NAME

clzonecluster, clzc − create and manage zone clusters

SYNOPSIS

/usr/cluster/bin/clzonecluster [*subcommand*] **-?**

/usr/cluster/bin/clzonecluster -V

/usr/cluster/bin/clzonecluster *subcommand* [*options*] **-v** [*zoneclustername*]

/usr/cluster/bin/clzonecluster boot [**-n** *nodename*[,...]] {+ | *zoneclustername* [...]}

/usr/cluster/bin/clzonecluster clone -Z *source-zoneclustername* [**-m** *method*] [**-n** *nodename*[,...]] { *zoneclustername* }

/usr/cluster/bin/clzonecluster configure [**-f** *commandfile*] *zoneclustername*

/usr/cluster/bin/clzonecluster delete [**-F**] *zoneclustername*

/usr/cluster/bin/clzonecluster halt [**-n** *nodename*[,...]] {+ | *zoneclustername*}

/usr/cluster/bin/clzonecluster install [**-n** *nodename*[,...]] *zoneclustername*

/usr/cluster/bin/clzonecluster list [+ | *zoneclustername* [...]]

/usr/cluster/bin/clzonecluster move -f *zonepath zoneclustername*

/usr/cluster/bin/clzonecluster ready [**-n** *nodename*[,...]] {+ | *zoneclustername* [...]}

/usr/cluster/bin/clzonecluster reboot [**-n** *nodename*[,...]] {+ | *zoneclustername* [...]}

/usr/cluster/bin/clzonecluster show [+ | *zoneclustername* [...]]

/usr/cluster/bin/clzonecluster status [+ | *zoneclustername* [...]]

/usr/cluster/bin/clzonecluster uninstall [**-F**] [**-n** *nodename*[,...]] *zoneclustername*

/usr/cluster/bin/clzonecluster verify [**-n** *nodename*[,...]] {+ | *zoneclustername* [...]}

DESCRIPTION

The **clzonecluster** command creates and modifies zone clusters for Sun Cluster configurations. The **clzc** command is the short form of the **clzonecluster** command; the commands are identical. The **clzonecluster** command is cluster-aware and supports a single source of administration. You can issue all forms of the command from one node to affect a single zone-cluster node or all nodes.

You can omit *subcommand* only if *options* is the **-?** option or the **-V** option.

The subcommands require at least one operand, except for the **list**, **show**, and **status** subcommands. However, many subcommands accept the plus sign operand (+) to apply the subcommand to all applicable objects. The **clzonecluster** commands can be run on any node of a zone cluster and can affect any or all of the zone cluster.

Each option has a long and a short form. Both forms of each option are given with the description of the option in OPTIONS.

SUBCOMMANDS

The following subcommands are supported:

boot

Boots the zone cluster.

The **boot** subcommand boots the zone cluster. The **boot** subcommand uses the **-n** flag to boot the zone cluster for a specified list of nodes. You can use the **boot** subcommand only from a global-cluster node.

clone

Clones the zone cluster.

The **clone** command clones the zone cluster. You can use the **clone** subcommand only from a global-cluster node.

configure

Launches an interactive utility to configure a zone cluster.

The **configure** subcommand uses the **zonecfg** command to configure a zone on each specified machine. The **configure** subcommand lets you specify properties that apply to each node of the zone cluster. These properties have the same meaning as established by the **zonecfg** command for individual zones. The **configure** subcommand supports the configuration of properties that are unknown to the **zonecfg** command.

The **configure** subcommand launches an interactive shell if you do not specify the **-f** option. The **-f** option takes a command file as its argument. The **configure** subcommand uses this file to create or modify zone clusters non-interactively. You can use the **configure** subcommand only from a global-cluster node.

Both the interactive and non-interactive forms of the **configure** command support several subcommands to edit the zone cluster configuration. See zonecfg(1M) for a list of available configuration subcommands.

The interactive **configure** utility enables you to create and modify the configuration of a zone cluster. Zone-cluster configuration consists of a number of resource types and properties. The **configure** utility uses the concept of **scope** to determine where the subcommand applies. There are three levels of scope that are used by the **configure** utility: cluster, resource, and node-specific resource. The default scope is cluster. The following list describes the three levels of scope:

- Cluster scope - Properties that affect the entire zone cluster. If the *zoneclustername* is **sczone**, the interactive shell of the **clzonecluster** command looks similar to the following:

  clzc:sczone>

- Node scope - A special resource scope that is nested inside the node resource scope. Settings inside the node scope affect a specific node in the zone cluster. For example, you can add a net resource to a specific node in the zone cluster. The interactive shell of the **clzonecluster** command looks similar to the following:

  clzc:sczone:node:net>

- Resource scope - Properties that apply to one specific resource. A resource scope prompt has the name of the resource type appended. For example, the interactive shell of the **clzonecluster** command looks similar to the following:

  clzc:sczone:net>

delete

Removes a specific zone cluster.

This subcommand deletes a specific zone cluster. When you use a wild card operand (*), the **delete** command removes the zone clusters that are configured on the global cluster. The zone cluster must be in the configured state before you run the **delete** subcommand.

halt

Stops a zone cluster or a specific node on the zone cluster.

When you specify a specific zone cluster, the **halt** subcommand applies only to that specific zone cluster. You can halt the entire zone cluster or just halt specific nodes of a zone cluster. If you do not specify a zone cluster, the **halt** subcommand applies to all zone clusters. You can also halt all zone clusters on specified machines.

The **halt** subcommand uses the **-n** option to halt zone clusters on specific nodes. By default, the **halt** subcommand stops all zone clusters on all nodes. If you specify the **+** operand in place of a zone name, all the zone clusters are stopped. You can use the **halt** subcommand only from a global-cluster node.

install

Installs a zone cluster.

This subcommand installs a zone cluster. You can use the **install** subcommand only from a global-cluster node.

list

Displays the names of configured zone clusters.

This subcommand reports the names of zone clusters that are configured in the cluster. If you run the **list** subcommand from a global-cluster node, the subcommand displays a list of all the zone clusters in the global cluster. If you run the **list** subcommand from a zone-cluster node, the subcommand displays only the name of the zone cluster. To see the list of nodes where the zone cluster is configured, use the **-v** option.

move

Moves the zonepath to a new zonepath.

This subcommand moves the zonepath to a new zonepath. You can use the **move** subcommand only from a global-cluster node.

ready

Prepares the zone for applications.

This subcommand prepares the zone for running applications. You can use the **ready** subcommand only from a global-cluster node.

reboot

Reboots a zone cluster.

This subcommand reboots the zone cluster and is similar to issuing a **halt** subcommand, followed by a **boot** subcommand. See the **halt** subcommand and the **boot** subcommand for more information.

show

Displays the properties of zone clusters.

Properties for a zone cluster include zone cluster name, brand, IP type, node list, and zonepath. The **show** subcommand runs from a zone cluster but applies only to that particular zone cluster. The zonepath is always / when you use this subcommand from a zone cluster. If zone cluster name is specified, this command applies only for that zone cluster.

status

Determines whether the zone cluster node is a member of the zone cluster.

The zone state can be one of the following: Configured, Installed, Ready, Running, and Shutting Down. When you run the **status** subcommand from a global-cluster zone, the state of all the zone clusters in the global cluster is displayed so you can see the state of your virtual cluster. When you run the **status** subcommand from a zone cluster, the state of only that particular zone cluster is displayed. Use the **zoneadm** command to check zone activity.

uninstall

Uninstalls a zone cluster.

This subcommand uninstalls a zone cluster. The **uninstall** subcommand uses the **zoneadm** command.

verify

Checks that the syntax of the specified information is correct.

This subcommand invokes the **zoneadm verify** command on each node in the zone cluster to ensure that each zone cluster member can be installed safely. For more information, see zoneadm(1M).

OPTIONS

Note -

The short and long form of each option are shown in this section.

The following options are supported:

-?
--help

Displays help information.

You can specify this option with or without a *subcommand*.

If you do not specify a *subcommand*, the list of all available subcommands is displayed.

If you specify a *subcommand*, the usage for that subcommand is displayed.

If you specify this option and other options, the other options are ignored.

-f{*commandfile* | *zonepath*}
--file-argument {*commandfile* | *zonepath*}

When used with the **configure** subcommand, the **-f** option specifies the command file argument. For example, **clzonecluster configure -f** *commandfile*. When used with the **move** subcommand, the **-f** option specifies the *zonepath*.

-F
--force

You can use the **-F** option during **delete** and **uninstall** operations. The **-F** option forcefully suppresses the **Are you sure you want to** *do this operation* [y/n]? questions.

-m *method*
--method *copymethod*

Use the *method* option to clone a zone cluster. The only valid method for cloning is the **copy** command. Before you run the **clone** subcommand, you must halt the source zone cluster.

-n *nodename*[...]
--nodelist *nodename*[,...]

Specifies the node list for the subcommand.

For example, **clzonecluster boot -n** *phys-schost-1, phys-schost-2 zoneclustername*.

-v
---verbose

Displays verbose information on the standard output (**stdout**) .

-V
--version

Displays the version of the command.

If you specify this option with other options, with subcommands, or with operands, they are all ignored. Only the version of the command is displayed. No other processing occurs.

-Z *source-zoneclustername*
--zonecluster *source-zoneclustername*

The zone cluster name that you want to clone.

Use the source zone-cluster name for cloning. The source zone cluster must be halted before you use this subcommand.

RESOURCES AND PROPERTIES

The **clzonecluster** command supports several resources and properties for zone clusters.

Resources

The following lists the resource types that are supported in the resource scope and where to find more information:

capped-cpu

See zonecfg(1M).

capped-memory

See zonecfg(1M).

dataset

See zonecfg(1M). Use this resource to export a ZFS data set to be used in the zone cluster for a highly-available ZFS file system. The exported data set is managed by the Sun Cluster software,

and is not passed down to the individual Solaris zone level when specified in the cluster scope. A data set cannot be shared between zone clusters.

dedicated-cpu

See zonecfg(1M). You can use a fixed number of CPUs that are dedicated to the zone cluster on each node.

device

See zonecfg(1M). You can add a device to only one zone cluster.

fs

See zonecfg(1M). Use this resource to export a file system to be used in the zone cluster. The file system types supported are UFS, Vxfs, single-machine QFS, shared QFS, ZFS (exported as a data set), and loopback file systems.

Highly-available file systems (for example, UFS, Vxfs, and single-machine QFS) are always specified in the cluster context. Sun Cluster manages highly-available file systems, and this information is not passed to the **zonecfg** command.

The administrator can specify a loopback mount in the cluster scope and that loopback mount is done on each zone cluster node. This approach is particularly useful for read-only mounts of common local directories, such as directories that contain executable files. This information is passed to the **zonecfg** command, which does the actual mounts.

Shared QFS, UFS, Vxfs, single-machine QFS, and ZFS are configured in at most one zone cluster.

inherit-pkg-dir

See zonecfg(1M).

net

See zonecfg(1M) for information about net resources.

Any net resource managed by Sun Cluster, such as Logical Host or Shared Address, is specified in the cluster scope. Any net resource managed by an application, such as an Oracle RAC VIP, is specified in the cluster scope. These net resources are not passed to the individual Solaris zone level.

The administrator can specify the Network Interface Card (NIC) to use with the specified IP Address. The system automatically selects a NIC that satisfies the following two requirements:

- The NIC already connects to the same subnet.
- The NIC has been configured for this zone cluster.

node

The node resource performs the following two purposes:

- Identifies a scope level. Any resource specified in a node scope belongs exclusively to this specific node.
- Identifies a node of the zone cluster. The administrator identifies the machine where the zone will run by identifying the global cluster global zone on that machine. The administrator also specifies information identifying network information for reaching this node.

rctl

See zonecfg(1M).

sysid

See **sysidcfg**(4). This resource specifies the system identification parameters for all zones of the zone cluster.

Properties

Each resource type has one or more properties. The following properties are supported for cluster:

(cluster)

zonename

The name of the zone cluster, as well as the name of each zone in the zone cluster.

(cluster)

zonepath

The zonepath of each zone in the zone cluster.

(cluster)

autoboot

See zonecfg(1M).

(cluster)

bootargs

See zonecfg(1M).

(cluster)

limitpriv

See zonecfg(1M).

(cluster)

brand

See zonecfg(1M). Cluster is the only brand type supported.

(cluster)

ip-type

See zonecfg(1M). IP-type is the only value supported.

(cluster)

pool

See zonecfg(1M).

(cluster)

cpu-shares

See zonecfg(1M).

(cluster)

max-lwps

See zonecfg(1M).

(cluster)

max-msg-ids

See zonecfg(1M).

(cluster)

max-sem-ids

See zonecfg(1M).

(cluster)

max-shm-ids

See zonecfg(1M).

(cluster)

max-shm-memory

See zonecfg(1M).

(cluster)

enable_priv_net

When set to true, Sun Cluster private network communication is enabled between the nodes of the zone cluster. The Sun Cluster private hostnames and IP addresses for the zone cluster nodes are automatically generated by the system. Private network is disabled if the value is set to false. The default value is true.

fs

See zonecfg(1M).

inherit pkg-dir

See zonecfg(1M).

net

See zonecfg(1M).

device

See zonecfg(1M).

rctl

See zonecfg(1M).

dataset

See zonecfg(1M).

dedicated-cpu

See zonecfg(1M).

capped-memory

See zonecfg(1M).

capped-cpu

See zonecfg(1M).

node

Includes physical-host, hostname, and net.

- physical-host - This property specifies a global cluster node that will host a zone cluster node.
- hostname - This property specifies the public host name of the zone cluster node on the global cluster node specified by the physical-host property.
- net - This resource specifies a network address and physical interface name for public network communication by the zone cluster node on the global cluster node specified by physical-host.

sysid

See **sysidcfg**(4). Includes root_password, name_service, security_policy, system_locale, timezone, terminal, and nfs4_domain. The administrator can later manually change any **sysidcfg** value following the normal Solaris procedures one node at a time.

- root_password - This property specifies the encrypted value of the common root password for all nodes of the zone cluster. Do not specify a clear text password. Encrypted password string from /etc/shadow must be used. This is a required property.
- name_service - This property specifies the naming service to be used in the zone cluster. It is an optional property, and the setting in the global zone is used by default.
- security_policy - The value is set to none by default.
- system_locale - The value is obtained from the environment of the **clzonecluster** command by default.
- timezone - The time zone to be used in the zone cluster. The global zone setting is used by default.
- terminal - The value is set to xterm by default.
- nfs4_domain - The value is set to dynamic by default.

Examples

In all the examples, the *zoneclustername* is **sczone**. The first global-cluster node is **phys-schost-1** and the second node is **phys-schost-2**. The first zone-cluster node is **zc-host-1** and the second one is **zc-host-2**.

Example 1 Creating a New Zone Cluster

The following example demonstrates how to create a two-node zone cluster comprised of sparse-root zones. The /usr/local is loopback mounted into the zone cluster nodes as /opt/local. Two IP addresses are exported to the zone cluster for use as highly-available IP addresses. A ZFS data set is exported to the zone cluster for use as a highly-available ZFS file system. Memory capping is used to limit the amount of memory that can be used in the zone cluster. The **proc_priocnlt** and **proc_clock**_highres privileges are added to the zone cluster to enable Oracle RAC to run. Default system identification values are used, except for the root password.

A UFS file system is exported to the zone cluster for use as a highly-available file system. It is assumed that the UFS file system is created on a Solaris Volume Manager metadevice.

```
phys-schost-1#clzonecluster configure sczone
sczone: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone> create
clzc:sczone> set zonepath=/zones/sczone
clzc:sczone> set limitpriv="default,proc_priocntl,proc_clock_highres"
clzc:sczone> add sysid
clzc:sczone:sysid> set root_password=xxxxxxxxxxxxx
clzc:sczone:sysid> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=phys-schost-1
clzc:sczone:node> set hostname=zc-host-1
clzc:sczone:node> add net
clzc:sczone:node:net> set address=zc-host-1
clzc:sczone:node:net> set physical=bge0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add node
clzc:sczone:node> set physical-host=phys-schost-2
clzc:sczone:node> set hostname=zc-host-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=zc-host-2
clzc:sczone:node:net> set physical=bge0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add net
clzc:sczone:net> set address=192.168.0.1
clzc:sczone:net> end
clzc:sczone> add net
clzc:sczone:net> set address=192.168.0.2
clzc:sczone:net> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/opt/local
clzc:sczone:fs> set special=/usr/local
clzc:sczone:fs> set type=lofs
clzc:sczone:fs> add options [ro,nodevices]
clzc:sczone:fs> end
clzc:sczone> add dataset
clzc:sczone:dataset> set name=tank/home
clzc:sczone:dataset> end
clzc:sczone> add capped-memory
clzc:sczone:capped-memory> set physical=3G
clzc:sczone:capped-memory> set swap=4G
clzc:sczone:capped-memory> set locked=3G
clzc:sczone:capped-memory> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/data/ha-data
clzc:sczone:fs> set special=/dev/md/ha-set/dsk/d10
clzc:sczone:fs> set raw=/dev/md/ha-set/rdsk/d10
```

```
clzc:sczone:fs> set type=ufs
clzc:sczone:fs> end
clzc:sczone> verify
clzc:sczone> commit
clzc:sczone> exit
```

The zone cluster is now configured. The following commands install and then boot the zone cluster from a global-cluster node:

```
phys-schost-1# clzonecluster install sczone
```

```
phys-schost-1# clzonecluster boot sczone
```

Example 2 Modifying an Existing Zone Cluster

The following example shows how to modify the configuration of the zone cluster created in Example 1. A multi-owner SVM metadevice is added to the zone cluster. The set number of the metaset is 1, and the set name is oraset. An additional public IP address is added to the zone-cluster node on phys-schost-2. A shared QFS file system is also added to the configuration. Note that the **special** property of a QFS file system must be set to the name of the MCF file. The **raw** property must be left unspecified.

A UFS file system is exported to the zone cluster for use as a highly-available file system. It is assumed that the UFS file system is created on a Solaris Volume Manager metadevice.

```
phys-schost-1# clzonecluster configure sczone
clzc:sczone> add device
clzc:sczone:device> set match=/dev/md/1/dsk/d100
clzc:sczone:device> end
clzc:sczone> add device
clzc:sczone:device> set match=/dev/md/oraset/dsk/d100
clzc:sczone:device> end
clzc:sczone> select node physical-host=phys-schost-2
clzc:sczone:node> add net
clzc:sczone:node:net> set address=192.168.0.3/24
clzc:sczone:node:net> set physical=bge0
clzc:sczone:node:net> end
clzc:sczone:node> end
clzc:sczone> add fs
clzc:sczone:fs> set dir=/qfs/ora_home
clzc:sczone:fs> set special=oracle_home
clzc:sczone:fs> set type=samfs
clzc:sczone:fs> end
clzc:sczone> exit
```

Example 3 Creating a New Zone Cluster Using an Existing Zone Cluster as a Template

The following example shows how to create a zone cluster called *sczone1*, using the *sczone* zone cluster created in Example 1 as a template. The new zone cluster's configuration will be the same as the original zone cluster. Some properties of the new zone cluster need to be modified to avoid conflicts. When the administrator removes a resource type without specifying a specific resource, the system removes all resources of that type. For example, **remove net** causes the removal of all net resources.

```
phys-schost-1# clzonecluster configure sczone1
sczone1: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone1> create -t sczone
clzc:sczone1>set zonepath=/zones/sczone1
clzc:sczone1> select node physical-host=phys-schost-1
clzc:sczone1:node> set hostname=zc-host-3
clzc:sczone1:node> select net address=zc-host-1
clzc:sczone1:node:net> set address=zc-host-3
clzc:sczone1:node:net> end
clzc:sczone1:node> end
clzc:sczone1> select node physical-host=phys-schost-2
clzc:sczone1:node> set hostname=zc-host-4
clzc:sczone1:node> select net address=zc-host-2
clzc:sczone1:node:net> set address=zc-host-4
clzc:sczone1:node:net> end
clzc:sczone1:node> remove net address=192.168.0.3/24
clzc:sczone1:node> end
clzc:sczone1> remove dataset name=tank/home
clzc:sczone1> remove net
clzc:sczone1> remove device
clzc:sczone1> remove fs dir=/qfs/ora_home
clzc:sczone1> exit
```

Example 4 Creating a Whole-Root Zone Cluster

The following example shows the creation of a new zone cluster, *sczone2*, but now the constituent zones will be whole-root zones.

```
phys-schost-1# clzonecluster configure sczone2
sczone2: No such zone cluster configured
Use 'create' to begin configuring a new zone cluster.
clzc:sczone2> create -b
...
Follow the steps in Example 1 for the rest of the configuration
...
clzc:sczone2> exit
```

OPERANDS

The following operands are supported:

zoneclustername The name of the zone cluster. You specify the name of the new zone cluster. The *zoneclustername* operand is supported for all subcommands.

\+ All nodes in the cluster. The + operand is supported only for a subset of subcommands.

EXIT STATUS

The complete set of exit status codes for all commands in this command set are listed on the **Intro**(1CL) man page.

If the command is successful for all specified operands, it returns zero (**CL_NOERR**). If an error occurs for an operand, the command processes the next operand in the operand list. The returned exit code always reflects the error that occurred first.

This command returns the following exit status codes:

0 CL_NOERR

No error.

The command that you issued completed successfully.

1 CL_ENOMEM

Not enough swap space.

A cluster node ran out of swap memory or ran out of other operating system resources.

3 CL_EINVAL

Invalid argument.

You typed the command incorrectly, or the syntax of the cluster configuration information that you supplied with the **-i** option was incorrect.

18CL_EINTERNAL

Internal error was encountered.

36 CL_ENOENT

No such object

The object that you specified cannot be found for one of the following reasons:

- The object does not exist.

- A directory in the path to the configuration file that you attempted to create with the **-o** option does not exist.
- The configuration file that you attempted to access with the **-i** option contains errors.

37 CL_EOP

Operation not allowed

You tried to perform an operation on an unsupported configuration, or you performed an unsupported operation.

ATTRIBUTES

See **attributes**(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	**SUNWsczu**
Interface Stability	Evolving

SEE ALSO

cluster(1CL), **Intro**(1CL), **scinstall**(1M), **clnode**(1CL), zoneadm(1M), **zonecfg**(1M).

NOTES

The superuser can run all forms of this command.

All users can run this command with the **-?** (help) or **-V** (version) option.

To run the **clzonecluster** command with subcommands, users other than superuser require RBAC authorizations. See the following table.

Subcommand	RBAC Authorization
boot	**solaris.cluster.admin**
check	**solaris.cluster.read**
clone	**solaris.cluster.admin**
configure	**solaris.cluster.admin**
delete	**solaris.cluster.admin**
export	**solaris.cluster.admin**
halt	**solaris.cluster.admin**
install	**solaris.cluster.admin**
list	**solaris.cluster.read**
monitor	**solaris.cluster.modify**
move	**solaris.cluster.admin**
ready	**solaris.cluster.admin**
reboot	**solaris.cluster.admin**
show	**solaris.cluster.read**
status	**solaris.cluster.read**
uninstall	**solaris.cluster.admin**
unmonitor	**solaris.cluster.modify**
verify	**solaris.cluster.admin**