ONTAP_snapmirror_sync

In this lab you will set up a snapmirror sync relationship between two volumes in two different clusters. You will reverse the relationship.

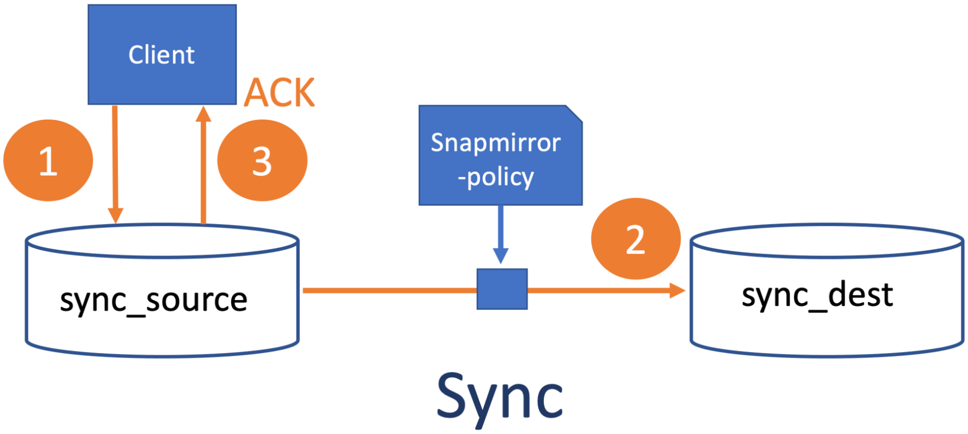
Prerequisites:

. 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 in for the snapmirror relationship and mount it
2. Create a destination volume of the type DP
3. Create a snapmirror relationship with the Sync snapmirror policy
4. Initualize the relationship, and check that it is InSync
5. On the linux NFS client, create a file in the volume
6. On the destination cluster, quiesce, break and delete the relationship
7. On the source cluster, release the relationship-info
8. On cluster1 create a new relationship to reverse the snapmirror and resync
9. On cluster2 mount the volume so it can be accessed by the linux client
10. On the linux client, access the new source volume

(see the next page for commands)



Commands

```
# 1. Create a source volume for the snapmirror relationship and mount it
cluster1::>
vol create sync_source -vserver c1_nfs -aggregate n1_data -size 1g -junction-path /ssource
```

```
# 2. Create a destination volume of the type DP
cluster2::>
vol create -vserver c2_nfs -volume sync_dest -size 1g -type DP -aggregate n1_data
```

```
# 3. Create a snapmirror relationship with the Sync snapmirror policy
cluster2::>
snapmirror create -source-path c1_nfs:sync_source -destination-path c2_nfs:sync_dest -policy Sync
```

```
# 4. Initialize the relationship
cluster2::>
snapmirror initialize -destination-path c2_nfs:sync_dest

cluster2::>
snapmirror show -fields status -destination-path c2_nfs:sync_dest
source-path        destination-path status
------------------ ---------------- ------
c1_nfs:sync_source c2_nfs:sync_dest  InSync
```

(status should show InSync)

```
# 5. Mount the source volume on the Linux NFS client and create a file
Linux:
mkdir /mnt/ssource
mkdir /mnt/sdest

mount 192.168.4.210:/ssource /mnt/ssource
echo content > /mnt/ssource/file
```

```
# 6. On cluster2 quiesce, break and delete the relationship
cluster2::>
snapmirror quiesce -destination-path c2_nfs:sync_dest
snapmirror break -destination-path c2_nfs:sync_dest
snapmirror delete -destination-path c2_nfs:sync_dest
```

# 7. On cluster1 release the relationship-info

```
cluster1::>
snapmirror release -relationship-info-only true -destination-path c2_nfs:sync_dest
```

(to avoid deletion of common snapshots)

# 8. On cluster1 create a new relationship to reverse the snapmirror and resync

```
snapmirror create -source-path c2_nfs:sync_dest -destination-path c1_nfs:sync_source -policy Sync
snapmirror resync -destination-path c1_nfs:sync_source
```

# 9. On cluster2 mount the volume so it can be accessed by the linux client

```
vol mount -vserver c2_nfs -volume sync_dest -junction-path /sync_dest
```

# 10. On the linux client, access the new source volume

Linux:

```
mount 192.168.4.211:/sync_dest /mnt/sdest
echo content > /mnt/sdest/file1
```