

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

VDCF Blueprint

Physical to Virtual (P2V)

Version 1.0
25. March 2019

Copyright © 2005-2019 JomaSoft GmbH
All rights reserved.

Table of Contents

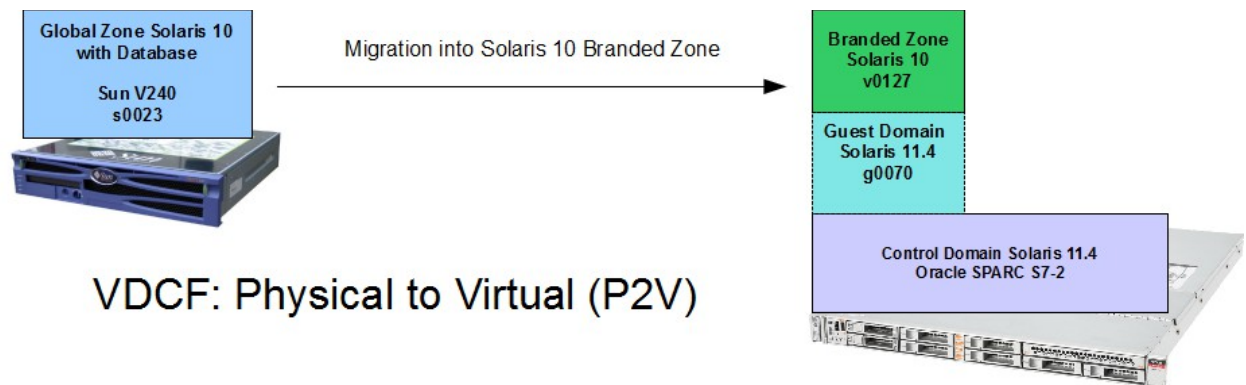
1 Introduction.....	3
1.1 Overview.....	3
1.2 Requirements.....	3
2 Import information about the old Server into VDCF.....	4
2.1 Preparations on the source System s0023.....	4
2.2 Import into the VDCF Management Server.....	5
3 Create and transfer Flash Archive.....	6
3.1 Create a Flash Archive using flarcreate.....	6
3.2 Move the Archive to the target Guest Domain.....	6
4 Create and install the Solaris 10 Branded Zone.....	7
5 Application/Database/Data Migration.....	9

1 Introduction

1.1 Overview

VDCF supports the Migration of Global Zones into vServers (Solaris Zones). This process is called "Physical to Virtual" (P2V) or "Lift and Shift".

This Migration process works most efficiently if the application data is stored on a Shared Storage. In this blueprint we migrate a small sample Oracle database from an old Sun V240 into a Solaris 10 Branded Zone on a Solaris 11.4 Guest Domain which is running on a modern SPARC S7-2 Server.



VDCF: Physical to Virtual (P2V)

The SPARC S7-2 Server and the g0070 Solaris 11.4 Guest Domain are already installed.

As a first step we prepare the target system named v0127 as a Solaris 10 Branded Zone. That branded zone will be installed based on a Flash Archive taken from the source V240 system named s0023. The Oracle database is stored on zpools on a SAN. This data is migrated in the last step.

1.2 Requirements

To use this feature VDCF 7.2.0 or later is required.

2 Import information about the old Server into VDCF

2.1 Preparations on the source System s0023

```
# hostname
s0023

# uname -a
SunOS s0023 5.10 Generic_150400-17 sun4u sparc SUNW,Sun-Fire-V240

# yes | pkgadd -d http://g0075-mngt/pkg/$(uname -p)/JSvdcf-client.pkg JSvdcf-client

## Downloading...
.....25%.....50%.....75%.....100%
## Download Complete

Processing package instance <JSvdcf-client> from <http://g0075-mngt/pkg/sparc/JSvdcf-
client.pkg>

JomaSoft VDCF - Client(sparc) 7.2.0
Copyright (c) 2005-2019 JomaSoft GmbH
All rights reserved.
Use is subject to license terms.

...
...

/opt/jomasoft/vdcf/client/smf/svm_metadb_recover.xml
/opt/jomasoft/vdcf/client/smf/vdcf_iscsi.xml
/opt/jomasoft/vdcf/client/smf/zfs_on_nfs.xml
[ verifying class <none> ]
## Executing postinstall script.

*** MANUAL-TASK TODO ***
Add Public Key of root at Management Server to vdcfexec authorized_keys using :
/opt/jomasoft/vdcf/client/sbin/update_key -u <FLASH_WEBSEVER_URL>

Installation of <JSvdcf-client> was successful.

# /opt/jomasoft/vdcf/client/sbin/update_key -u http://g0075-mngt

Obtaining public key ... done.
SSH Key updated successfully.
```

2.2 Import into the VDCF Management Server

In this step we discover the source system (LUNs, zpools, filesystems and network interfaces) and add this information as configuration data into the VDCF Repository.

```
-bash-4.4$ node -c import name=s0023

Importing new Node s0023 ...
Discover Systeminfo ...
Discover Rootdiskinfo ...
Discover Diskinfo ...
This may take some time, it depends on the number of disks

Discover Netinfo ...
Node configuration successfully added.
System registration done for s0023.
node with all vservers being checked: s0023
check on node s0023 successful
patch deployment updated from node s0023
registering disks from node s0023
New visible Lun 6001438012599B9B0001100001BC0000 Size: 5.00 GB
New visible Lun 6001438012599B62000110002A2A0000 Size: 35.00 GB
New visible Lun 6001438012599B6200011000118D0000 Size: 10.00 GB
New visible Lun 6001438012599B620001100028720000 Size: 5.00 GB
importing node datasets from node s0023
Root Dataset s0023_root (ZPOOL: rpool) with Size 68.00 GB successfully imported from Node
s0023
Successfully added node filesystem 'ROOT/zfsroot' with mountpoint '/' (ZPOOL: rpool) to
dataset 's0023_root'
Successfully added node filesystem 'export' with mountpoint '/export' (ZPOOL: rpool) to
dataset 's0023_root'
Successfully added node filesystem 'export/home' with mountpoint '/export/home' (ZPOOL:
rpool) to dataset 's0023_root'
Dataset oratest (ZPOOL) with Size 34.79 GB successfully imported from Node s0023
Successfully added node filesystem 'data/oracle' with mountpoint '/oracle' to dataset
'oratest'
Successfully added node filesystem 'data/oradata' with mountpoint '/oradata' to dataset
'oratest'
Dataset oratestlog (ZPOOL) with Size 4.96 GB successfully imported from Node s0023
Successfully added node filesystem 'data/oradata_redo' with mountpoint '/oradata/redo' to
dataset 'oratestlog'
No vServer found on Node s0023.
WARN: Add console configuration manually using: console -c add node=s0023
Node s0023 import finished
```

3 Create and transfer Flash Archive

3.1 Create a Flash Archive using flarcreate

On the source system we create a Solaris 10 Flash Archive which includes OS and configuration and omit the application/database filesystems (because this data is already on shared storage).

On the source system s0023:

```
-bash-3.2$ df -h | grep ora

oratest/data/oracle          34G   4.3G   29G   13%   /oracle
oratest/data/oradata        34G   1.3G   29G    5%   /oradata
oratestlog/data/oradata_redo 4.9G   151M   4.7G    4%   /oradata/redo

# time flarcreate -c -S -n s0023_u11 -L pax -x /oradata -x /oracle /var/tmp/s0023-s10-2019-03-18.flar

Archive format requested is pax
Full Flash
Checking integrity...
Integrity OK.
Running precreation scripts...
Precreation scripts done.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.

Running pre-exit scripts...
Pre-exit scripts done.

real    6m10.909s
user    3m43.090s
sys     0m59.308s

# ls -lh /var/tmp/s0023-s10-2019-03-18.flar
-rw-r--r--  1 root      root           1.4G Mar 18 18:34 /var/tmp/s0023-s10-2019-03-18.flar
```

3.2 Move the Archive to the target Guest Domain

Then we transfer the Flash Archive to our target Guest Domain g0070.

```
# scp /var/tmp/s0023-s10-2019-03-18.flar marcel@g0070:
Password:
s0023-s10-2019-03-18 100% |
*****| 1464 MB      01:10
```

On g0070 (the system where the branded zone v0127 gets installed) we put the flar into /var/tmp/images/:

```
# mkdir /var/tmp/images
# mv s0023-s10-2019-03-18.flar /var/tmp/images/v0127.flar
```

4 Create and install the Solaris 10 Branded Zone

First we have to define the new branded zone v0127 in VDCF using vserver type SOL10 (Solaris 10 branded zone):

```
-bash-4.4$ vserver -c create name=v0127 type=SOL10 node=g0070 comment="p2v"  
  
creating vServer v0127  
vServer created successfully  
  
-bash-4.4$ dataset -c create name=root vserver=v0127 size=20g  
  
Creating vServer dataset <v0127_root>  
Disk 6001438012599B620001100030A10000 (MPXIO) with Size 20.0 GB selected.  
Dataset v0127_root (ZPOOL) created successfully
```

The vServer/Solaris Zone is installed using a new IP address. After the migration this can be changed.

```
-bash-4.4$ vserver -c addnet name=v0127 type=management ipaddr=v0127-mngt  
  
netmask=255.255.255.0  
adding network  
network definitions added  
  
bash-4.4$ vserver -c addfs name=v0127 type=root  
  
adding root filesystem to vServer: v0127  
Filesystem /zones/v0127 defined on dataset v0127_root (ZPOOL).  
root filesystem defined successfully
```

The vServer commit command installs the branded zone using the prepared Flash Archive which is already stored on the Guest domain g0070.

```
bash-4.4$ vserver -c commit name=v0127

committing datasets for vServer v0127
dataset commit successful
committing filesystems for vServer v0127
Filesystem /zones/v0127 created in v0127_root on g0070
filesystem commit successful
committing vServer v0127 - this may take a moment ...
Progress being logged to /var/log/zones/zoneadm.20190318T170903Z.v0127.install
  Installing: This may take several minutes...
Postprocessing: This may take a while...
  Postprocess: Updating the image to run within a zone
  Postprocess: Migrating data
    from: v0127_root/root/rpool/ROOT/zbe-0
    to: v0127_root/root/rpool/export
  Postprocess: A backup copy of /export is stored at /export.backup.20190318T171322Z.
It can be deleted after verifying it was migrated correctly.
  Postprocess: Migrating data
    from: v0127_root/root/rpool/ROOT/zbe-0
    to: v0127_root/root/rpool
  Postprocess: A backup copy of /rpool is stored at /rpool.backup.20190318T171322Z.
It can be deleted after verifying it was migrated correctly.

  Result: Installation completed successfully.
Log saved in non-global zone as
/zones/v0127/root/var/log/zones/zoneadm.20190318T170903Z.v0127.install
vServer successfully committed.
commit successful

-bash-4.4$ vserver -c boot name=v0127
```

After booting the vServer our physical system is now running in a branded zone!

5 Application/Database/Data Migration

Only in this last migration step we have a few minutes downtime for the database.
We can execute all the commands on the central VDCF Management Server.

```
-bash-4.4$ date
Tuesday, March 19, 2019 at 7:40:28 AM CET

-bash-4.4$ serverconfig -c exec server=s0023 user=oracle command="/oracle/stopdb"

Executing on Node s0023 (Discovered Node)
Executing command as oracle: /oracle/stopdb
Oracle Corporation      SunOS 5.10      Generic Patch   January 2005
Exit-Code: 0
Connection to 192.168.20.23 closed.
execution successful
```

We "remove" (detach) the datasets (zpool) from the old Server.

```
-bash-4.4$ dataset -c detach name=oratestlog
detaching dataset: oratestlog
dataset detached successfully

-bash-4.4$ dataset -c detach name=oratest
detaching dataset: oratest
dataset detached successfully
```

Then we assign them, attach and mount on the target vServer (Branded Zone).
All the required steps to configure the Guest Domain (LDom) is done by the tool.

```
-bash-4.4$ dataset -c assign name=oratestlog vserver=v0127
assigning dataset oratestlog to vserver v0127
dataset assigned successfully

-bash-4.4$ dataset -c assign name=oratest vserver=v0127
assigning dataset oratest to vserver v0127
dataset assigned successfully

-bash-4.4$ dataset -c attach name=oratest
attaching dataset: oratest
dataset attached successfully

-bash-4.4$ dataset -c attach name=oratestlog
attaching dataset: oratestlog
dataset attached successfully

-bash-4.4$ vserver -c mount name=v0127 dataset=oratest
mounting filesystems on dataset of vServer v0127
dataset (oratest) filesystems mounted successfully

-bash-4.4$ vserver -c mount name=v0127 dataset=oratestlog
mounting filesystems on dataset of vServer v0127
dataset (oratestlog) filesystems mounted successfully
```



```
-bash-4.4$ serverconfig -c exec server=v0127 user=oracle command="/oracle/startdb"
```

```
Executing on vServer v0127 (p2v) on Node g0070  
Executing command as oracle: /oracle/startdb  
Oracle Corporation      SunOS 5.10      Generic Patch   January 2005  
Exit-Code: 0  
Connection to 192.168.20.70 closed.  
execution successful
```

All done. Downtime was less than 5 minutes for this small database.

```
-bash-4.4$ date  
Tuesday, March 19, 2019 at 7:43:55 AM CET
```