ONTAP_Snapmirror_Strict_Sync_Access_fencing

In this lab we will setup a StrictSync relationship and test the fencing of the source volume when the destination is offline.

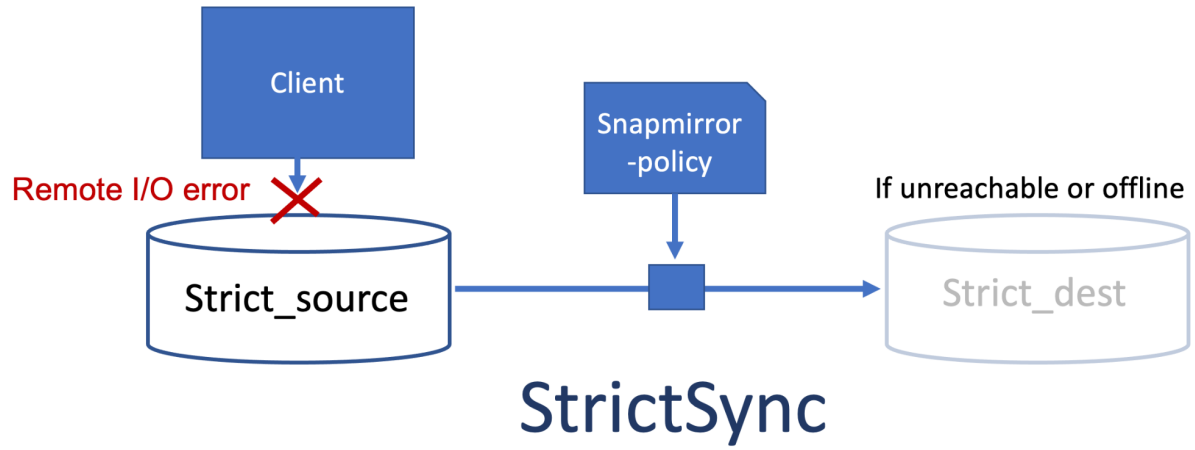
Prerequisites:

. Two clusters are peered **(ONTAP_Cluster_Peering)**
. Two SVMs are peered **(ONTAP_SVM_Peering)**
. Export policy to export that allows writing to an NFS client **(ONTAP_NFS_SVM)**
. NFS client that can access the source volume

This is what we will do:

1. Create a source volume for the strictsync relationship and mount it
2. Create a destination volume for the strictsync relationship of the type dp
3. Create a snapmirror relationship with the stictsync policy and initialize the relationship
4. Create a file in the volume from the linux nfs client
5. To test the fencing mechanisme we offline the volume
6. Run the volume show command and filter on these two fields:
   “is-protocol-access-fenced”
   “protocol-access-fenced-by”
7. On the linux client try to access the volume. This will fail
8. Check the relationship on cluster2
9. Online the volume on the cluster2 again
10. Check the relationship on cluster2 once more. It should have repaired the relationship

(see next page for commands)



Commands

1. Create a source volume for the strictsync relationship and mount it
cluster1::>
vol create -vserver c1_nfs -volume strict_source -aggregate n1_data -size 1g -state online -junction-path /strict_source

2. Create a destination volume for the strictsync relationship of the type dp
cluster2::>
vol create -vserver c2_nfs -volume strict_dest -aggregate n1_data -size 1g -type dp

3. Create a snapmirror relationship with the stictsync policy and initialize the relationship
cluster2::>
snapmirror create -source-path c1_nfs:strict_source -destination-path c2_nfs:strict_dest -policy StrictSync

snapmirror initialize -destination-path c2_nfs:strict_dest

4. Create a file in the volume from the linux nfs client
linux::>
mkdir /mnt/strict_source
mount 192.168.0.210:/strict_source /mnt/strict_source
cd /mnt/strict_source/
echo content > file1

5. To test the fencing mechanisme we offline the volume
cluster2::>
vol offline -vserver c2_nfs -volume strict_dest

6. Run the volume show command and filter on these two fields:
“is-protocol-access-fenced”
“protocol-access-fenced-by”
cluster1::>
volume show -fields is-protocol-access-fenced, protocol-access-fenced-by -volume strict_source

vserver	volume	is-protocol-access-fenced	protocol-access-fenced-by
c1_nfs	strict_source	true	snapmirror_synchronous

7. On the linux client try to access the volume. This will fail.
Linux:
cd /mnt/strict_source
-bash: cd: /mnt/strict_source: Remote I/O error

8. Check the relationship on cluster2
cluster2::>
snapmirror show
c1_nfs:strict_source XDP c2_nfs:strict_dest Snapmirrored Transferring 0B **false** 07/24 09:28:02

9. Online the volume on the cluster2 again
cluster2::>
vol online -vserver c2_nfs -volume strict_dest

10. Check the relationship on cluster2 once more. It should have repaired the relationship
cluster2::>
snapmirror show
(status should be InSync)