

FSPortType : Exports as More than Local Disk Proxies

Mark Morgan

Vanamala Venkataswamy

Date: 02/09/2012

Thinking of an export as a means of proxying a local directory structure was unnecessarily restrictive. In fact, an export can serve as a proxy to any hierarchically arranged file system. Further, nearly all data staging protocols can be used to expose such information. Instead of using exports to expose local directory structures on a machine, a new type of resource fork based export (which henceforth referred to as a FSPortType) that exposes the structure of a data staging protocol, thus unifying the concepts. Such a FSPortType could then be created that would expose file systems through protocols such http, CIFS/SMB and NFS.

The benefits to such unification would be fairly significant. First of all, we would suddenly gain access to a lot of additional types of data (versioned, remote, local, and embedded in other file structures like tar, jar, and zip). This would be true not only for the grid namespace, but also for the compute job data staging. Additionally, doing so will provide significant alternatives to that of having users run their own grid containers. Users could instead simply create an export on a remote machine giving that export any supported protocol that they already had on their machine (particularly things like CIFS and NFS – we could even create a new, lightweight file protocol if we needed to). Finally, this solution would solve the secure data transfer problem quite nicely. Instead of providing account information for an SCP in a job description document, the user could instead create an export with the appropriate account information once on a trusted machine and through encrypted and protected channels and then stage data for jobs using the much more secure grid data staging protocol.

Another benefit to modeling exports this way is that we simplify one of the greater export confusions that has been dogging us for a while now. It's very confusing to people that they are specifying a local path on a remote machine (in the case where the export isn't on their local machine) and it has made GUI'ifying the experience difficult. However, while the new FSPortType would still allow such operations, the general use case would no longer require that concept. Instead, the much easier concept of simply proxying a remote file protocol such as FTP or CIFS would be presented and coming up with GUIs to aid in this description would be significantly easier and more intuitive.

Filesystem export

1. Select a RNS Path on which the user has write permissions Ex. /home/vana.
2. Right click on the RNS, Select 'Create Filesystem Proxy'. Select 'Filesystem' tab.

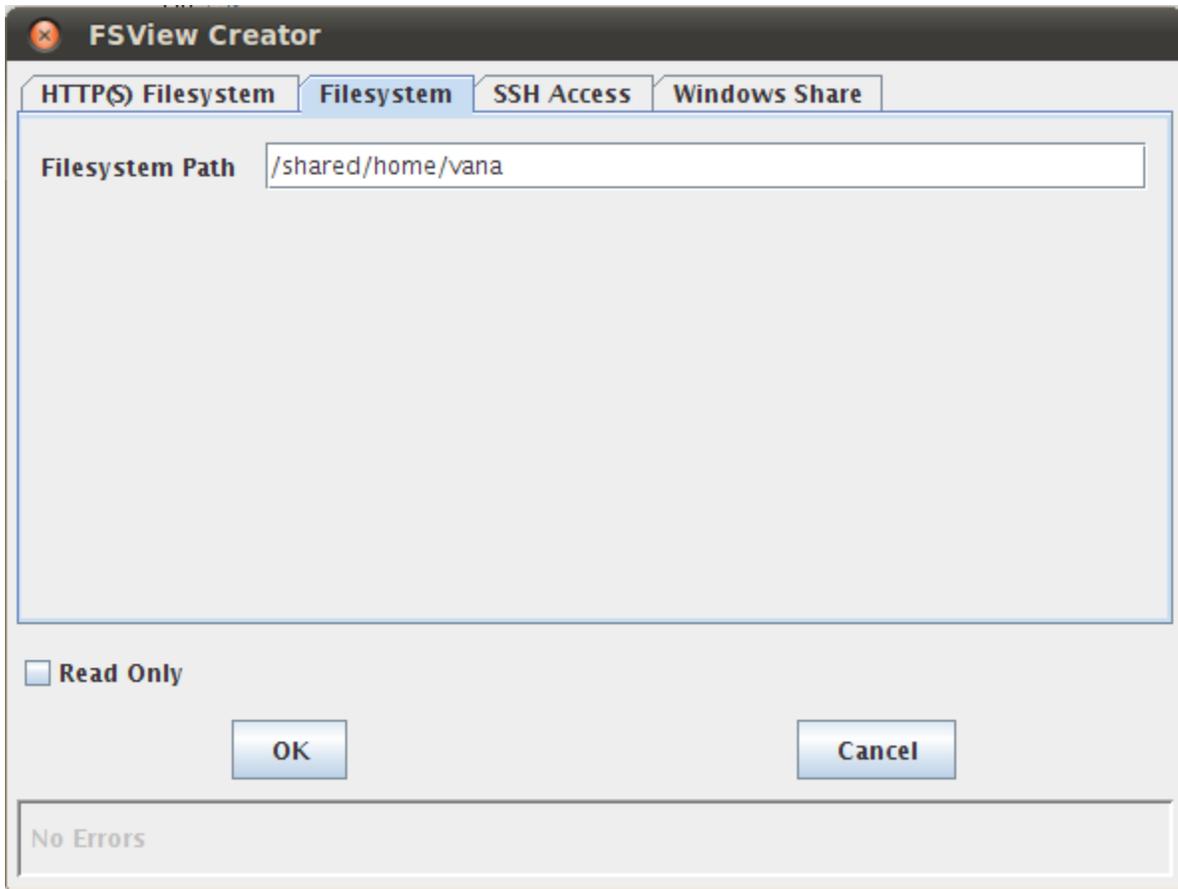


Figure 1. Filesystem export

3. Enter file system path Ex. /shared/home/vana (this is a network shared file system path, it should be visible to the machine on which container is running. Ex. BootstrapContainer). Refer Figure 1.

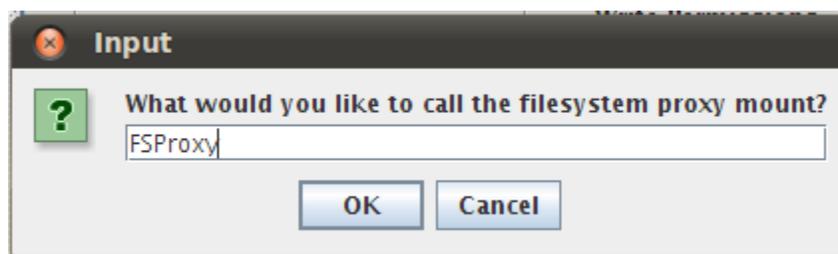


Figure 2. Name of the Filesystem proxy name

4. Provide file system proxy mount name Ex. FSProxy. Click on OK. Refer Figure 2.

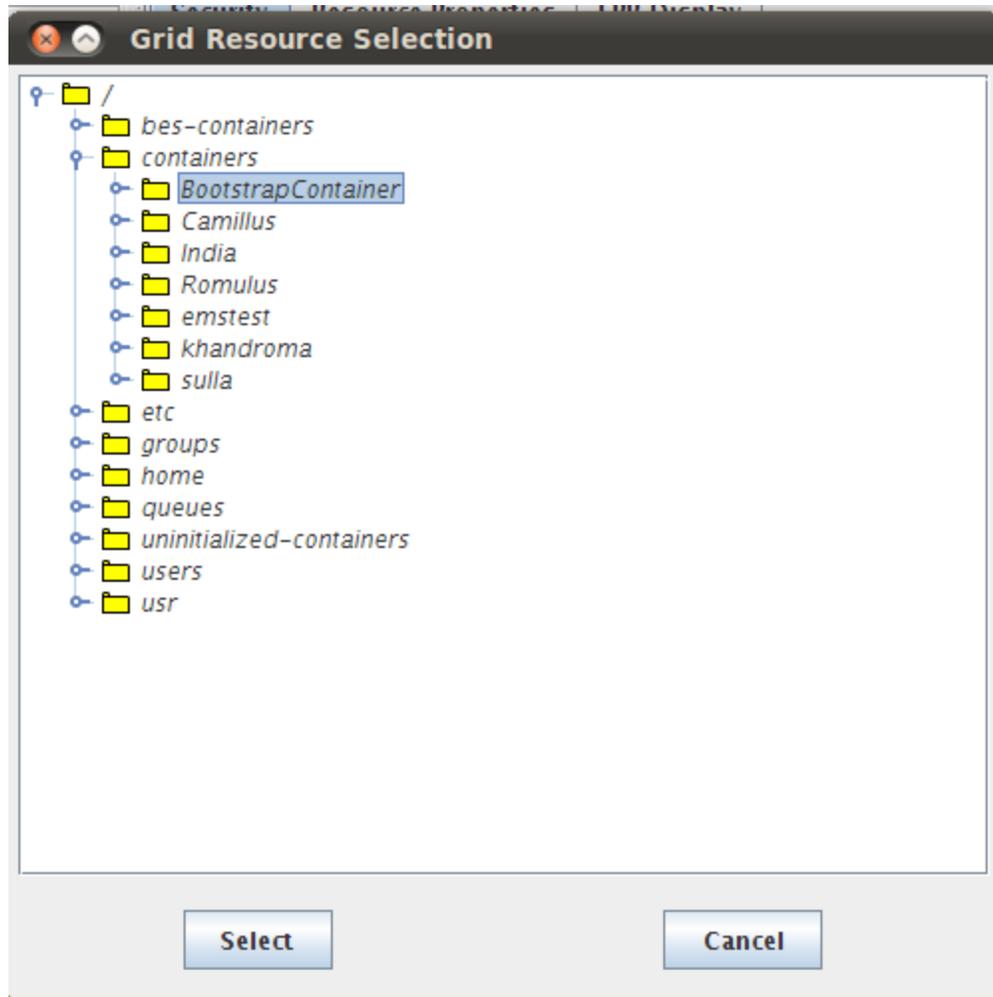


Figure 3. Container that will act as proxy to the shared file system

5. In Grid Resource select, pick the container you want to use a proxy to file system you are sharing (Ex. /containers/BootstrapContainer). Click on Select. Refer Figure 3.

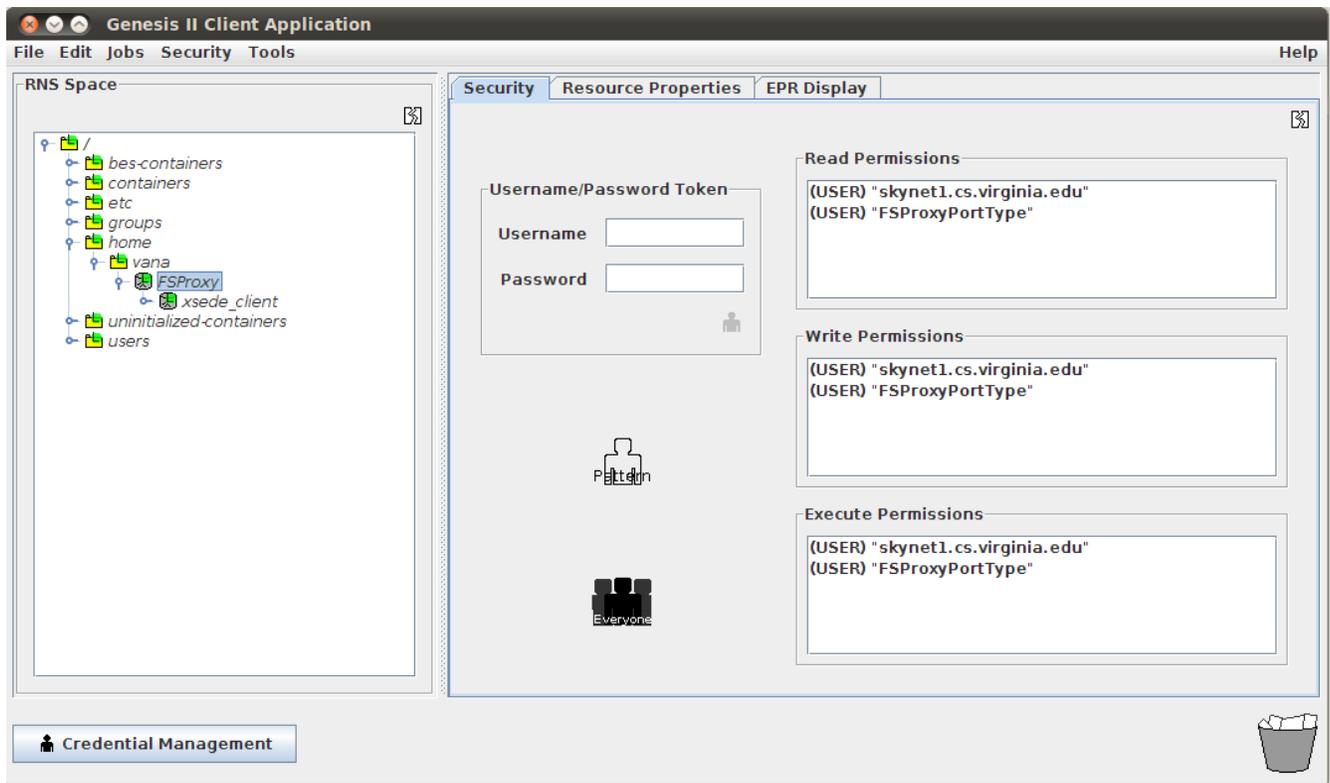


Figure 4. Listing the shared file system from grid name space.

6. This will export all the files and directories in /shared/home/vana from the machine using BootstrapContainer as proxy. All the operations on the FSProxy export will be handled by BootstrapContainer. Refer Figure 4.

SSH Access

1. Select an RNS Path Ex. /home/vana and right click to select 'Create Filesystem Proxy'. Select the SSH Access tab. Refer Figure 5.

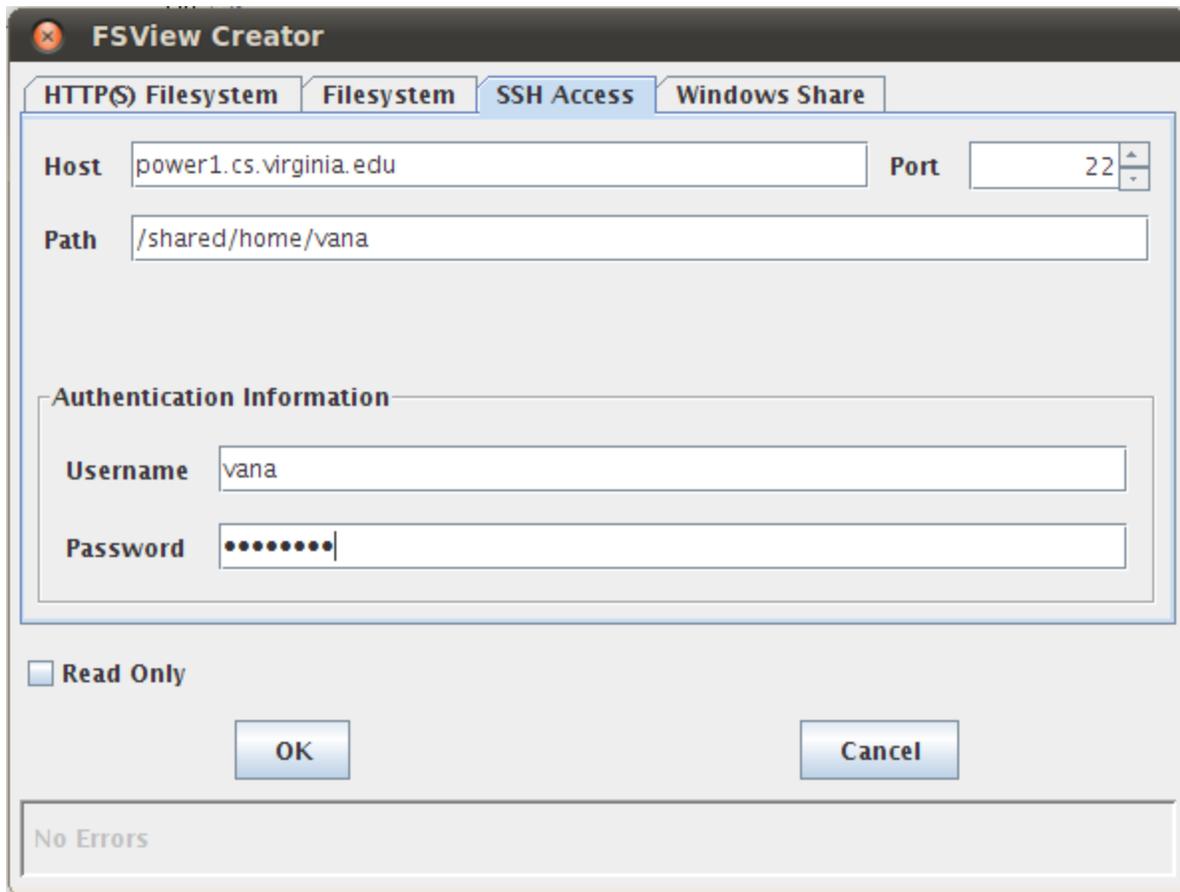


Figure 5. SSH access to shared file system path on power1.cs.virginia.edu

2. Enter host name (Ex. power1.cs.virginia.edu) on which you want to provide the path to be shared /shared/home/vana. This host machine should be SSH accessible to the machine on which container is running. Enter username and password (same username and password used to login to machine). Click on OK. Refer Figure 5.
3. Provide file system proxy mount name Ex. SSHProxy. Click on OK. Refer Figure 6.

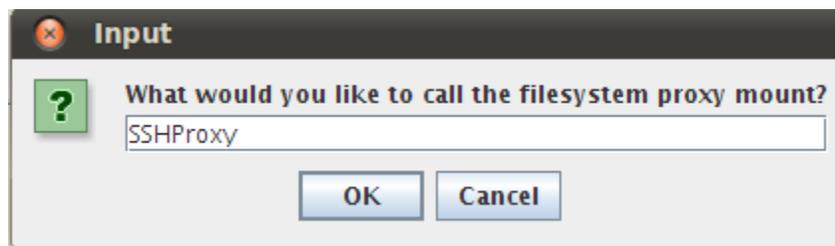


Figure 6. Name of the ssh proxy access

4. In Grid Resource select a container that can ssh to the host machine (Ex.

/containers/BootstrapContainer). Click on Select. Refer Figure 7.

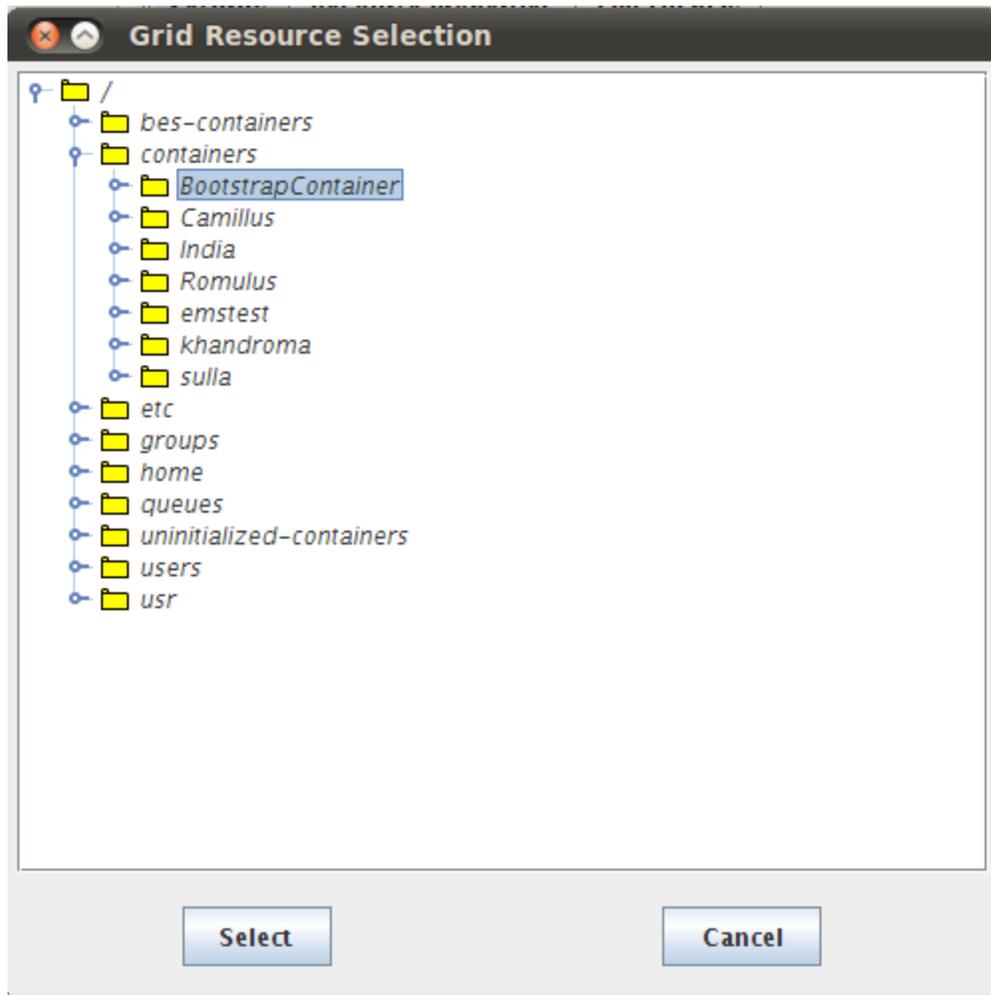


Figure 7. Container that will act as SSH proxy for shared file system on power1

5. This will export all the files and directories in /shared/home/vana from the machine on which BootstrapContainer is running on. You can list the SSHProxy to access files/directories or to add/deleted files and directories. Refer Figure 8.

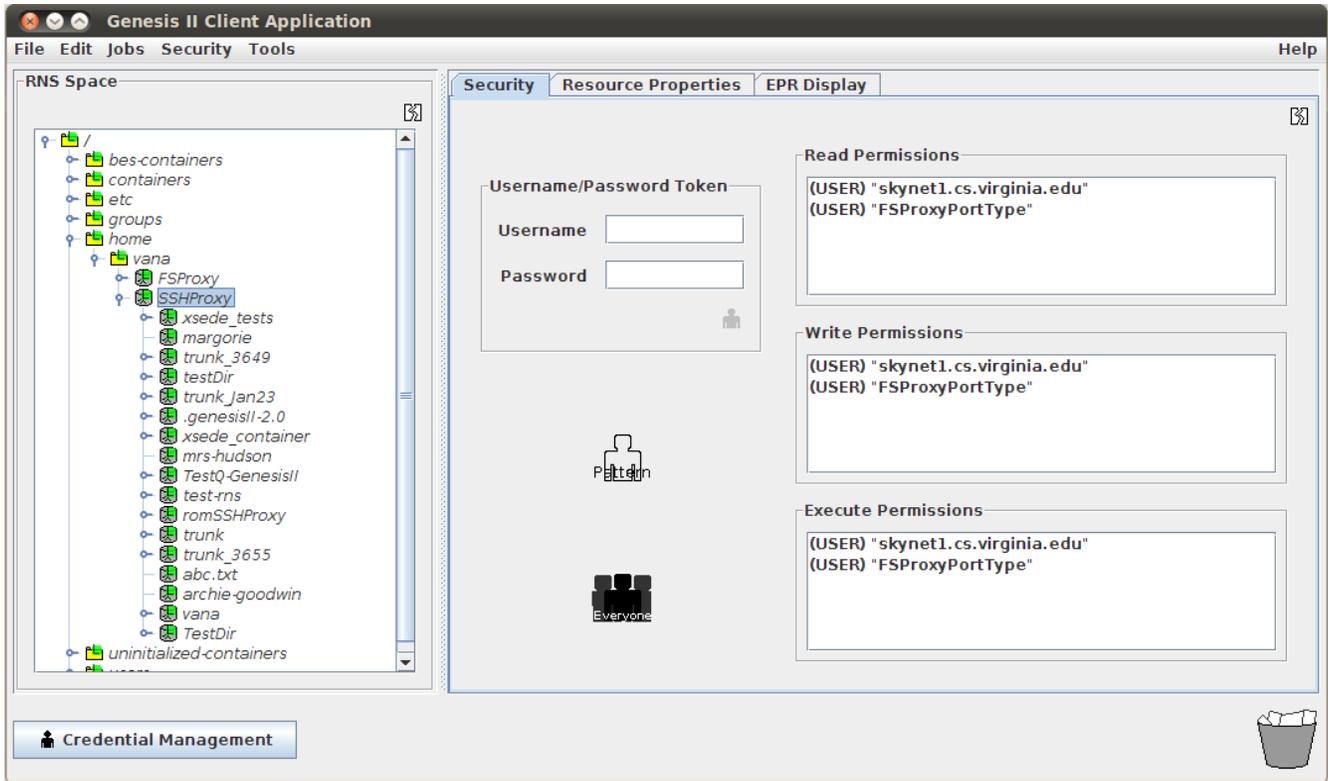


Figure 8. Listing the contents of the SSH shared file system from power1.cs.virginia.edu

Windows Share

1. Select an RNS Path Ex. /home/vana. Right click "Create Filesystem Proxy". Select 'Windows Share' tab. Refer Figure 9.

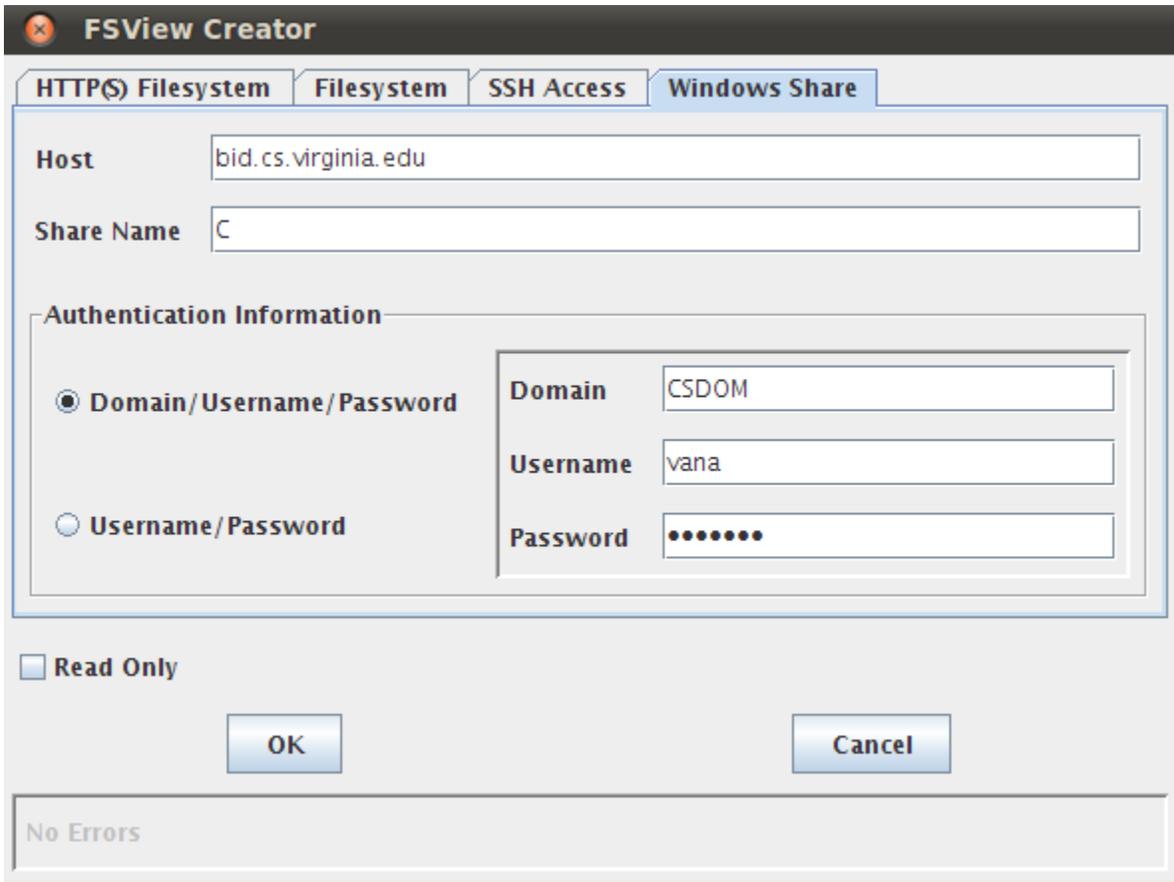


Figure 9. Windows share

2. Enter Windows host name on which you have a network shared drive/path (Ex. bid.cs.virginia.edu). Enter username and password (username and password can be set on the windows share/drive by its owner). Click on OK. Refer Figure 9.
3. Provide file system proxy mount name Ex. WindowsShare. Click on OK. Refer Figure 10.

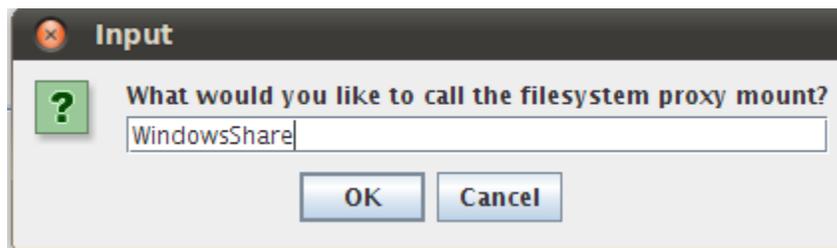


Figure 10. Windows Proxy share name

4. In Grid Resource select a container that can access windows share on bid.cs.virginia.edu (Ex. /containers/BootstrapContainer). Refer Figure 11.

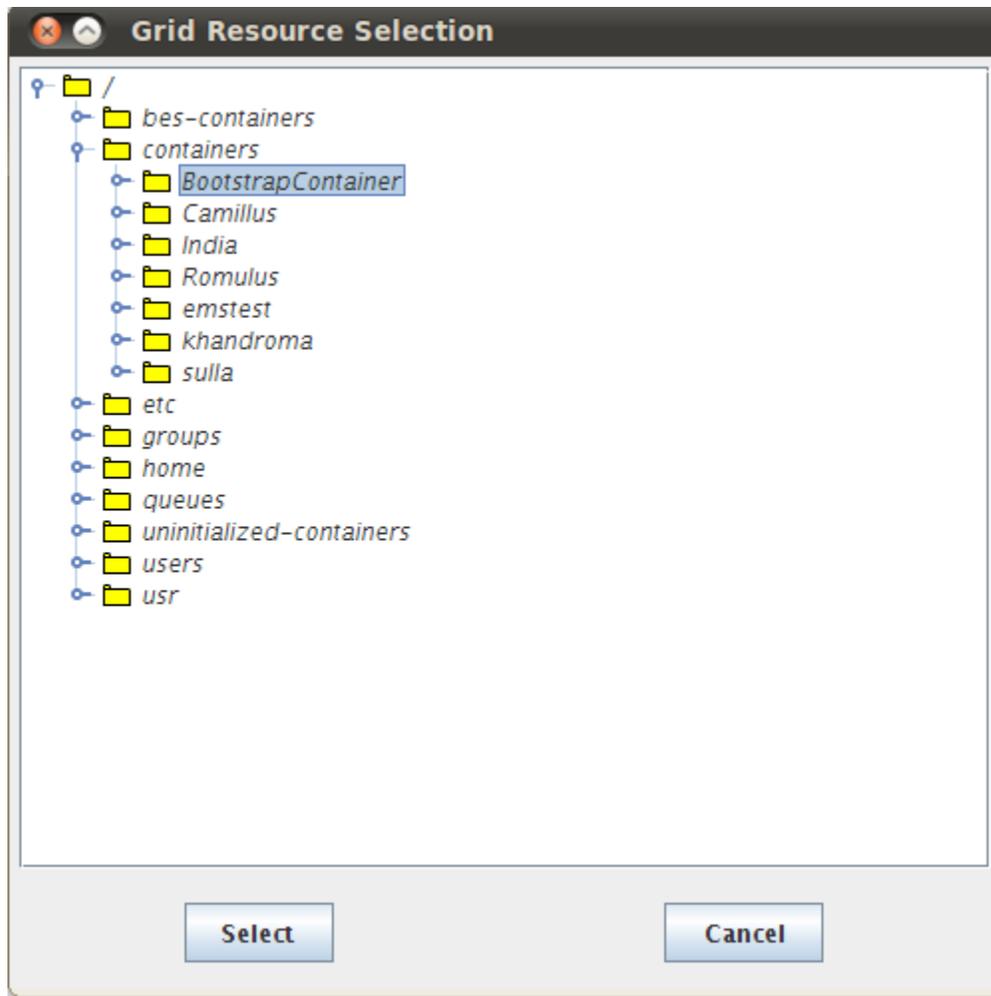


Figure 11. Container that will act as proxy to Windows share.

5. This will export all the files and directories in C: drive from the machine `bid.cs.virginia.edu`. You can list the `WindowsShare` to access windows files/directories. Refer Figure 12.

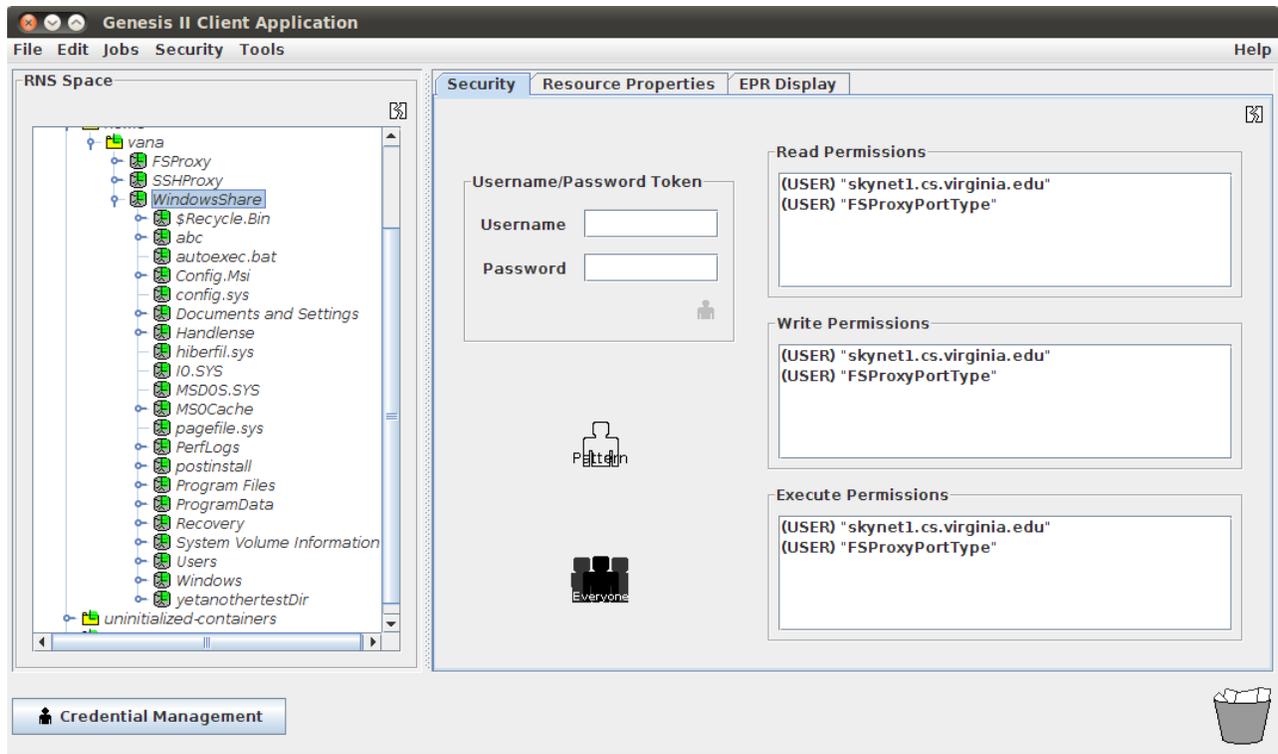


Figure 12. Listing windows C drive from grid name space.