

Querying ZFS Storage Pool Status

The `zpool list` command provides several ways to request information regarding pool status. The information available generally falls into three categories: basic usage information, I/O statistics, and health status. All three types of storage pool information are covered in this section.

- [Displaying Information About ZFS Storage Pools](#)
- [Viewing I/O Statistics for ZFS Storage Pools](#)
- [Determining the Health Status of ZFS Storage Pools](#)

Displaying Information About ZFS Storage Pools

You can use the `zpool list` command to display basic information about pools.

Listing Information About All Storage Pools or a Specific Pool

With no arguments, the `zpool list` command displays the following information for all pools on the system:

```
# zpool list
```

NAME	SIZE	ALLOC	FREE	CAP	HEALTH	ALROOT
tank	80.0G	22.3G	47.7G	28%	ONLINE	-
dozer	1.2T	384G	816G	32%	ONLINE	-

This command output displays the following information:

NAME

The name of the pool.

SIZE

The total size of the pool, equal to the sum of the sizes of all top-level virtual devices.

ALLOC

The amount of physical space allocated to all datasets and internal metadata. Note that this amount differs from the amount of disk space as reported at the file system level.

For more information about determining available file system space, see [ZFS Disk Space Accounting](#).

FREE

The amount of unallocated space in the pool.

CAP (CAPACITY)

The amount of disk space used, expressed as a percentage of the total disk space.

HEALTH

The current health status of the pool.

For more information about pool health, see [Determining the Health Status of ZFS Storage Pools](#).

ALTROOT

The alternate root of the pool, if one exists.

For more information about alternate root pools, see [Using ZFS Alternate Root Pools](#).

You can also gather statistics for a specific pool by specifying the pool name. For example:

```
# zpool list tank
NAME                SIZE    ALLOC  FREE   CAP  HEALTH  ALTROOT
tank                80.0G   22.3G  47.7G  28%  ONLINE  -
```

Listing Specific Storage Pool Statistics

Specific statistics can be requested by using the `-o` option. This option provides custom reports or a quick way to list pertinent information. For example, to list only the name and size of each pool, you use the following syntax:

```
# zpool list -o name,size
NAME                SIZE
tank                80.0G
dozer               1.2T
```

The column names correspond to the properties that are listed in [Listing Information About All Storage Pools or a Specific Pool](#).

Scripting ZFS Storage Pool Output

The default output for the `zpool list` command is designed for readability and is not easy to use as part of a shell script. To aid programmatic uses of the command, the `-H` option can be used to suppress the column headings and separate fields by tabs, rather than by spaces. For example, to request a list of all pool names on the system, you would use the following syntax:

```
# zpool list -Ho name
tank
dozer
```

Here is another example:

```
# zpool list -H -o name,size
tank    80.0G
dozer   1.2T
```

Displaying ZFS Storage Pool Command History

ZFS automatically logs successful `zfs` and `zpool` commands that modify pool state information. This information can be displayed by using the `zpool history` command.

For example, the following syntax displays the command output for the root pool:

```
# zpool history
History for 'rpool':
2010-05-11.10:18:54 zpool create -f -o failmode=continue -R /a -m legacy -o
cachefile=/tmp/root/etc/zfs/zpool.cache rpool mirror c1t0d0s0 c1t1d0s0
2010-05-11.10:18:55 zfs set canmount=noauto rpool
2010-05-11.10:18:55 zfs set mountpoint=/rpool rpool
2010-05-11.10:18:56 zfs create -o mountpoint=legacy rpool/ROOT
2010-05-11.10:18:57 zfs create -b 8192 -V 2048m rpool/swap
2010-05-11.10:18:58 zfs create -b 131072 -V 1536m rpool/dump
2010-05-11.10:19:01 zfs create -o canmount=noauto rpool/ROOT/zfsBE
```

```
2010-05-11.10:19:02 zpool set bootfs=rpool/ROOT/zfsBE rpool
2010-05-11.10:19:02 zfs set mountpoint=/ rpool/ROOT/zfsBE
2010-05-11.10:19:03 zfs set canmount=on rpool
2010-05-11.10:19:04 zfs create -o mountpoint=/export rpool/export
2010-05-11.10:19:05 zfs create rpool/export/home
2010-05-11.11:11:10 zpool set bootfs=rpool rpool
2010-05-11.11:11:10 zpool set bootfs=rpool/ROOT/zfsBE rpool
```

You can use similar output on your system to identify the **actual** ZFS commands that were executed to troubleshoot an error condition.

The features of the history log are as follows:

- The log cannot be disabled.
- The log is saved persistently on disk, which means that the log is saved across system reboots.
- The log is implemented as a ring buffer. The minimum size is 128 KB. The maximum size is 32 MB.
- For smaller pools, the maximum size is capped at 1 percent of the pool size, where the size is determined at pool creation time.
- The log requires no administration, which means that tuning the size of the log or changing the location of the log is unnecessary.

To identify the command history of a specific storage pool, use syntax similar to the following:

```
# zpool history tank
History for 'tank':
2010-05-13.14:13:15 zpool create tank mirror c1t2d0 c1t3d0
2010-05-13.14:21:19 zfs create tank/snaps
2010-05-14.08:10:29 zfs create tank/ws01
2010-05-14.08:10:54 zfs snapshot tank/ws01@now
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix
```

Use the `-l` option to display a long format that includes the user name, the host name, and the zone in which the operation was performed. For example:

```
# zpool history -l tank
History for 'tank':
2010-05-13.14:13:15 zpool create tank mirror c1t2d0 c1t3d0 [user root on neo]
2010-05-13.14:21:19 zfs create tank/snaps [user root on neo]
2010-05-14.08:10:29 zfs create tank/ws01 [user root on neo]
2010-05-14.08:10:54 zfs snapshot tank/ws01@now [user root on neo]
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix [user root on neo]
```

Use the `-i` option to display internal event information that can be used for diagnostic purposes. For example:

```
# zpool history -i tank
2010-05-13.14:13:15 zpool create -f tank mirror c1t2d0 c1t23d0
2010-05-13.14:13:45 [internal pool create txg:6] pool spa 19; zfs spa 19; zpl 4;...
2010-05-13.14:21:19 zfs create tank/snaps
2010-05-13.14:22:02 [internal replay_inc_sync txg:20451] dataset = 41
2010-05-13.14:25:25 [internal snapshot txg:20480] dataset = 52
2010-05-13.14:25:25 [internal destroy_begin_sync txg:20481] dataset = 41
2010-05-13.14:25:26 [internal destroy txg:20488] dataset = 41
2010-05-13.14:25:26 [internal reservation set txg:20488] 0 dataset = 0
2010-05-14.08:10:29 zfs create tank/ws01
2010-05-14.08:10:54 [internal snapshot txg:53992] dataset = 42
2010-05-14.08:10:54 zfs snapshot tank/ws01@now
2010-05-14.08:11:04 [internal create txg:53994] dataset = 58
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix
```

Viewing I/O Statistics for ZFS Storage Pools

To request I/O statistics for a pool or specific virtual devices, use the `zpool iostat` command. Similar to the `iostat` command, this command can display a static snapshot of all I/O activity, as well as updated statistics for every specified interval. The following statistics are reported:

`alloc capacity`

The amount of data currently stored in the pool or device. This amount differs from the amount of disk space available to actual file systems by a small margin due to internal implementation details.

For more information about the differences between pool space and dataset space, see [ZFS Disk Space Accounting](#).

`free capacity`

The amount of disk space available in the pool or device. As with the used statistic, this amount differs from the amount of disk space available to datasets by a small margin.

`read operations`

The number of read I/O operations sent to the pool or device, including metadata requests.

`write operations`

The number of write I/O operations sent to the pool or device.

`read bandwidth`

The bandwidth of all read operations (including metadata), expressed as units per second.

`write bandwidth`

The bandwidth of all write operations, expressed as units per second.

Listing Pool-Wide I/O Statistics

With no options, the `zpool iostat` command displays the accumulated statistics since boot for all pools on the system. For example:

```
# zpool iostat
          capacity      operations      bandwidth
pool      alloc  free   read  write   read  write
-----
rpool     6.05G 61.9G    0     0    786   107
tank      31.3G 36.7G    4     1 296K 86.1K
-----
```

Because these statistics are cumulative since boot, bandwidth might appear low if the pool is relatively idle. You can request a more accurate view of current bandwidth usage by specifying an interval. For example:

```
# zpool iostat tank 2
          capacity      operations      bandwidth
pool      alloc  free   read  write   read  write
-----
tank      18.5G 49.5G    0   187    0 23.3M
tank      18.5G 49.5G    0   464    0 57.7M
tank      18.5G 49.5G    0   457    0 56.6M
tank      18.8G 49.2G    0   435    0 51.3M
```

In this example, the command displays usage statistics for the pool tank every two seconds until you type Control-C. Alternately, you can specify an additional `count` argument, which causes the command to terminate after the specified number of iterations. For example, `zpool iostat 2 3` would print a summary every two seconds for three iterations, for a total of six seconds. If there is only a single pool, then the statistics are displayed on consecutive lines. If more than one pool exists, then an additional dashed line delineates each iteration to provide visual separation.

Listing Virtual Device I/O Statistics

In addition to pool-wide I/O statistics, the `zpool iostat` command can display I/O statistics for virtual devices. This command can be used to identify abnormally slow devices or to observe the distribution of I/O generated by ZFS. To request the complete virtual device layout as well as all I/O statistics, use the `zpool iostat -v` command. For example:

```
# zpool iostat -v
          capacity      operations      bandwidth
pool      alloc  free  read  write  read  write
-----  -
rpool     6.05G  61.9G    0     0    785   107
  mirror  6.05G  61.9G    0     0    785   107
    c1t0d0s0  -     -     0     0    578   109
    c1t1d0s0  -     -     0     0    595   109
-----  -
tank      36.5G  31.5G     4     1   295K  146K
  mirror  36.5G  31.5G   126    45  8.13M  4.01M
    c1t2d0  -     -     0     3   100K  386K
    c1t3d0  -     -     0     3   104K  386K
-----  -
```

Note two important points when viewing I/O statistics for virtual devices:

- First, disk space usage statistics are only available for top-level virtual devices. The way in which disk space is allocated among mirror and RAID-Z virtual devices is particular to the implementation and not easily expressed as a single number.
- Second, the numbers might not add up exactly as you would expect them to. In particular, operations across RAID-Z and mirrored devices will not be exactly equal. This difference is particularly noticeable immediately after a pool is created, as a significant amount of I/O is done directly to the disks as part of pool creation, which is not accounted for at the mirror level. Over time, these numbers gradually equalize. However, broken, unresponsive, or offline devices can affect this symmetry as well.

You can use the same set of options (interval and count) when examining virtual device statistics.

Determining the Health Status of ZFS Storage Pools

ZFS provides an integrated method of examining pool and device health. The health of a pool is determined from the state of all its devices. This state information is displayed by using the `zpool status` command. In addition, potential pool and device failures are reported by `fmfd`, displayed on the system console, and logged in the `/var/adm/messages` file.

This section describes how to determine pool and device health. This chapter does not document how to repair or recover from unhealthy pools. For more information about troubleshooting and data recovery, see [Chapter 11, Oracle Solaris ZFS Troubleshooting and Pool Recovery](#).

Each device can fall into one of the following states:

ONLINE

The device or virtual device is in normal working order. Although some transient errors might still occur, the device is otherwise in working order.

DEGRADED

The virtual device has experienced a failure but can still function. This state is most common when a mirror or RAID-Z device has lost one or more constituent devices. The fault tolerance of the pool might be compromised, as a subsequent fault in another device might be unrecoverable.

FAULTED

The device or virtual device is completely inaccessible. This status typically indicates total failure of the device, such that ZFS is incapable of sending data to it or receiving data from it. If a top-level virtual device is in this state, then the pool is completely inaccessible.

OFFLINE

The device has been explicitly taken offline by the administrator.

UNAVAIL

The device or virtual device cannot be opened. In some cases, pools with `UNAVAIL` devices appear in `DEGRADED` mode. If a top-level virtual device is `UNAVAIL`, then nothing in the pool can be accessed.

REMOVED

The device was physically removed while the system was running. Device removal detection is hardware-dependent and might not be supported on all platforms.

The health of a pool is determined from the health of all its top-level virtual devices. If all virtual devices are `ONLINE`, then the pool is also `ONLINE`. If any one of the virtual devices is `DEGRADED` or `UNAVAIL`, then the pool is also `DEGRADED`. If a top-level virtual device is `FAULTED` or `OFFLINE`, then the pool is also `FAULTED`. A pool in the `FAULTED` state is completely inaccessible. No data can be recovered until the necessary devices are attached or repaired. A pool in the `DEGRADED` state continues to run, but you might not achieve the same level of data redundancy or data throughput than if the pool were online.

Basic Storage Pool Health Status

You can quickly review pool health status by using the `zpool status` command as follows:

```
# zpool status -x
all pools are healthy
```

Specific pools can be examined by specifying a pool name in the command syntax. Any pool that is not in the `ONLINE` state should be investigated for potential problems, as described in the next section.

Detailed Health Status

You can request a more detailed health summary status by using the `-v` option. For example:

```
# zpool status -v tank
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: scrub completed after 0h0m with 0 errors on Wed Jan 20 15:13:59 2010
config:

NAME          STATE          READ WRITE CKSUM
tank          DEGRADED       0    0    0
  mirror-0    DEGRADED       0    0    0
    clt0d0    ONLINE        0    0    0
    clt1d0    UNAVAIL       0    0    0  cannot open

errors: No known data errors
```

This output displays a complete description of why the pool is in its current state, including a readable description of the problem and a link to a knowledge article for more information. Each knowledge article provides up-to-date information about the best way to recover from your current problem. Using the detailed configuration information, you can determine which device is damaged and how to repair the pool.

In the preceding example, the faulted device should be replaced. After the device is replaced, use the `zpool online` command to bring the device online. For example:

```
# zpool online tank c1t0d0
Bringing device c1t0d0 online
# zpool status -x
all pools are healthy
```

If the `autoreplace` property is on, you might not have to online the replaced device.

If a pool has an offline device, the command output identifies the problem pool. For example:

```
# zpool status -x
pool: tank
state: DEGRADED
status: One or more devices has been taken offline by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 15:15:09 2010
config:

    NAME          STATE          READ WRITE CKSUM
    tank          DEGRADED       0     0     0
      mirror-0    DEGRADED       0     0     0
        c1t0d0    ONLINE        0     0     0
        c1t1d0    OFFLINE        0     0     0 48K resilvered

errors: No known data errors
```

The `READ` and `WRITE` columns provide a count of I/O errors that occurred on the device, while the `CKSUM` column provides a count of uncorrectable checksum errors that occurred on the device. Both error counts indicate a potential device failure, and some corrective action is needed. If non-zero errors are reported for a top-level virtual device, portions of your data might have become inaccessible.

The `errors:` field identifies any known data errors.

In the preceding example output, the offline device is not causing data errors.

For more information about diagnosing and repairing faulted pools and data, see [Chapter 11, Oracle Solaris ZFS Troubleshooting and Pool Recovery](#).

[Original Article](#)

Revision #1

Created 2023-11-10 05:30:44 UTC by Ryan

Updated 2025-02-12 01:12:50 UTC by Ryan