Sophior.SVN	/var/svnbackup	Writable

2. Use the container and dump the file into this path

```
svnrndump dump -r0:HEAD [URL_TO_YOUR_REPO] > Your_dumpfile.dmp
```

This will show the progress as follows:

```
* Dumped revision 882.
* Dumped revision 883.
* Dumped revision 884.
* Dumped revision 885.
* Dumped revision 886.
* Dumped revision 887.
* Dumped revision 888.
* Dumped revision 889.
* Dumped revision 890.
* Dumped revision 891.
* Dumped revision 892.
* Dumped revision 893.
* Dumped revision 894.
* Dumped revision 895.
* Dumped revision 896.
* Dumped revision 897.
* Dumped revision 898.
* Dumped revision 899.
* Dumped revision 900.
* Dumped revision 901.
* Dumped revision 902.
* Dumped revision 903.
* Dumped revision 904.
```

```
svnadmin load /path/yourrepo < /path/your/dum.dmp
```

```
* Dumped revision 2372.  
* Dumped revision 2373.  
* Dumped revision 2374.  
* Dumped revision 2375.  
/var/svnbackup # ls  
Sophior.dmp          Sophior_svn_20241021.dmp  
/var/svnbackup # svnadmin load /home/Sophior < /var/svnbackup/Sophior.dmp  
<<< Started new transaction, based on original revision 1  
* editing path : sophior ... done.
```

What didn't work for me:

This is the official documentation : [dump your SVN](#)

```
svnadmin dump /path/to/your/old/repo > backup.dump  
svnadmin load /path/to/your/new/repo < backup.dump.dmp
```

If you redirect the input to your host, it circumvents the large file in the docker.

```
cd /tmp  
docker exec -it edf77623c9207 svnadmin dump /home/svn/YourRepository > YourRepository_svn.dmp
```

But importing it back, an error is produced:

```
' into a number04: Could not convert ' 2
```

Migrate configuration after export

After you have the repository in a dmp file, you don't have your configuration yet.

In my case with the docker, I can see which configuration files are required:

```
docker run -d --name svn-server -p 80:80 -p 3690:3690 -v <hostpath>:/home/svn -v svn_config:/etc/subversion -v svnadmin_config:/opt/svnadmin/data elleflorio/svn-server
```

In reality, I need to copy the following files over (and use my mapped drive to transfer these files before restarting the container)

```
/opt/svnadmin/data # cp /var/svnbackup/svnadmin/config.ini config.ini
/opt/svnadmin/data # cp /var/svnbackup/svnadmin/config.ini config.tpl.ini
/opt/svnadmin/data # cp /var/svnbackup/svnadmin/config.tpl.ini config.tpl.ini
/opt/svnadmin/data # cp /var/svnbackup/svnadmin/userroleassignments.ini userroleassignments.ini

/var/svnbackup # cp ./subversion/passwd /etc/subversion/passwd
/var/svnbackup # cp ./subversion/subversion-access-control /etc/subversion/subve

/home/svn/YourRepo/conf # cp /var/svnbackup/conf/hooks-env.tmpl hooks-env.tmpl
/home/svn/YourRepo/conf # cp /var/svnbackup/conf/passwd passwd
/home/svn/YourRepo/conf # cp /var/svnbackup/conf/authz authz
/home/svn/YourRepo/conf # cp /var/svnbackup/conf/svnserve.conf svnserve.conf
```

In windows, to switch the clients, you can use your hosts file. So your existing projects just keep working as they always have. (provided you've used the same URL before with another IP)

```
hosts
File Edit View

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#       ::1              localhost

192.1.2.3 svn.yourserver.com
```

> This PC > Local Disk (C:) > Windows > System32 > drivers > etc

w folder