

Docker volumes provide persistent storage to containers, and they can be shared between containers to allow them to interact with the same data. However, sharing volumes creates additional challenges in the context of a Swarm cluster, where containers may be running on different nodes. In this lesson, we will discuss how you can create shared volumes that work across multiple swarm nodes using the `vieux/sshfs` volume driver.

Relevant Documentation

- <https://docs.docker.com/storage/volumes/#share-data-among-machines>

Lesson Reference

1. Install the `vieux/sshfs` plugin on all nodes in the swarm.

```
docker plugin install --grant-all-permissions vieux/sshfs
```

2. Set up an additional server to use for storage. You can use the `Ubuntu 18.04 Bionic Beaver LTS` image with a size of `Small`. On this new storage server, create a directory with a file that can be used for testing.

```
mkdir /home/cloud_user/external  
echo External storage! > /home/cloud_user/external/message.txt
```

3. On the swarm manager, manually create a Docker volume that uses the external storage server for storage. Be sure to replace the text `<STORAGE_SERVER_PRIVATE_IP>` and `<PASSWORD>` with actual values.

```
docker volume create --driver vieux/sshfs \  
-o sshcmd=cloud_user@<STORAGE_SERVER_PRIVATE_IP>:/home/cloud_user/external \  
-o password=<PASSWORD> \  
sshvolume  
  
docker volume ls
```

4. Create a service that automatically manages the shared volume, creating the volume on swarm nodes as needed. Be sure to replace the text `<STORAGE_SERVER_PRIVATE_IP>` and `<PASSWORD>` with actual values.

```
docker service create \  
--replicas=3 \  
--name storage-service \  
--mount volume-driver=vieux/sshfs,source=cluster-volume,destination=/app,volume-opt=sshcmd=cloud_user@<
```

5. Check the service logs to verify that the service is reading the test data from the external storage server.

```
docker service logs storage-service
```