

NAME

zfs-destroy - destroy ZFS dataset, snapshots, or bookmark

SYNOPSIS

zfs destroy [-Rfnprv] *filesystem|volume*

zfs destroy [-Rdnprv] *filesystem|volume@snap[%snap[,snap[%snap]]]<?>*

zfs destroy *filesystem|volume#bookmark*

DESCRIPTION

zfs destroy [-Rfnprv] *filesystem|volume*

Destroys the given dataset. By default, the command unshares any file systems that are currently shared, unmounts any file systems that are currently mounted, and refuses to destroy a dataset that has active dependents (children or clones).

- R Recursively destroy all dependents, including cloned file systems outside the target hierarchy.
- f Forcibly unmount file systems. This option has no effect on non-file systems or unmounted file systems.
- n Do a dry-run ("No-op") deletion. No data will be deleted. This is useful in conjunction with the -v or -p flags to determine what data would be deleted.
- p Print machine-parsable verbose information about the deleted data.
- r Recursively destroy all children.
- v Print verbose information about the deleted data.

Extreme care should be taken when applying either the -r or the -R options, as they can destroy large portions of a pool and cause unexpected behavior for mounted file systems in use.

zfs destroy [-Rdnprv] *filesystem|volume@snap[%snap[,snap[%snap]]]<?>*

The given snapshots are destroyed immediately if and only if the **zfs destroy** command without the -d option would have destroyed it. Such immediate destruction would occur, for example, if the snapshot had no clones and the user-initiated reference count were zero.

If a snapshot does not qualify for immediate destruction, it is marked for deferred deletion. In this state, it exists as a usable, visible snapshot until both of the preconditions listed above are met, at which point it is destroyed.

An inclusive range of snapshots may be specified by separating the first and last snapshots with a percent sign. The first and/or last snapshots may be left blank, in which case the filesystem's oldest or newest snapshot will be implied.

Multiple snapshots (or ranges of snapshots) of the same filesystem or volume may be specified in a comma-separated list of snapshots. Only the snapshot's short name (the part after the @) should be specified when using a range or comma-separated list to identify multiple snapshots.

- R** Recursively destroy all clones of these snapshots, including the clones, snapshots, and children. If this flag is specified, the **-d** flag will have no effect.
- d** Destroy immediately. If a snapshot cannot be destroyed now, mark it for deferred destruction.
- n** Do a dry-run ("No-op") deletion. No data will be deleted. This is useful in conjunction with the **-p** or **-v** flags to determine what data would be deleted.
- p** Print machine-parsable verbose information about the deleted data.
- r** Destroy (or mark for deferred deletion) all snapshots with this name in descendent file systems.
- v** Print verbose information about the deleted data.

Extreme care should be taken when applying either the **-r** or the **-R** options, as they can destroy large portions of a pool and cause unexpected behavior for mounted file systems in use.

zfs destroy *filesystem|volume#bookmark*

The given bookmark is destroyed.

EXAMPLES

Example 1: Creating and Destroying Multiple Snapshots

The following command creates snapshots named *yesterday* of *pool/home* and all of its descendent file systems. Each snapshot is mounted on demand in the *.zfs/snapshot* directory at the root of its file system. The second command destroys the newly created snapshots.

```
# zfs snapshot -r pool/home@yesterday
# zfs destroy -r pool/home@yesterday
```

Example 2: Promoting a ZFS Clone

The following commands illustrate how to test out changes to a file system, and then replace the original file system with the changed one, using clones, clone promotion, and renaming:

```
# zfs create pool/project/production
```

```
    populate /pool/project/production with data
# zfs snapshot pool/project/production@today
# zfs clone pool/project/production@today pool/project/beta
    make changes to /pool/project/beta and test them
# zfs promote pool/project/beta
# zfs rename pool/project/production pool/project/legacy
# zfs rename pool/project/beta pool/project/production
    once the legacy version is no longer needed, it can be destroyed
# zfs destroy pool/project/legacy
```

Example 3: Performing a Rolling Snapshot

The following example shows how to maintain a history of snapshots with a consistent naming scheme. To keep a week's worth of snapshots, the user destroys the oldest snapshot, renames the remaining snapshots, and then creates a new snapshot, as follows:

```
# zfs destroy -r pool/users@7daysago
# zfs rename -r pool/users@6daysago @7daysago
# zfs rename -r pool/users@5daysago @6daysago
# zfs rename -r pool/users@4daysago @5daysago
# zfs rename -r pool/users@3daysago @4daysago
# zfs rename -r pool/users@2daysago @3daysago
# zfs rename -r pool/users@yesterday @2daysago
# zfs rename -r pool/users@today @yesterday
# zfs snapshot -r pool/users@today
```

SEE ALSO

zfs-create(8), zfs-hold(8)