CDOT uses policies for most configurations. All policies are named after the volume the policy is being applied to. Pay attention to the switches being used to understand what is being done – seeing the volume name multiple times in a command can be confusing.

Everything should always be in lowercase.

The “-fields” switch on show commands is very useful. It can display data not listed in the default show command output and usually formats the output better.

***CREATING NEW BACKUPS ON CDOT***

**PRIMARY: CREATE SCHEDULE**

Create a cron job schedule named for the volume it will be applied to. This will state what time to create the snapshot – this is effectively the snapshot schedule. \*This is NOT performed for oracle database volumes (i.e. s01ora1) as the snapshots are scheduled to be created by database team. However, it is performed for oracle admin volumes (i.e. s01oraadm1).

The cron schedules are set at the cluster level rather than the vserver level. Running the base cron show command will list all cron schedules for the cluster.

ssh <PRIMARY CLUSTER> job schedule cron show

Running the show command and specifying the name will list a specific cron job. It will return no results if the schedule has not yet been created.

ssh <PRIMARY CLUSTER> job schedule cron show -name <VOLUME NAME>

This will schedule a cron for 23:00 local data center time.

ssh <PRIMARY CLUSTER> job schedule cron create -name <VOLUME NAME> -hour 23 -minute 0

This will schedule a cron for 01:00 local data center time.

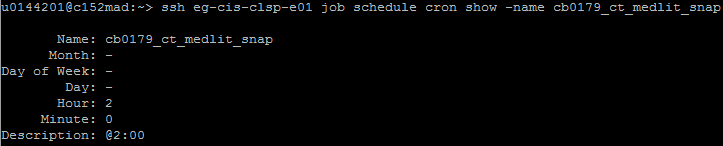
ssh <PRIMARY CLUSTER> job schedule cron create -name <VOLUME NAME> -hour 1 -minute 0

Running the show command after creating the cron will list the details of the schedule.

ssh <PRIMARY CLUSTER> job schedule cron show -name <VOLUME NAME>

Example:

The schedule for this volume is set for 02:00



**PRIMARY: CREATE POLICY**

Create the snapshot policy, which will use the schedule created in the last step. The policy will also be named after the volume. The policy will dictate how many snapshots to keep locally, the snapmirror-label, and the snapshot prefix. On the primary vserver, the local retention is always set to 7. For standard snapvault relationships, the snapmirror-label in the policy will be “snapvault”. The snapshot is prefixed as “sv\_” followed by the volume name. \*This is NOT performed for oracle database volumes (i.e. s01ora1) as the snapshots are scheduled to be created by database team. However, it is performed for oracle admin volumes (i.e. s01oraadm1).

This will show all snapshot policies for a given vserver. If no policies exist it will return no results.

ssh <PRIMARY CLUSTER> volume snapshot policy show -vserver <PRIMARY VSERVER>

To look for the specific policy, it will be named after the volume it is applied to, or will be applied to. If the policy hasn’t been created yet it will return no results.

ssh <PRIMARY CLUSTER> volume snapshot policy show -vserver <PRIMARY VSERVER> -policy <VOLUME NAME>

Create the policy. The “schedule1” switch is referring to the cron job created in the last section.

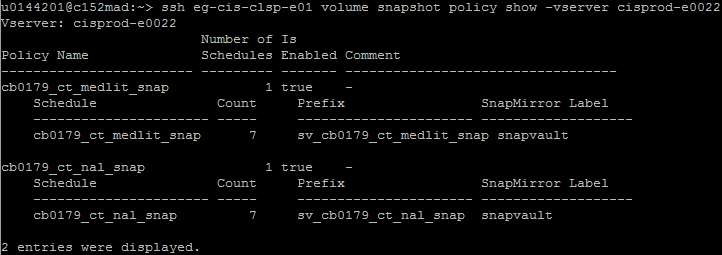
ssh <PRIMARY CLUSTER> volume snapshot policy create -vserver <PRIMARY VSERVER> -policy <VOLUME NAME> -enabled true -schedule1 <VOLUME NAME> -count1 7 -snapmirror-label1 snapvault -prefix1 sv\_<VOLUME NAME>

Confirm the policy has been created.

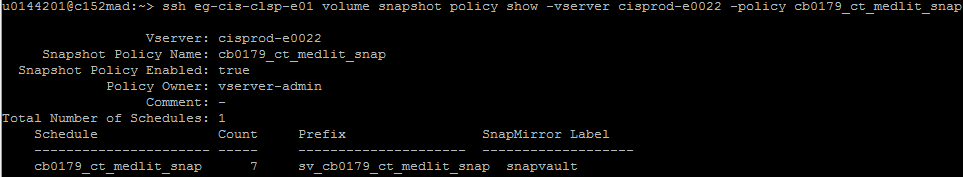
ssh <PRIMARY CLUSTER> volume snapshot policy show -vserver <PRIMARY VSERVER> -policy <VOLUME NAME>

Example:

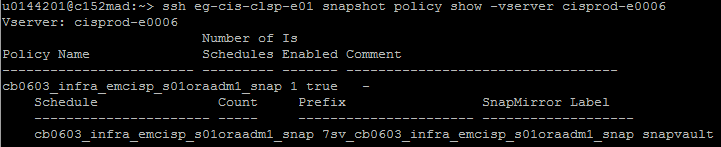
This shows the snapshot policies on the vserver. The policy name and schedule name are both named after the volume the policy will be applied to. The snapshot prefix is set to sv\_ followed by the volume name. Both policies are set to keep 7 snapshots and have the “snapvault” snapmirror-label configured.



This shows a specific policy on the vserver. Notice the output is formatted differently than the vserver view above.



This shows the snapshot policies on a LION vserver. The only policy is for the admin volume.



**PRIMARY: APPLY POLICY**

After the cron schedule is created and the snapshot policy is created, the policy can be applied to the volume it is named for. Once applied, snapshots will start being created for the volume at the scheduled time. \*This is NOT performed for oracle database volumes (i.e. s01ora1) as the snapshots are scheduled to be created by database team. However, it is performed for oracle admin volumes (i.e. s01oraadm1).

This shows what snapshot policies are currently applied to all volumes in a vserver.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -fields snapshot-policy

This shows what snapshot policy is currently applied a specific volume.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -fields snapshot-policy

Apply the policy to the volume. The policy is named after the volume.

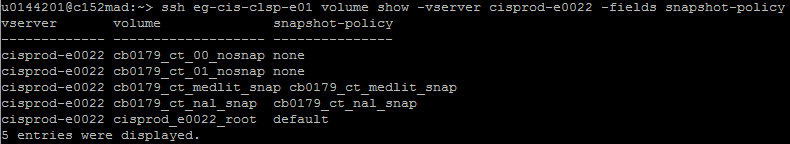
ssh <PRIMARY CLUSTER> volume modify -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -snapshot-policy <VOLUME NAME>

Confirm the policy has been applied.

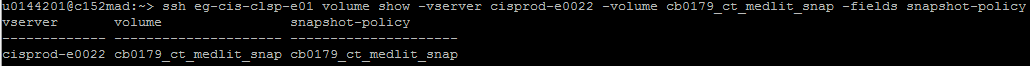
ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -fields snapshot-policy

Example:

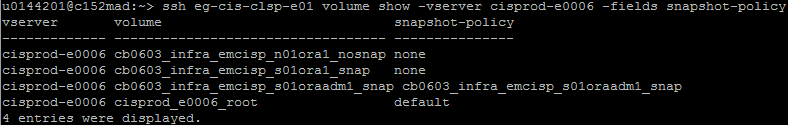
This shows all volumes on this vserver and their applied snapshot policies. The \_nosnap volumes have no policy set, root volume uses the default policy, and the two \_snap data volumes have policies applied which match their volume name.



This shows the policy for a specified volume.



This shows the snapshot policies applied to volumes in a LION vserver. The oracle database volume (s01ora1) does not have a policy applied because the snapshots are scheduled and created by the database team.



**PRIMARY: VOLUME SPACE MANAGEMENT**

The space management setting tells the volume what to do if it gets full. By default, volumes have volume\_grow set, but \_snap volumes need to be set to snap\_delete. This only dictates what the volume should attempt, but does not mean that autosize or snap autodelete are actually enabled; those would need to be checked separately.

\*Autosize is disabled on the primary and is enabled on the backup/secondary.

\*Autodelete is enabled on the primary and is disabled on the backup/secondary.

\*On the primary, \_nosnap will have volume\_grow set but autosize will be disabled.

\*On the primary, \_snap will have snap\_delete set, autosize disabled, and snap autodelete enabled.

\*On the secondary, all volumes are set to volume\_grow, have autosize enabled, and snap autodelete disabled.

This shows the space management setting for all volumes in the vserver.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -fields space-mgmt-try-first

This shows the space management setting for a single volume in the vserver.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -fields space-mgmt-try-first

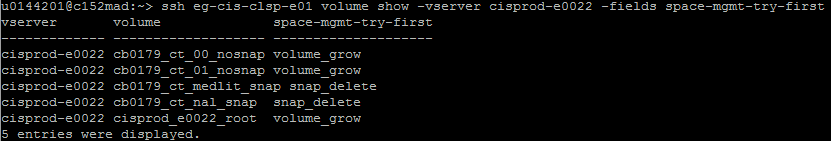
Set the volume to snap\_delete.

ssh <PRIMARY CLUSTER> volume modify -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -space-mgmt-try-first snap\_delete

Confirm the volume is now set to snap\_delete.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -fields space-mgmt-try-first

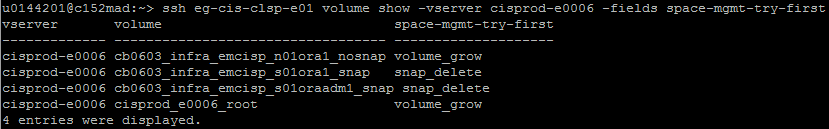
This shows the space management setting for all volumes in the vserver. The \_snap volumes are set to snap\_delete and the rest are set to volume\_grow. This does not mean that autosize or snap autodelete are actually enabled, those would need to be checked separately.



This shows the space management setting for a single volume in the vserver. It is a \_snap volume so it is set to snap\_delete.



This shows the space management settings for the volumes in a LION vserver. All the \_snap volumes have snap\_delete configured.



**PRIMARY: SNAPSHOT AUTODELETE**

To avoid snapshot overflow we enable snapshot autodelete on \_snap volumes. It doesn’t really matter if \_nosnap volumes have autodelete enabled or not since they have no snapshot reserve to trigger this effect.

\*Autodelete is always enabled on the primary and is never enabled on backup storage.

\*On the primary, \_snap will have snap\_delete set, autosize disabled, and snap autodelete enabled.

These are the options which should be set when autodelete is enabled:

Enabled true

Commitment try

Trigger snap\_reserve

Target Free Space 20%

Delete Order oldest\_first

Defer Delete user\_created

Defer Delete Prefix (not specified)

Destroy List none

This shows the snapshot autodelete settings for all volumes in a vserver.

ssh <PRIMARY CLUSTER> volume snapshot autodelete show -vserver <PRIMARY VSERVER>

This shows the snapshot autodelete settings for a single volume in a vserver.

ssh <PRIMARY CLUSTER> volume snapshot autodelete show -vserver <PRIMARY VSERVER> -volume <VOLUME>

This shows the snapshot autodelete settings for all snapshot volumes in a vserver. This was commonly used in the tech refresh cheat sheets.

ssh <PRIMARY CLUSTER> volume snapshot autodelete show -vserver <PRIMARY VSERVER> -volume ‘\*\_snap\*’

Enable snapshot autodelete if it is disabled.

ssh <PRIMARY CLUSTER> volume snapshot autodelete modify -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -enabled true

Set the trigger to snap\_reserve if not done already.

ssh <PRIMARY CLUSTER> volume snapshot autodelete modify -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -trigger snap\_reserve

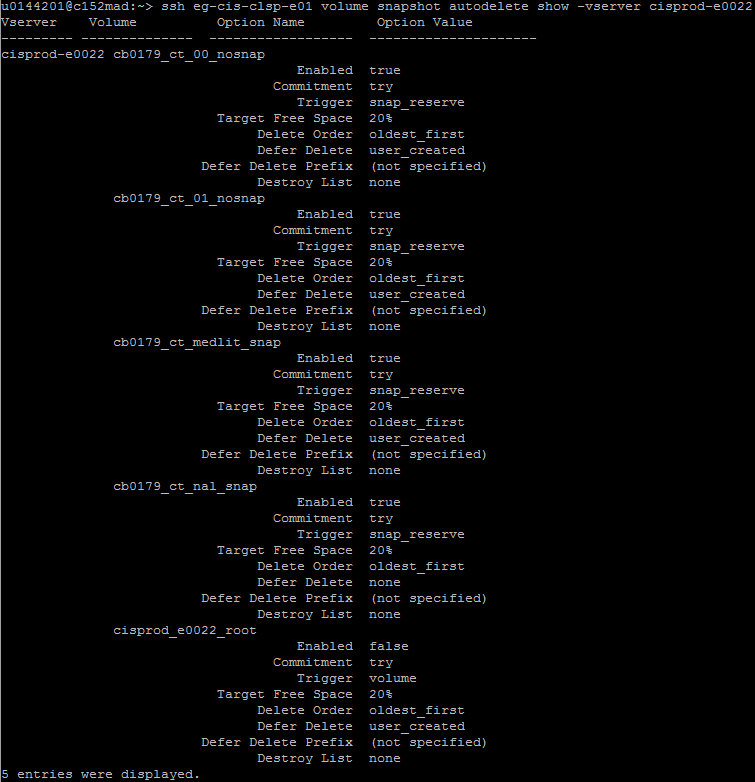
The modify command can be used to modify any of the other autodelete settings if required, but

ssh <PRIMARY CLUSTER> volume snapshot autodelete modify -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> <option> <setting>

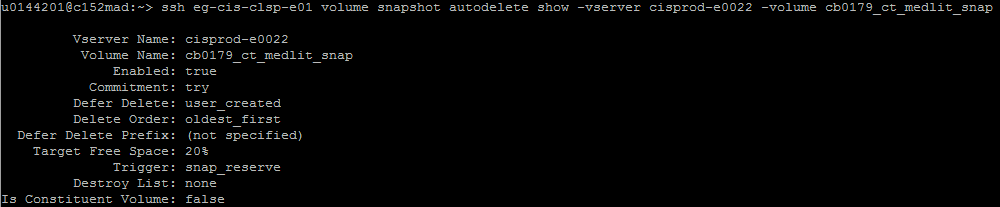
Confirm the settings afterwards.

ssh <PRIMARY CLUSTER> volume snapshot autodelete show -vserver <PRIMARY VSERVER> -volume <VOLUME>

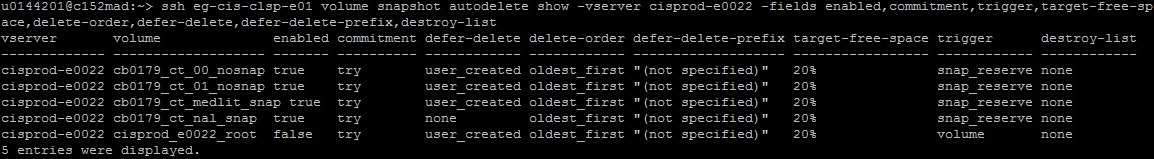
This shows the snapshot autodelete settings for the volumes in this vserver.



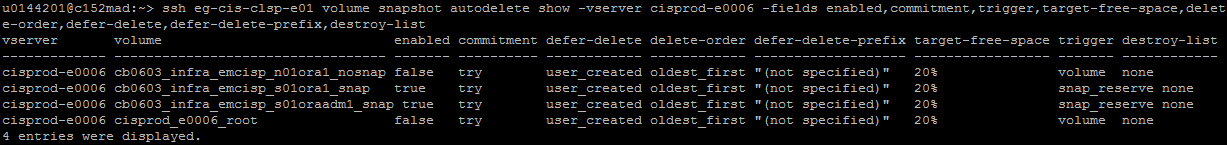
This shows the snapshot autodelete settings for a specified volume in a vserver, notice it is a little different than the above.



This shows the snapshot autodelete settings for the volumes in this vserver, but using the “-fields” switch which formats the output differently.



This shows the snapshot autodelete settings for the volumes in a LION vserver, using the “-fields” switch. Both \_snap volumes have snap autodelete enabled with the snap\_reserve trigger.



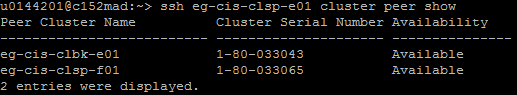
**PRIMARY: SELECT BACKUP CLUSTER**

Now that we have snapshots being created on the primary, we need to get them vaulted. The first step is to select a backup location.

Check which clusters are peered to the primary cluster. Some clusters may have a mix of data and backup clusters peered to it. Backup clusters are named “BK” to represent backup (i.e. eg-cis-clbk-e01). Backups are always same-site, so select the appropriate backup cluster for the data center.

ssh <PRIMARY CLUSTER> cluster peer show

This shows which clusters are peered to this cluster. It shows one data cluster and one backup cluster.



**SECONDARY: SELECT BACKUP VSERVER**

Once a backup cluster is identified, next identify which vserver is appropriate. Backup vservers may be specific to an application such as CLEAR, or a specific protocol such as ISCSI. Our examples are standard NFS so the standard NFS backup vserver will be used.

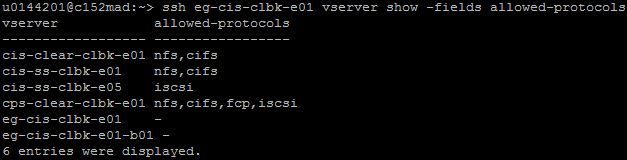
The vservers should have protocols specifically allowed and can be shown using the “-fields” switch.

ssh <BACKUP CLUSTER> vserver show -fields allowed-protocols

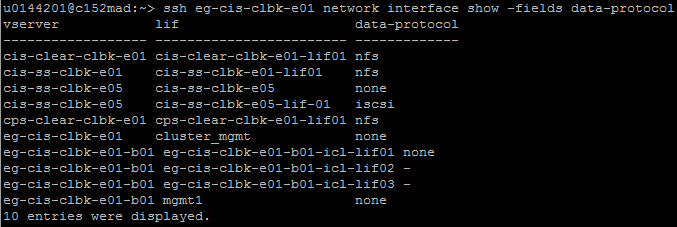
Another way to confirm is to check what protocol traffic is allowed on the vserver LIFs, again using the “-fields” switch.

ssh <BACKUP CLUSTER> network interface show -fields data-protocol

This shows the vservers and their allowed protocols.



This shows the vserver LIFs and their allowed protocols. Notice how some vserver LIFs list only nfs while the vserver allowed protocols in the above screenshot list nfs and cifs. The protocol is ultimately locked down on the LIF so in this case cifs connections will not be allowed for those vservers.



**SECONDARY: INITIATE VSERVER PEER**

Now that a backup vserver is identified, it needs to be peered with the primary vserver. The vserver peer relationship is initiated from the backup vserver and then accepted on the primary vserver. The vserver peer is only configured once, once it is in place any future snapvault transfers can be set up without doing this step.

Check if a peer relationship already exists. The peer state is important.

ssh <BACKUP CLUSTER> vserver peer show -vserver <BACKUP VSERVER> -peer-vserver <PRIMARY VSERVER>

What the different peer states mean:

peered Vserver peer relationship is established and the

applications can use peer relationship

pending Vserver peer relationship is initiated in the

peer cluster and local cluster needs to accept

the request

initializing Communicating to the peer cluster for

initializing the peer relationship

initiated Relationship initiated in local cluster

rejected Relationship initiated and peer cluster rejected

the same

suspended Relationship suspended, should not initiate any

data transfers

deleted Relationship got deleted on peer Cluster

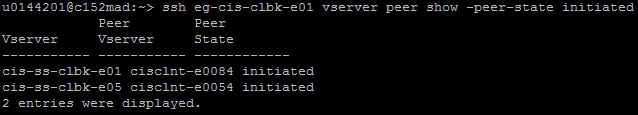
This can be used to show all peer relationships for the backup vserver. The list is long because the single backup vserver is peered to many primary vservers.

ssh <BACKUP CLUSTER> vserver peer show -vserver <BACKUP VSERVER>

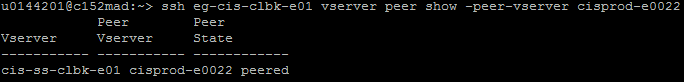
Initiate the vserver peer relationship.

ssh <BACKUP CLUSTER> vserver peer create -vserver <BACKUP VSERVER> -peer-vserver <PRIMARY VSERVER> -applications snapmirror -peer-cluster <PRIMARY CLUSTER>

This shows vserver peer relationships which have been initiated by the backup vserver, but not yet accepted by the primary vserver.



This shows a fully peered vserver peer relationship, accepted by the primary vserver.



**PRIMARY: ACCEPT VSERVER PEER**

After the vserver peer relationship is initiated from the backup cluster in the previous section, the primary vserver has to accept the relationship.

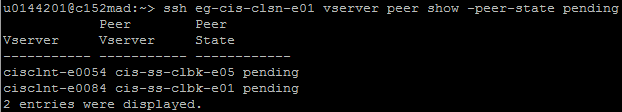
Check if any peer relationships exist for the primary vserver. The peer state is important, the different state meanings were described in the previous section.

ssh <PRIMARY CLUSTER> vserver peer show -vserver <PRIMARY VSERVER> -peer-vserver <BACKUP VSERVER>

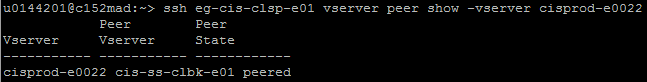
Accept the vserver peer relationship.

ssh <PRIMARY CLUSTER> vserver peer accept -vserver <PRIMARY VSERVER> -peer-vserver <BACKUP VSERVER>

This shows vserver peer relationships which have been initiated by the backup vserver and are pending to be accepted by the primary vserver.



This shows a fully peered vserver peer relationship, accepted by the primary vserver.



**SECONDARY: CREATE BACKUP VOLUME**

Create the backup volume which will hold the transferred snapshots. Backup volumes are named sv\_<retention##>\_<VOLUME NAME>. The volume is created as a DP style which means “data protection” and is read-only. Snapvault relationships will only work if the secondary volume is type DP.

Backup volumes are typically 3 times the size of the source volume. The language of the primary and secondary volumes must match.

Check the language and size of the primary volume. The size field is the total volume size including snap reserve. Language is usually en\_US.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -volume <VOLUME NAME> -fields language,size

This can be used to check the language and size of all volumes in the vserver.

ssh <PRIMARY CLUSTER> volume show -vserver <PRIMARY VSERVER> -fields language,size

Get the backup aggregate name. Backup clusters will have a single data aggregate, most will be called aggr1\_data\_b01.

ssh <BACKUP CLUSTER> aggr show

Check if a backup volume already exists with the same name.

ssh <BACKUP CLUSTER> volume show -vserver <BACKUP VSERVER> -volume sv\_<retention##>\_<VOLUME NAME>

This can be used to check for all volume with the primary volume name, but may list different retentions in the name or exist on different backup vservers in the backup cluster.

ssh <BACKUP CLUSTER> volume show -volume ‘\*<VOLUME NAME>’

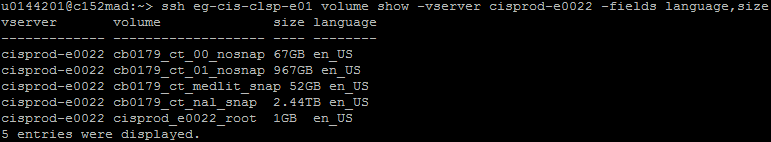
Create the backup volume.

ssh <BACKUP CLUSTER> volume create -vserver <BACKUP VSERVER> -volume sv\_<retention##>\_<VOLUME NAME> -aggregate <backup data aggregate> -size <size> -security-style unix -space-guarantee none -percent-snapshot-space 0 -language <language> -type DP

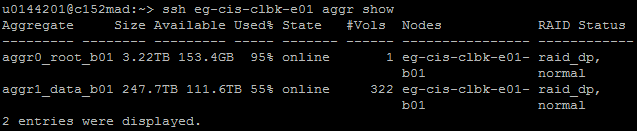
Confirm the backup volume is created.

ssh <BACKUP CLUSTER> volume show -vserver <BACKUP VSERVER> -volume sv\_<retention##>\_<VOLUME NAME>

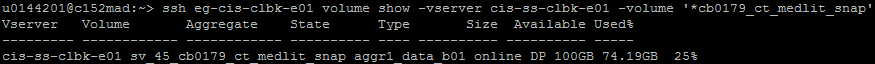
This lists the language and size for all volumes in a vserver.



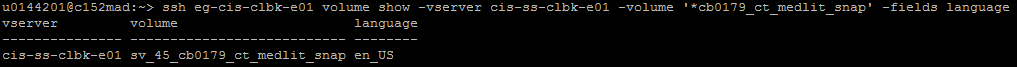
This lists the aggregates on the backup cluster. Always use the data aggregate.



This command checked the chosen backup vserver for any volumes named for the primary volume, no matter the retention in the volume name. The volume displayed is named for 45-day retention and is DP style. The volume is showing size 100g, which includes the snapshot reserve.



Use the “-fields” switch to check the language of the backup volume.



**SECONDARY: CREATE SCHEDULE**

On the primary cluster, the schedules were unique per volume. On the backup cluster we configure one schedule per hour. The schedules are named xdp\_<hour#> with hour being a number between 0 and 23.

Snapvault transfers should be scheduled one hour after snapshot creation. For example, if the primary volume is scheduled for 19:00, the vault should be scheduled for 20:00. If the is scheduled for 23:00, the vault should be scheduled for 00:00.

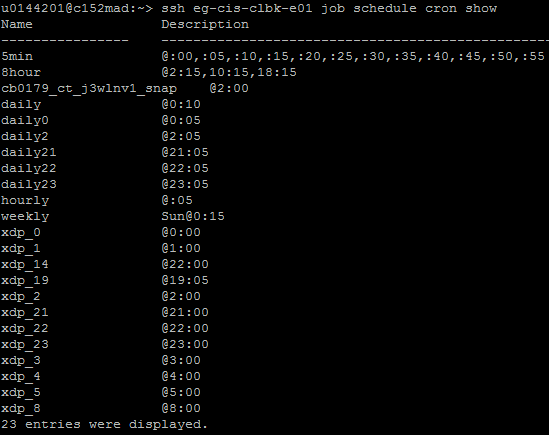
Show the existing schedules. Some schedules may be named wrong, for example xdp\_14 scheduled for @22:00 is not correct. If this happens, create and use a new one with the appropriate name.

ssh <BACKUP CLUSTER> job schedule cron show

If the schedule required does not exist, create it.

ssh <BACKUP CLUSTER> job schedule cron create -name xdp\_<hour#0-23> -hour <0-23> -minute 0

This shows the schedules for this backup cluster. Schedules xdp\_14 and xdp\_22 are both configured for 22:00. The xdp\_22 schedule would be the right one to use.



**SECONDARY: CREATE POLICY**

The snapmirror policy will be named after the backup volume. The snapmirror policy defines the archive retention. This is also where the snapmirror-label comes into play. The snapmirror-label configured in the policy must match the snapmirror-label applied to the snapshots created on the primary (configured in the snapshot policy on the primary.) For standard snapvault relationships, the snapmirror-label is set to “snapvault”.

Check if the policy already exists.

ssh <BACKUP CLUSTER> snapmirror policy show -vserver <BACKUP VSERVER> -policy sv\_<retention##>\_<VOLUME NAME>

Create the snapmirror policy. This just creates the base policy, but has no rules yet. If show is run again it would show an empty policy.

ssh <BACKUP CLUSTER> snapmirror policy create -vserver <BACKUP VSERVER> -policy sv\_<retention##>\_<VOLUME NAME>

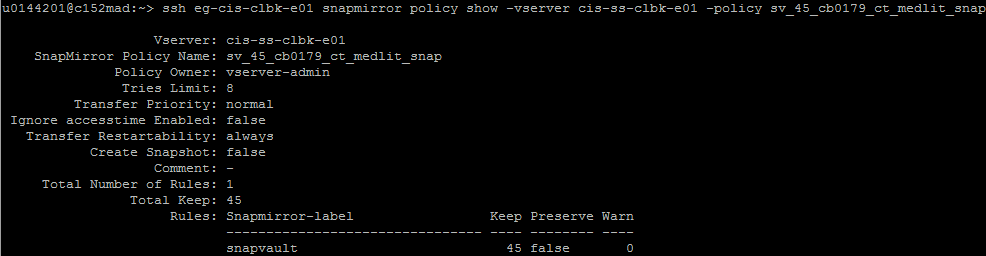
Add a rule to the snapmirror policy, defining the snapmirror-label and retention.

ssh <BACKUP CLUSTER> snapmirror policy add-rule -vserver <BACKUP VSERVER> -policy sv\_<retention##>\_<VOLUME NAME> -snapmirror-label snapvault -keep <retention#>

Confirm the policy is correct.

ssh <BACKUP CLUSTER> snapmirror policy show -vserver <BACKUP VSERVER> -policy sv\_<retention##>\_<VOLUME NAME>

This shows the snapmirror policy for a single backup volume. It is set with a 45-day retention and snapmirror-label is “snapvault”.



**SECONDARY: INITIALIZE SNAPVAULT**

The last thing to do is to create the snapvault relationship and then initialize it. The snapmirror command is still used for this. A snapvault relationship is a snapmirror-relationship that is style XDP, X for XFER (transfer). A plain DP relationship is a standard snapmirror. A TDP style relationship is used specifically for 7Mode to CDOT mirrors.

Check for existing snapmirrors configured for the backup volume.

ssh <BACKUP CLUSTER> snapmirror show -destination-path <BACKUP VSERVER>:sv\_<retention##>\_<VOLUME NAME>

Create the XDP snapmirror relationship. Make sure to set the type as XDP. Set the schedule which defines when the snapvault occurs, and the policy which defines the retention. The policy and backup volume have the same name.

ssh <BACKUP CLUSTER> snapmirror create -source-path <PRIMARY VSERVER>:<VOLUME NAME> -destination-path <BACKUP VSERVER>:sv\_<retention##>\_<VOLUME NAME> -type XDP -schedule xdp\_<hour#0-23> -policy sv\_<retention##>\_<VOLUME NAME>

Initialize the XDP snapmirror relationship.

ssh <BACKUP CLUSTER> snapmirror initialize -destination-path <BACKUP VSERVER>:sv\_<retention##>\_<VOLUME NAME>

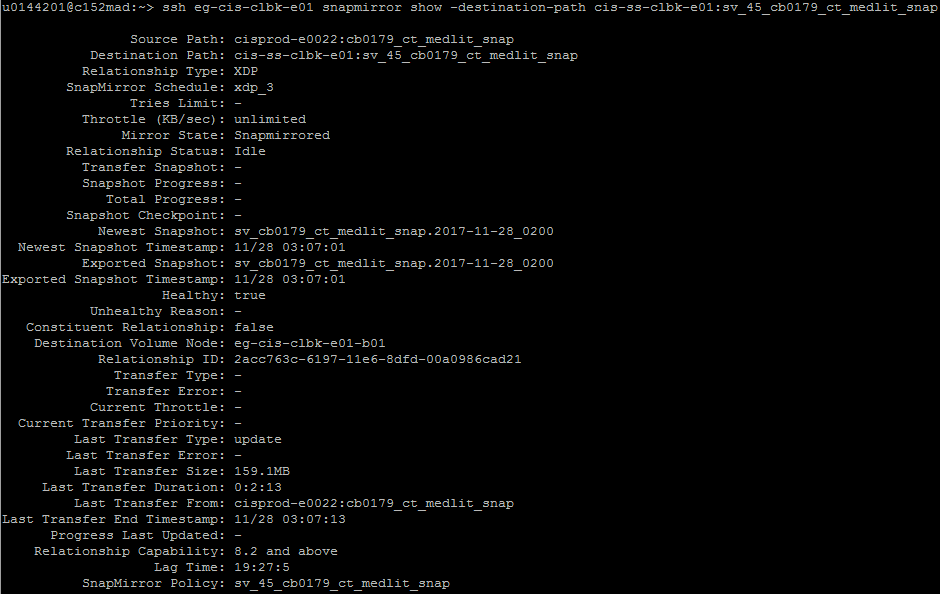
Check up on the snapmirror until it completes.

ssh <BACKUP CLUSTER> snapmirror show -destination-path <BACKUP VSERVER>:sv\_<retention##>\_<VOLUME NAME>

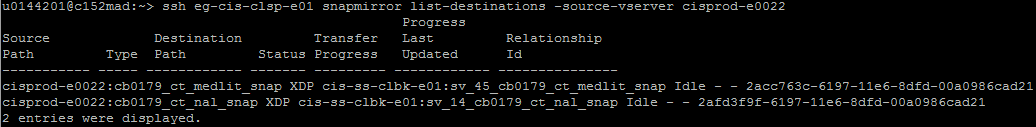
The relationship can also be checked on the primary, but with less detail, using the snapmirror list-destinations command.

ssh <PRIMARY CLUSTER> snapmirror list-destinations -source-path <PRIMARY VSERVER>:<VOLUME NAME>

This shows the XDP snapmirror relationship details for a backup volume. The below volume is scheduled to vault at 03:00 local data center time. The lag is currently 19 hours and the relationship is healthy.



This shows the XDP snapmirror relationships for a primary vserver, using the -source-vserver switch instead of -source-path. Not many details are shown on the primary.



2015NR templates were referenced as well as D&E documentation on sharepoint and storage PCM.

Base CDOT snapvault standards

<https://theshare.thomsonreuters.com/sites/DCO_Storage/Unified%20Storage%20DE%20Documents/NETAPP/Ontap%20CDOT%20Standards/Snapvault%20Express%20Guide.pdf>

NFS LION on CDOT

<https://theshare.thomsonreuters.com/sites/DCO_Storage/Unified%20Storage%20DE%20Documents/NETAPP/Ontap%20CDOT%20Standards/Oracle%20on%20NetApp%20cDOT%20-%20Deployment%20Guidelines.docx>

Generic NFS on CDOT

<https://theshare.thomsonreuters.com/sites/DCO_Storage/Unified%20Storage%20DE%20Documents/NETAPP/Ontap%20CDOT%20Standards/NFSv4%20cDOT.docx>

Replication guidelines CDOT

<https://theshare.thomsonreuters.com/sites/DCO_Storage/Unified%20Storage%20DE%20Documents/NETAPP/Ontap%20CDOT%20Standards/Thomson%20Reuters%20-%20cDOT%20Replication%20-%20Deployment%20Guidelines%20v4.docx>