



## > Solaris™ 10 How To Guides

# HOW TO MANAGE SYSTEMS WITH ZFS IN SOLARIS™ CONTAINERS



# About This ZFS How To Guide

This guide is intended to show a new user the capabilities of ZFS when coupled with Solaris Containers. It describes the assignment of a ZFS file system to a zone, and some of the ZFS administrative tasks possible in such a configuration. Assigning a file system to a zone gives a zone administrator full delegated control of the ZFS file system, enabling them to take snapshots, create sub file systems and perform other tasks without requiring the global zone administrator's intervention.

After reading this guide, the user will have an understanding of how to delegate a ZFS file system to a container and to manage it from within the container.

Contributors: Paul Steeves and Solaris ZFS Engineering team

## Contents

<b>Solaris ZFS and Containers: Overview</b>	<b>Page 1</b>
<b>ZFS and Containers: An Example</b>	<b>Page 2 &gt; 8</b>
Creating a zpool	Page 2
Creating a Zone	Page 2
Allocating a ZFS File System to a Zone	Page 4
Creating New File Systems	Page 5
Applying Quota to the File Systems	Page 6
Changing the Mountpoint of a File System	Page 6
Setting the Compression Property	Page 7
Taking a Snapshot	Page 8
<b>Summary</b>	<b>Page 8</b>
<b>For More Information</b>	<b>Page 9</b>

# ZFS How To Guide

## Solaris ZFS and Containers: Overview

ZFS is a new file system from Sun provided in the Solaris 10 OS. It provides very high levels of data integrity and performance, and improves the ease of file system management by eliminating the need for a volume manager. Also, because it is 128-bit based, it opens the door to virtually unlimited data scalability.

Traditional file systems have a model of a single device or volume for each file system. This makes administration complex and potentially prone to administrative errors which could result in the destruction of data or lack of access to the data. ZFS uses a pooled storage model in which storage capacity is held within an expandable pool called a *zpool*. Multiple file systems can exist within a *zpool* and they can share the capacity and IO resources of the pool. This new architectural approach, the storage version of virtual memory, permits easier administration of file systems and enables performance breakthroughs.

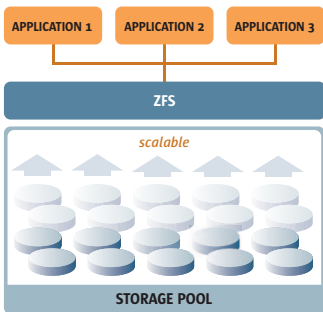


Figure 1—ZFS automatically grows shared storage

Solaris Containers is Sun's operating system virtualization technique. It consists of several technologies that work together to foster improved resource management and isolate the environment from the underlying OS. With Solaris Containers, you can determine how to allocate resources among applications and services, and ensure they do not interfere with one another.

ZFS and Containers are tightly integrated to allow the zone administrator delegated rights to control the ZFS file systems. By using ZFS and Containers in combination you can assign a portion of the storage pool to a container. This means that the administrator for that container has the authority to manage that portion of the storage pool, but it is isolated from the rest of the system. (See the figure below.) This lets you benefit from both the data integrity of ZFS, as well as the high availability and resource control of containers.

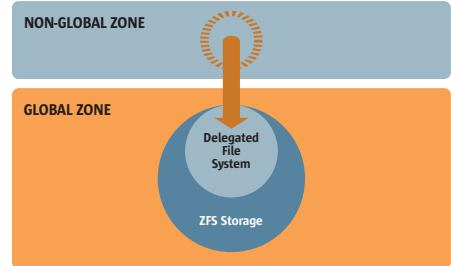


Figure 2—Containers let you isolate ZFS file systems

Note that while Containers technology has several aspects, the example in this guide will focus solely on the zones aspect.

## ZFS and Containers: An Example

This section is a step-by-step guide that shows how to perform certain ZFS file system tasks inside of Solaris Containers; for example, taking snapshots and managing data compression. It does this by going through the following steps:

- Creating a zpool
- Creating a Zone
- Allocating a ZFS File System to a Zone
- Creating New File Systems
- Applying Quota to the File Systems
- Changing the Mountpoint of a File System
- Setting the Compression Property
- Taking a Snapshot

Each of these steps is described in detail below.

### Creating a zpool

ZFS uses device names or partition names when dealing with pools and devices. For a device, this will be something like `c1t0d0` (for a SCSI device) or `c1d1` (for an IDE device). For a partition, it will be something like `c1t0dos0` (for a SCSI device) or `c1d1s0` (for an IDE device). This example creates a pool that is mirrored using two disks.

1. To create a zpool in the global zone, use the `zpool create` command. Typically, you use two devices to provide redundancy.

```
Global# zpool create mypool mirror c2t5d0 c2t6d0
```

Note that the `zpool create` command may fail if the devices are in use or contain some types of existing data (e.g. UFS file system). If they are in use, you will need to unmount them or otherwise stop using them. If they contain existing data you can use the `-f` (force) flag to override the safety check, but be sure that you are not destroying any data you want to retain.

2. Examine the pool properties using the `zpool list` command.

```
Global# zpool list
```

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
mypool	199G	164K	199G	0%	ONLINE	-

This shows you that there is one zpool, named `mypool`, with a capacity of 199GBytes.

### Creating a Zone

To show ZFS working in an environment that is isolated from other applications on the system, you need to create a zone. To create a zone:

1. Create a directory where the zone file system will reside, using the `mkdir` command. Be sure to chose a location where the file system has at least 80MBytes of available space.

```
Global# mkdir /zones
```

Note that in this example, for the sake of brevity, the root file system of the zone is a UFS file system.

2. Configure the zone (myzone), using the zonecfg command, and specify the location of the zone's files (/zones/myzone). Use the following series of commands.

```
Global# zonecfg -z myzone
myzone: No such zone configured
      Use 'create' to begin configuring a new zone
zonecfg:myzone> create
zonecfg:myzone> set zonepath=/zones/myzone
zonecfg:myzone> verify
zonecfg:myzone> commit
zonecfg:myzone> exit
```

Again, for the purpose of streamlining, this example uses a very minimal zone. For more details on creating zones see the Solaris Containers How To Do Stuff guide at:

[www.sun.com/software/solaris/howtoguides/containersLowRes.jsp](http://www.sun.com/software/solaris/howtoguides/containersLowRes.jsp)

3. Install the zone by using the zoneadm.

```
Global# zoneadm -z myzone install
Preparing to install zone <myzone>
      [Output from zoneadm, this may take a few mins]
```

4. Boot the zone to complete the installation, using the zoneadm command.

```
Global# zoneadm -z myzone boot
```

5. Use the zlogin command to connect to the zone console.

```
Global# zlogin -C myzone
      [Connected to zone 'myzone' console]
      [Initial zone boot output, service descriptions are loaded etc.]
```

It may take a short while for the first boot to configure everything, load all the service descriptors, and so on. You will need to answer the system configuration details. Some suggestions are:

```
Terminal=(12)X Terminal Emulator (xterms)
Not Networked
No Kerberos
Name service = None
Time Zone = your-time-zone
root passwd = (Your choice - remember it though!)
```

The zone will reboot after you have provided the configuration information.

6. Before you can proceed to the next stage, the configured zone needs to be shutdown (configuration changes are only applied when the zone boots).

```
Global# zlogin myzone init 5
```

## Allocating a ZFS File System to a Zone

Now that you have a zpool (mypool) and a zone (myzone) you are ready to allocate a ZFS file system to the zone.

1. To create a ZFS file system, use the `zfs create` command.

```
Global# zfs create mypool/myzonefs
```

2. To apply a quota to the file system, use the `zfs set quota` command.

```
Global# zfs set quota=5G mypool/myzonefs
```

The file system and all of its child file systems can be no larger than the designated quota. Note that both these steps must be performed in the global zone. Also notice that creating the file system in ZFS is much simpler than with a traditional file system/volume manager combination.

To illustrate the isolation/security properties of containers with ZFS this example now creates a ZFS file system that will remain outside the container. There is no need to apply a quota to this outside file system.

3. To create this other file system, again use the `zfs create` command.

```
Global# zfs create mypool/myfs
```

4. To show the properties of all the pool and the file systems, use the `ZFS list` command.

```
Global# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
mypool	396G	197G	99.5K	/mypool
mypool/myfs	98.5K	197G	98.5K	/mypool/myfs
mypool/myzonefs	98.5K	5G	98.5K	/mypool/myzonefs

To make the file system (myzonefs) available in the zone (myzone), the zone configuration needs to be updated.

5. To update the zone configuration, use the `zonecfg` command.

```
Global# zonecfg -z myzone
zonecfg:myzone> add dataset
zonecfg:myzone:dataset> set name=mypool/myzonefs
zonecfg:myzone:dataset> end
zonecfg:myzone> commit
zonecfg:myzone> exit
```

The `mypool/myzonefs` file system is now added to the zone configuration. Note that these steps must be performed with the zone shut down, otherwise the zone configuration changes would not be visible until the next reboot. To check that the zone is shut down try logging into it using `zlogin myzone`. If the zone is shut down the login will fail; if the zone is running you will see a login prompt—login as root and shut the zone down with `init 5`. These steps are performed in the global zone.

- Now boot the zone.

```
Global# zoneadm -z myzone boot
```

- Log in to the zone. (Leave a few seconds for the zone to boot.)

```
Global# zlogin -C myzone
[Connected to zone 'myzone' pts/3]
[Usual Solaris login sequence]
```

- List the ZFS file systems in the zone.

NAME	USED	AVAIL	REFER	MOUNTPOINT
mypool	0M	200B	-	/mypool
mypool/myzonefs	8K	5G	8K	/mypool/myzonefs

Note the 5GByte maximum available from the external quota and that the other file systems in the pool (mypool/myfs) are not visible. This demonstrates the isolation property that Containers provide.

## Creating New File Systems

Administering ZFS file systems from the non-global zone is done just like it is in the global zone, although you are limited to operating within the file system that is allocated to the zone (mypool/myzonefs). New ZFS file systems are always created as a child of this file system because this is the only ZFS file system the non-global zone can see. The administrator in the non-global zone can create these file systems. There is no need to involve administrator of the global zone, though the administrator could do so if it were necessary.

- To create a new file system, use the `zfs create` command.

```
MyZone# zfs create mypool/myzonefs/tim
MyZone# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
mypool	504M	197G	99K	/mypool
mypool/myzonefs	197K	5.00G	98.5K	/mypool/myzonefs
mypool/myzonefs/tim	98.5K	5.00G	98.5K	/mypool/myzonefs/tim

The non-global zone administrator can create as many child file systems as s/he wants and each child file system can have its own file systems, and in that way form a hierarchy.

As a demonstration that the non-global zone administrator is limited to the assigned file systems, this example demonstrates trying to break security by creating a file system outside the container's "space".

- Try to create another file system outside of mypool/myzonefs, using the `zfs create` command.

```
MyZone# zfs create mypool/myzonefs1
cannot create 'mypool/myzonefs1': permission denied
```

As you can see, ZFS and zones security denies permission for the non-global zone to access resources it has not been allocated and the operation fails.

## Applying Quota to the File Systems

Typically, to prevent the user consuming all of the space, a non-global zone administrator will want to apply a quota to the new file system. Of course, the child's quota can't be more than 5GByte as that's the quota specified by the global zone administrator to all of the file systems below mypool/myzonefs.

1. To set a quota on our new file system, use the `zfs set quota` command.

```
MyZone# zfs set quota=1G mypool/myzonefs/tim
MyZone# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
mypool	508M	197G	99K	/mypool
mypool/myzonefs	198k	5.00G	99K	/mypool/myzonefs
mypool/myzonefs/tim	98.5K	1024M	98.5K	/mypool/myzonefs/tim

The administrator of the non-global zone has set the quota of the child file system to be 1G. They have full authority to do this because they are operating on their delegated resources and do not need to involve the global zone administrator.

The ZFS property inheritance mechanism applies across zone boundaries, so the non-global zone administrator can specify his/her own property values should s/he wish to do so. As with normal ZFS property inheritance, these override inherited values.

## Changing the Mountpoint of a File System

Now that the file system is set up and has the correct quota assigned to it, it is ready for use. However, the place where the file system appears (the mountpoint) is partially dictated by what the global zone administrator initially chose as the pool name (in this example, mypool/myzonefs). But typically, a non-global zone administrator would want to change it.

1. To change the mountpoint, use the `zfs set mountpoint` command.

```
MyZone# zfs set mountpoint=/export/home/tim mypool/myzonefs/tim
MyZone# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
mypool	508M	197G	99K	/mypool
mypool/myzonefs	198k	5.00G	99K	/mypool/myzonefs
mypool/myzonefs/tim	98.5K	1024M	98.5K	/export/home/tim

Note that the mountpoint can be changed for any file system independently.



## Setting the Compression Property

The next example demonstrates the compression property. If compression is enabled, ZFS will transparently compress all of the data before it is written to disk.

The benefits of compression are both saved disk space and possible write speed improvements.

1. To see what the current compression setting is, use the `zfs get` command.

```
MyZone# zfs get compression mypool mypool/myzonefs mypool/myzonefs/tim
```

NAME	PROPERTY	VALUE	SOURCE
mypool	compression	off	default
mypool/myzonefs	compression	off	default
mypool/myzonefs/tim	compression	off	default

Be aware that the compression property on the pool is inherited by the file system and child file system. So if the non-global zone administrator sets the compression property for the delegated file system, it will set it for everything below, as well.

2. To set the compression for the file system, use the `zfs set` command.

```
MyZone# zfs set compression=on mypool/myzonefs
```

3. Examine the compression property again in the non-global zone.

```
MyZone# zfs get compression mypool mypool/myzonefs mypool/myzonefs/tim
```

NAME	PROPERTY	VALUE	SOURCE
mypool	compression	off	default
mypool/myzonefs	compression	on	local
mypool/myzonefs/tim	compression	on	Inherited from mypool/myzonefs

Note the compression property has been inherited by `mypool/myzonefs/tim` as with normal ZFS administration.

## Taking a Snapshot

One of the major advantages of ZFS is the ability to create an instant snapshot of any file system. By delegating a file system to a non-global zone this feature becomes available as an option for the non-global zone administrator.

1. To take a snapshot named “1st “ of the file system, use the `zfs snapshot` command.

```
MyZone# zfs snapshot mypool/myzonefs@1st
MyZone# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
mypool	512K	197G	99K	/mypool
mypool/myzonefs	198K	5.00G	99K	/mypool/myzonefs
mypool/myzonefs@1st	0K	-	99K	-
mypool/myzonefs/tim	98.5K	1024M	98.5K	/export/home/tim

As with ZFS file systems in the global zone, this snapshot is now accessible from the root of the file system in `.zfs/snapshot/1st`.

## Summary

Once a zone has been created and a ZFS file system has been allocated to it, the administrator for that (non-global) zone can create file systems, take snapshots, create clones, and perform all the other functions of an administrator—within that zone. Yet the global zone, and any other zones, are fully isolated from whatever happens in that zone.

The integration of Solaris Containers and Solaris ZFS is just another way that the Solaris 10 OS is providing cost benefits to customers by allowing them to safely consolidate applications and more easily manage the data those applications use.

## For More information

While this Solaris How To Guide provides the rudimentary steps needed to get Solaris Container technology and the Solaris ZFS file system working in collaboration with each other, more sophisticated configurations are possible. For more information regarding Solaris Containers and Solaris ZFS, visit [sun.com/solaris](http://sun.com/solaris).

Manuals	
Solaris ZFS Administration Guide	<a href="http://opensolaris.org/os/community/zfs/docs/zfsadmin_0417.pdf">opensolaris.org/os/community/zfs/docs/zfsadmin_0417.pdf</a>
Web-Based Training	
Solaris ZFS Learning Center (Including video presentations and demos)	<a href="http://sun.com/solaris/zfs">sun.com/solaris/zfs</a>
Solaris Containers Learning Center	<a href="http://sun.com/solaris/containers">sun.com/solaris/containers</a>
Solaris Containers How to Guide	<a href="http://sun.com/solaris/howtoguides/containers">sun.com/solaris/howtoguides/containers</a>
Frequently Asked Questions (FAQ)	
ZFS FAQ	<a href="http://sun.com/solaris/zfs/faq">sun.com/solaris/zfs/faq</a>
Additional Q&A	<a href="http://opensolaris.org/os/community/zfs/faq">opensolaris.org/os/community/zfs/faq</a>
Community Resources	
OpenSolaris ZFS Community	<a href="http://opensolaris.org/os/community/zfs">opensolaris.org/os/community/zfs</a>
Sun ZFS Blogs	<a href="http://opensolaris.org/os/community/zfs/blogs">opensolaris.org/os/community/zfs/blogs</a>

[sun.com/solaris](http://sun.com/solaris)

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web [sun.com](http://sun.com)



©2005 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.