

VDCF - Virtual Datacenter Control Framework for the Solaris™ Operating System

Release Notes

Version 4.0
7. April 2011

Copyright © 2005-2011 JomaSoft GmbH
All rights reserved.

1 Introduction

This documentation describes the differences of the releases of the Virtual Datacenter Control Framework (VDCF) for the Solaris Operating System. It explains how to upgrade to the newest release.

See these other documents for further information:

<i>VDCF – Installation Guide</i>	for information about installing or upgrading this product
<i>VDCF – Quick Reference</i>	for a short command overview
<i>VDCF Base – Administration Guide</i>	for information about the VDCF Base usage
<i>VDCF vServer – Administration Guide</i>	for information about the VDCF vServer product usage
<i>VDCF LDom – Administration Guide</i>	for information about the VDCF LDom product usage
<i>VDCF – Monitoring</i>	for information about VDCF Monitoring

These and all other VDCF documents can be found at:

<http://www.jomasoft.ch/products/VDCF/docs/>

2 What's new in Release 4.0

2.1 New major feature: High Availability (HA) / Automated Failover

In this release we have implemented a new High Availability feature. It's available as an Enterprise Extension to our customers holding an VDCF Enterprise License.

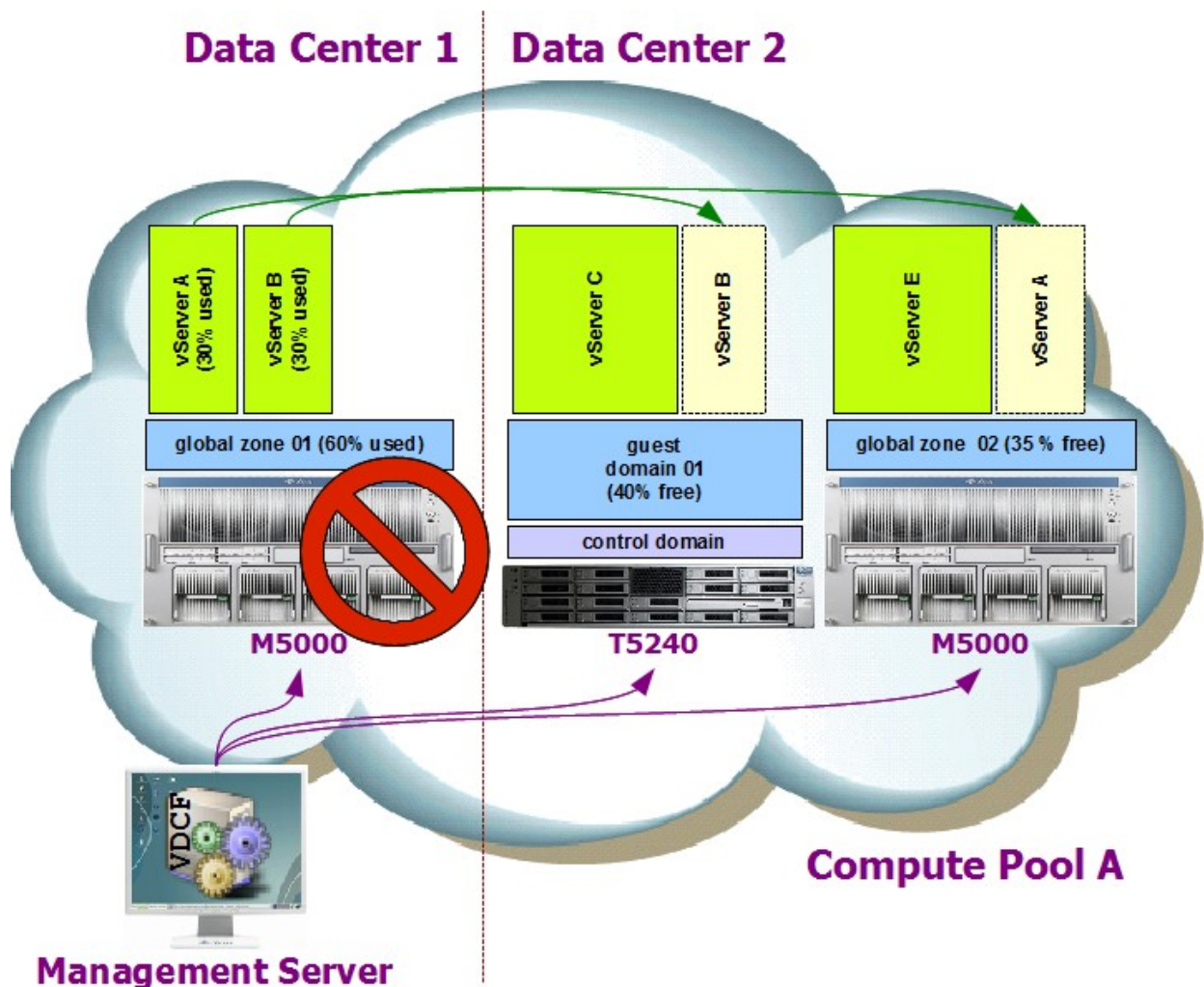
This feature is controlled by this new command:

hamon (High Availability Monitoring)

Used to check the health of VDCF Nodes. If a Node fails it may initiate evacuation of vServers. The vServers are then distributed to the available Nodes based on their resource usage values (RAM and CPU). This solution fills the gap between manual failover and full featured failover using Cluster Software.

See the 'VDCF Monitoring' Guide and the manpages for more information.

The following picture illustrates the migrations if the M5000 in Data Center 1 fails.



2.2 New tools

`vserver_local` Utility for easy failover of VDCF, if you installed VDCF itself in a Zone. More about this feature can be found in the VDCF Installation Guide.

2.3 New operations

`dataset -c update` Updates dataset size information in database with effective values.

`node -c activate` Activate a INACTIVE Node

`node -c evacuate` Detach all vServers and distributes them to Nodes with free resources. (used also by the hamon feature)

`node -c register` Register a system into VDCF, as alternative to install it using VDCF

`nodecfg -c modify_net` Modify network interface configuration of configured Node

`vserver -c reattach` Attaches multiple vServers to previous Node or control domain

2.4 Enhanced operations

diskadm -c show	To display the disk comments the new optional flag 'comment' may be used. Set the config value <code>DISKS_SHOW_COMMENT="TRUE"</code> to always show the disk comments.
node -c remove	New optional flag 'force' to remove node from patch target configurations.
nodecfg -c modify	Additional attributes to update: 'location', 'serial' and 'hostid'
nodecfg -c discover	new optional argument 'nonroot' to discover a node using user vdcfexec instead of root.
vserver -c addnet	Additional stack type 'exclusive' to create exclusive ip-stack vserver.
vserver -c attach	New optional flag 'boot' to boot vServer after successful attach.
vserver -c boot vserver -c reboot vserver -c shutdown	Additional arguments 'node' and 'cdom' to boot, reboot and shutdown multiple vServer with one command.
vserver -c create	Additional optional argument 'hostid' to define hostid for vServer on Solaris 10 Update 9 or later. Additional optional arguments 'category' and 'priority' for High Availability Monitoring
vserver -c modify	Additional arguments 'hostid' and 'clear_hostid' to set and clear hostid for vServer on Solaris 10 Update 9 or later. Additional arguments 'category' and 'priority' for High Availability Monitoring
vserver -c revert	new argument 'all' to revert all filesystems of a vServer.
zfsadm -c destroy zfsadm -c rollback	Arguments 'filesystem' and 'mountpoint' are now optional. Provide the snapshot name as <code>/zfsds/myfs@mysnapshot</code> to the 'snapshot' argument.

2.5 Deprecated operations

Don't use this operations anymore, they will be removed **in the next major** VDCF release.

vserver -c zfs* replaced by zfsadm command

build -c list replaced by build -c show
flash -c list_versions

dataset -c show_lun replaced by diskadm -c show

This release allows you to temporary disable the deprecated features, by adding the following line in `/var/opt/jomasoft/vdcf/conf/customize.cfg`

```
export FR_DEPRECATED_REJECT="TRUE"
```

Using this setting the operations return an error, which should help you to find scripts, where this operations are still used.

3 How to upgrade VDCF from a previous release

3.1 VDCF prerequisites

You must have VDCF Release 1.2.8 or later installed.
If you currently use an older Release, you need to migrate to VDCF 1.2.8 first.

Check your currently installed packages and versions:

```
$ vdcfadm -c show_version
```

Package	Version	Arch.	Install-Date	Name
JSvdcf-base	2.3.7	sparc	Feb 27 2010 11:15	VDCF - Base
JSvdcf-patch	2.3.7	sparc	Feb 27 2010 11:16	VDCF - Patch Management
JSvdcf-vserver	2.3.7	sparc	Feb 18 2010 22:00	VDCF - Virtual Server Management

3.2 Replace VDCF prerequisites and packages

Replace all installed VDCF packages (Base, Patch, vServer, LDom).

It is also **required** to replace vServer Enterprise Extensions (Resource Management) with the newest versions.

3.2.1 Remove packages

```
pkgrm JSvdcf-rm (if installed ..)
pkgrm JSvdcf-ldom (if installed ..)
pkgrm JSvdcf-vserver
pkgrm JSvdcf-patch JSvdcf-base
```

3.2.2 Replace VDCF prerequisites

If your previous VDCF version is older than version 3.0, you need to replace some of the VDCF prerequisites software using the installation script `install_vdcf_prereqs`.

```
SMClibgcc 3.3 is replaced by SMClgcc346
SMCsqlite 3.3.6 is replaced by SMCsqlite 3.6.13
```

A typical upgrade looks like this:

```
# cd <download-dir>/vdcf_prereqs

# ./install_vdcf_prereqs
Checking VDCF Prerequisites ...
Found old prerequisite SMClibgcc (Version: 3.3)
WARN: This package is not required by VDCF anymore.
WARN: You may remove this package, if no other software requires it.
Found old prerequisite SMCsqlite (Version: 3.3.6)
Removing this package now ...

Removal of <SMCsqlite> was successful.
```

```
Found SMCTk package
SMCTk tk
      (sparc) 8.4.9
VDCF skips installation of SMCTk (Version: 8.4.9)
Found SMCTcl package
SMCTcl tcl
      (sparc) 8.4.9
VDCF skips installation of SMCTcl (Version: 8.4.9)
Found SMCexpect package
SMCexpect expect
      (sparc) 5.40
VDCF skips installation of SMCexpect (Version: 5.40)
Found SMCncurs package
SMCncurs ncurses
      (sparc) 5.5
VDCF skips installation of SMCncurs (Version: 5.5)
Found SMCreadl package
SMCreadl readline
      (sparc) 5.2
VDCF skips installation of SMCreadl (Version: 5.2)
Installing SMClgcc346 ...
Installing SMCsqlite ...
Check /var/tmp/vdcf_prereqs.log for pkgadd details.
Finished
```

3.2.3 Install new packages

If you own a valid *VDCF vServer Enterprise* or a *VDCF LDom license* you may download and install the *VDCF LDom package* for managing Solaris Logical Domains. For the *VDCF Monitoring package* a *VDCF vServer Enterprise license* is required.

```
pkgadd JSvdcf-base JSvdcf-patch
pkgadd JSvdcf-vserver
pkgadd JSvdcf-ldom
pkgadd JSvdcf-rm
pkgadd JSvdcf-monitor
```

3.3 Update Client Package

It is **required** to update the VDCF client package on all nodes.
If a node is down, please boot it to update the client package.

```
vdcfadm -c update_node all
```

Verify that all nodes got the new VDCF client package:

```
vdcfadm -c show_node all
```

4 Known Issues

4.1 ZFS data filesystems

Due to the Solaris bug **6449301** it is currently not supported to create a new vServer with ZFS data filesystems.

The following workarounds are available:

First create a vServer with a root filesystem only. After a successful commit add your data filesystems.

If you need to re-create a vServer after a `vserver -c commit uninstall` command has completed you have to detach the ZPOOL dataset containing the ZFS data filesystems before you create the vServer using the commit operation. After the successful commit you attach the ZPOOL dataset and commit again.