# Pool Operations

A DAOS pool is a storage reservation that can span any storage nodes and is managed by the administrator. The amount of space allocated to a pool is decided at creation time and can eventually be expanded through the management interface.

## Pool Creation/Destroy

A DAOS pool can be created and destroyed through a utility called dmg to manage storage pools from the command line.

**To create a pool:**

$ dmg pool create --size=NTB

This command creates a pool distributed across the DAOS servers with a target size on each server that is comprised of N TB of NVMe storage and N \* 0.06 (i.e. 6% of NVMe) of SCM storage. The default SCM:NVMe ratio may be adjusted at pool creation time as described below. The UUID allocated to the newly created pool is printed to stdout (referred to as ${puuid}) as well as the pool service replica ranks.

NB: The –scm-size and –nvme-size options still exist, but should be considered deprecated and will likely be removed in a future release.

$ dmg pool create --help  
...  
[create command options]  
 -g, --group= DAOS pool to be owned by given group, format name@domain  
 -u, --user= DAOS pool to be owned by given user, format name@domain  
 -p, --name= Unique name for pool (set as label)  
 -a, --acl-file= Access Control List file path for DAOS pool  
 -z, --size= Total size of DAOS pool (auto)  
 -t, --scm-ratio= Percentage of SCM:NVMe for pool storage (auto) (default: 6)  
 -k, --nranks= Number of ranks to use (auto) (default: all)  
 -v, --nsvc= Number of pool service replicas (default: 3)  
 -s, --scm-size= Per-server SCM allocation for DAOS pool (manual)  
 -n, --nvme-size= Per-server NVMe allocation for DAOS pool (manual)  
 -r, --ranks= Storage server unique identifiers (ranks) for DAOS pool  
 -S, --sys= DAOS system that pool is to be a part of (default: daos\_server)

The typical output of this command is as follows:

$ dmg pool create --size 50GB  
Creating DAOS pool with automatic storage allocation: 50 GB NVMe + 6.00% SCM  
Pool created with 6.00% SCM/NVMe ratio  
-----------------------------------------  
 UUID : 8a05bf3a-a088-4a77-bb9f-df989fce7cc8  
 Replica Ranks : [1-3]  
 Target Ranks : [0-15]  
 Size : 50 GB  
 SCM : 3.0 GB (188 MB / rank)  
 NVMe : 50 GB (3.2 GB / rank)

This created a pool with UUID 8a05bf3a-a088-4a77-bb9f-df989fce7cc8, with redundancy enabled by default (pool service replicas on ranks 1-3).

If no redundancy is desired, use –nsvc=1 in order to specify that only a single pool service replica should be created.

**To destroy a pool:**

$ dmg pool destroy --pool=${puuid}

**To evict handles/connections to a pool:**

$ dmg pool evict --pool=${puuid}  
`  
  
\*\*To see a list of the pools in your DAOS system:\*\*  
  
```bash  
$ dmg system list-pools

This will return a table of pool UUIDs and the ranks of their pool service replicas. For example:

$ dmg system list-pools  
localhost:10001: connected  
Pool UUID Svc Replicas  
--------- ------------  
2a8ec3b2-729b-4617-bf51-77f37f764194 0,1  
a106d667-5c5d-4d6f-ac3a-89099196c41a 0  
85141a07-e3ba-42a6-81c2-3f18253c5e47 0

## Pool Properties

|  |  |
| --- | --- |
| **Pool Property** | **Description** |
| DAOS\_PROP\_PO\_LABEL | A string that the administrator can associate with a pool. e.g., project A, project B, IO500 test pool |
| DAOS\_PROP\_PO\_ACL | Access control list (ACL) associated with the pool |
| DAOS\_PROP\_PO\_SPACE\_RB | Space reserved on each target for rebuild purpose |
| DAOS\_PROP\_PO\_SELF\_HEAL | Define whether the pool wants automatically-trigger or manually-triggered self-healing |
| DAOS\_PROP\_PO\_RECLAIM | Tune space reclaim strategy based on time interval, io activities |

At creation time, currently only ACL may be specified via dmg pool create.

While those pool properties are currently stored persistently with pool metadata, many of them are still under development. Moreover, the ability to modify some of those properties on an existing pool will be provided in a future release.

### Modifying DAOS\_PROP\_PO\_RECLAIM property

To modify a pool’s DAOS\_PO\_RECLAIM property:

$ dmg pool set-prop --pool=<UUID> --name=reclaim --value=${strategy}

Three reclaim strategies are supported:

* “disabled” : Never trigger aggregation.
* “lazy” : Trigger aggregation only when there is no IO activities or SCM free space is under pressure (default strategy)
* “time” : Trigger aggregation regularly despite of IO activities.

### Querying a pool’s properties

The user-level administration daos utility may be used to query a pool’s properties. Refer to the manual page for full daos usage details.

$ daos pool get-prop --pool=b1e9f5c0-ce10-42ab-b19e-081032400611  
Pool properties for b1e9f5c0-ce10-42ab-b19e-081032400611 :  
label: pool label not set  
rebuild space ratio: 0%  
self-healing: auto-exclude,auto-rebuild  
reclaim strategy: lazy  
owner: username@  
owner-group: username@  
Access Control List:  
#  
# (remainder of output not shown)  
#

Additionally, a pool’s properties may be retrieved using the libdaos API daos\_pool\_query() function. Refer to the file src/include/daos\_pool.h Doxygen comments and the online documentation available [here](https://daos-stack.github.io/html/).

## Access Control Lists

Client user and group access for pools are controlled by [Access Control Lists (ACLs)](https://daos-stack.github.io/overview/security/#access-control-lists). Most pool-related tasks are performed using the DMG administrative tool, which is authenticated by the administrative certificate rather than user-specific credentials.

Access-controlled client pool accesses include:

* Connecting to the pool.
* Querying the pool.
* Creating containers in the pool.
* Deleting containers in the pool.

This is reflected in the set of supported [pool permissions](https://daos-stack.github.io/overview/security/#permissions).

A user must be able to connect to the pool in order to access any containers inside, regardless of their permissions on those containers.

### Ownership

Pool ownership conveys no special privileges for access control decisions. All desired privileges of the owner-user (OWNER@) and owner-group (GROUP@) must be explicitly defined by an administrator in the pool ACL.

### Creating a pool with a custom ACL

To create a pool with a custom ACL:

$ dmg pool create --size <size> --acl-file <path>

The ACL file format is detailed in the [here](https://daos-stack.github.io/overview/security/#acl-file).

### Displaying a Pool’s ACL

To view a pool’s ACL:

$ dmg pool get-acl --pool <UUID>

The output is in the same string format used in the ACL file during creation, with one Access Control Entry (i.e., ACE) per line.

### Modifying a Pool’s ACL

For all of these commands using an ACL file, the ACL file must be in the format noted above for pool creation.

#### Overwriting the ACL

To replace a pool’s ACL with a new ACL:

$ dmg pool overwrite-acl --pool <UUID> --acl-file <path>

#### Adding and Updating ACEs

To add or update multiple entries in an existing pool ACL:

$ dmg pool update-acl --pool <UUID> --acl-file <path>

To add or update a single entry in an existing pool ACL:

$ dmg pool update-acl --pool <UUID> --entry <ACE>

If there is no existing entry for the principal in the ACL, the new entry is added to the ACL. If there is already an entry for the principal, that entry is replaced with the new one.

#### Removing an ACE

To delete an entry for a given principal in an existing pool ACL:

$ dmg pool delete-acl --pool <UUID> --principal <principal>

The principal corresponds to the principal portion of an ACE that was set during pool creation or a previous pool ACL operation. For the delete operation, the principal argument must be formatted as follows:

* Named user: u:username@
* Named group: g:groupname@
* Special principals: OWNER@, GROUP@, and EVERYONE@

The entry for that principal will be completely removed. This does not always mean that the principal will have no access. Rather, their access to the pool will be decided based on the remaining ACL rules.

## Pool Query

The pool query operation retrieves information (i.e., the number of targets, space usage, rebuild status, property list, and more) about a created pool. It is integrated into the dmg utility.

**To query a pool:**

$ dmg pool query --pool <UUID>

Below is the output for a pool created with SCM space only.

pool=47293abe-aa6f-4147-97f6-42a9f796d64a  
 Pool 47293abe-aa6f-4147-97f6-42a9f796d64a, ntarget=64, disabled=8  
 Pool space info:  
 - Target(VOS) count:56  
 - SCM:  
 Total size: 28GB  
 Free: 28GB, min:505MB, max:512MB, mean:512MB  
 - NVMe:  
 Total size: 0  
 Free: 0, min:0, max:0, mean:0  
 Rebuild done, 10 objs, 1026 recs

The total and free sizes are the sum across all the targets whereas min/max/mean gives information about individual targets. A min value close to 0 means that one target is running out of space.

NB: the Versioning Object Store (VOS) may reserve a portion of the SCM and NVMe allocations to mitigate fragmentation and for background operations (e.g., aggregation, garbage collection). The amount of storage set aside depends on the size of the target, and may take up 2+ GB. Therefore, out of space conditions may occur even while pool query may not reveal min approaching zero.

The example below shows a rebuild in progress and NVMe space allocated.

pool=95886b8b-7eb8-454d-845c-fc0ae0ba5671  
 Pool 95886b8b-7eb8-454d-845c-fc0ae0ba5671, ntarget=64, disabled=8  
 Pool space info:  
 - Target(VOS) count:56  
 - SCM:  
 Total size: 28GB  
 Free: 28GB, min:470MB, max:512MB, mean:509MB  
 - NVMe:  
 Total size: 56GB  
 Free: 28GB, min:470MB, max:512MB, mean:509MB  
 Rebuild busy, 75 objs, 9722 recs

Additional status and telemetry data are planned to be exported through management tools and will be documented here once available.

## Pool Modifications

### Target Exclusion and Self-Healing

## Pool Exclude

An operator can exclude one or more targets from a specific DAOS pool using the rank the target resides on as well as the target idx on that rank. If a target idx list is not provided then all targets on the rank will be excluded. Excluding a target will automatically start the rebuild process.

**To exclude a target from a pool:**

$ dmg pool exclude --pool=${puuid} --rank=${rank} --target-idx=${idx1},${idx2},${idx3}

The pool target exclude command accepts 3 parameters:

* The pool UUID of the pool that the targets will be excluded from.
* The rank of the target(s) to be excluded.
* The target Indices of the targets to be excluded from that rank (optional).

## Pool Drain

Alternatively when an operator would like to remove one or more pool targets without the system operating in degraded mode Drain can be used. A pool drain operation will initiate rebuild without excluding the designated target until after the rebuild is complete. This allows the target(s) drained to continue to perform I/O while the rebuild operation is ongoing. Drain additionally enables non-replicated data to be rebuilt onto another target whereas in a conventional failure scenario non-replicated data would not be integrated into a rebuild and would be lost.

**To drain a target from a pool:**

$ dmg pool drain --pool=${puuid} --rank=${rank} --target-idx=${idx1},${idx2},${idx3}

The pool target drain command accepts 3 parameters:

* The pool UUID of the pool that the targets will be drained from.
* The rank of the target(s) to be drained.
* The target Indices of the targets to be drained from that rank (optional).

### Target Reintegration

After a target failure an operator can fix the underlying issue and reintegrate the affected targets to restore the pool to its original state. The operator can either reintegrate specific targets for a rank by supplying a target idx list, or reintegrate an entire rank by omitting the list.

$ dmg pool reintegrate --pool=${puuid} --rank=${rank} --target-idx=${idx1},${idx2},${idx3}

The pool reintegrate command accepts 3 parameters:

* The pool UUID of the pool that the targets will be reintegrated into.
* The rank of the affected targets.
* The target Indices of the targets to be reintegrated on that rank (optional).

When rebuild is triggered it will list the operations and their related targets by their rank ID and target index.

Target (rank 5 idx 0) is down.  
Target (rank 5 idx 1) is down.  
...  
(rank 5 idx 0) is excluded.  
(rank 5 idx 1) is excluded.

These should be the same values used when reintegrating the targets.

$ dmg pool reintegrate --pool=${puuid} --rank=5 --target-idx=0,1

### Pool Extension

#### Target Addition & Space Rebalancing

Full Support for online target addition and automatic space rebalancing is planned for a future release and will be documented here once available.

Until then the following command(s) are placeholders and offer limited functionality related to Online Server Addition/Rebalancing operations.

An operator can choose to extend a pool to include ranks not currently in the pool. This will automatically trigger a server rebalance operation where objects within the extended pool will be rebalanced across the new storage.

$ dmg pool extend --pool=${puuid} --ranks=${rank1},${rank2}...

The pool extend command accepts 2 required parameters:

* The pool UUID of the pool to be extended.
* A comma separated list of server ranks to include in the pool.

The pool rebalance operation will work most efficiently when the pool is extended to its desired size in a single operation, as opposed to multiple, small extensions.

#### Pool Shard Resize

Support for quiescent pool shard resize is currently not supported and is under consideration.

## Pool Catastrophic Recovery

A DAOS pool is instantiated on each target by a set of pmemobj files managed by PMDK and SPDK blobs on SSDs. Tools to verify and repair this persistent data is scheduled for DAOS v2.4 and will be documented here once available.

Meanwhile, PMDK provides a recovery tool (i.e., pmempool check) to verify and possibly repair a pmemobj file. As discussed in the previous section, the rebuild status can be consulted via the pool query and will be expanded with more information.

## Recovering Ownership of a Pool’s Container

Typically users are expected to manage their containers. However, in the event that a container is orphaned and no users have the privileges to change the ownership, an administrator can transfer ownership of the container to a new user and/or group.

To change the owner user:

$ dmg cont set-owner --pool <UUID> --cont <UUID> --user <owner-user>

To change the owner group:

$ dmg cont set-owner --pool <UUID> --cont <UUID> --group <owner-group>

The user and group names are case sensitive and must be formatted as [DAOS ACL user/group principals](https://daos-stack.github.io/overview/security/#principal).

Because this is an administrative action, it does not require the administrator to have any privileges assigned in the container ACL.