

**Definitions**

**HA/SAF** #High Availability Storage Foundation  
**Hypervisor** #Domain 0, host, microkernel, VM Monitor  
 #Privileged domain  
**Virtual Machine (vm)** #DomU, guest, unprivileged domain  
**Migration** #Relocation of vm; about 100ms downtime  
**STONITH** #Shoot The Other Node In The Head

**Xen restrictions**

#[novell.com/documentation sles10 guide](http://novell.com/documentation/sles10/guide):  
 ISA or PCMCIA devices tend not to work. ISA DMA is not supported.  
 ACPI support in Xen is improving, but may not be up to date with respect to Linux. Some large-memory machines may not boot with ACPI enabled.  
 Power management features, such as the suspend-to-disk feature, are not supported and should be disabled on both Domain 0 and any VM.  
 SLES9 SP3 as VM unsupported #Possible SLES10 SPI feature

**See SLES10 GA Release Notes, e.g:**

No Full Virtualization guarantee #Intel VT or AMD Virtualization cpu works  
[http://wiki.xensource.com/xenwiki/HVM\\_Compatible\\_Processors](http://wiki.xensource.com/xenwiki/HVM_Compatible_Processors)  
 Migration on x86-64 not stable #Will be available via update (release note)  
 Hot-add-memory not yet #Will be available via update (release note)  
 Max 16GB memory is stable #PAE kernel for memory > 4GB

**Xen VM Migration setup**

#  
<http://www.novell.com/documentation/vmserver/index.html>

**Install sles10 on 3 machines (2 nodes)**

#Reserve >=4 GB unpart. space #Use bare part./LVM/EVMS/or sparse file  
 #Include High Availability & XEN pattern, and yast2-vm, yast2-heartbeat  
 #Do not use Network Manager  
 #Enable ip\_forward #By using yast or sysctl.conf  
**Packages iSCSI** #iSCSI for demo purposes  
 open-iscsi & iscsitarget #Do not plan initiator & target on 1 node  
**Packages XEN** #Physical Address Extension: 32bit & 4G  
 kernel-xen[pa]e  
 xen, xen-libs, xen-tools, xen-tools-ioemu

**Survive a reboot**

/etc/init.d/powersaved stop #Prevent power management conflicts  
 chkconfig powersaved off #Do not survive a reboot  
 /etc/init.d/xend start #Start XEN daemon & survive reboot:  
 checkconfig --level 35 xend on #Possibly troubleshoot network aft reboot  
 #If required, change eth0 settings in three steps:  
 /etc/xen/scripts/network-bridge stop netdev=eth0  
 yast2 lan  
 /etc/xen/scripts/network-bridge start netdev=eth0

**Disable SuSEfirewall**

#Or enable xenbr0 forwarding:  
 vi /etc/sysconfig/SuSEfirewall2  
 FW\_FORWARD\_ALWAYS\_INOUT\_DEV="xenbr0"  
 /etc/init.d/SuSEfirewall2\_setup restart

**Default bootmenu**

#default title nr starts with 0  
 vi /boot/grub/menu.lst  
 title SUSE Linux Enterprise XEN Server 10  
 root (hd0,4)  
 kernel /boot/xen-pae.gz  
 module /boot/vmlinuz-xenpae root=/dev/sda5 vga=0x31a splash=silent  
 showopts  
 module /boot/initrd-xenpae

[http://en.opensuse.org/Installing\\_Xen3](http://en.opensuse.org/Installing_Xen3)

brctl show #Output after a reboot, e.g.:  

bridge name	bridge id	STP enabled	interfaces
xenbr0	8000.feffffff	no	vif0.0 peth0

**Setup a Virtual Machine**

#Prepare XEN installation source, e.g.  
 #Copy SLES10 ISO file to host  
 #Show unpartitioned space  
 fdisk -l  
 #No cluster support for SLE10 and LVM vg. Use EVMS container(s) instead.  
 vgdisplay #Show free space for LVM  
 lvdisplay #Show Logical Volumes  
 mount #Show mounted volumes  
 evms\_activate #If EVMS is not used during installation  
 evmsgui #Show free space for EVMS  
 dmsetup ls #Map Volume Name to mapper device  
 system-vm2(253,3) #Maps to: /dev/dm-3  
 vi /etc/xen/scripts/block #Remove -y for fast unsecure sparse file  
 do\_or\_die losetup -y "\$sloopdev" "\$file"

Yast, System, Virtual Machine Management (Xen)

xm create /etc/xen/vm/vm1 #Start VM. Create domain in memory.  
 xm list #Get <DomId>  
 xm console 2 #Open screen <DomId>  
 #Disconnect: <Ctrl-]>  
 xm top #Monitor domains. Base info: xm info  
 xm shutdown 2 #Shutdown VM <DomId>  
 xm reboot 2 #Restart VM <DomId>  
 xm destroy 2 #Kill VM <DomId>  
 xm migrate 2 host #Migrate <DomId> hostname  
 xm pause 2 #Standby VM <DomId>  
 xm unpause 2 #Start VM <DomId> from standby  
 xm save 2 file #Suspend VM <DomId> to filename  
 xm restore file #Resume VM from filename  
 xm mem-set 0 512 #Set memory of domain 0 to 512MB

**Heartbeat 2 Setup**

#At least 2 ethernet cards, 2 nodes  
 #Recommended redundant NICs #Software RAID 1 is not supported  
 #Optional STONITH Agent for power supply  
[http://www.novell.com/documentation/sles10/hb2/data/hb2\\_config.html](http://www.novell.com/documentation/sles10/hb2/data/hb2_config.html)

**Heartbeat 2 definitions**

#Four Layers  
 Messaging/Infrastructure Layer #"I am alive" "Heartbeat" Layer  
 Membership Layer #Cluster Consensus Membership service  
 Resource Allocation Layer #Resource Administration Services  
 Resource Layer # (RA) Resource Agents with scripts  
 Standby node #Node with ability to run a resource  
 Cluster Resource Manager # (CRM) Master of Ceremony. One is DC:  
 Designated Coordinator # (DC) Own/react on master CIB changes  
 Cluster Information Base # (CIB) XML cluster setup/view  
 Policy Engine # (PE) Step ordering  
 Transition Engine # (TE) Step execution  
 Local Resource Manger # (LRM) Call Resource Agents  
**Setup name resolution** #Make all hostnames resolvable on nodes  
 #Or setup DNS. Check via: ping nodename  
 vi /etc/hosts #Initial time set: ntpdate ntp.srv.com, or:  
**Setup time sync**  
 ssh node1 date \$(date +%m%d%H%M)  
 ssh node2 date \$(date +%m%d%H%M)  
 vi /etc/ntp.conf #Time syncs only with max 1000 sec delta  
 server ntp.srv.com #Comment out server and fudge lines  
 /etc/init.d/ntp start #Start time sync and wait 5 minutes.  
 ntpq -p #Check time sync. (or ntptrace)

**Initial heartbeat setup**

```

yast2 heartbeat #Add nodes & propagate configuration
                  #On one of the nodes
Add node(s), Next #Unlimited. Tested to 16
Select authentication key, Next #Same on all nodes (none, md5, or sha1)
On and Survive reboot #chkconfig --level 35 heartbeat on
/usr/lib/heartbeat/ha_propagate #Replicate configuration to nodes
cat /etc/ha.d/authkeys #View configuration file
cat /etc/ha.d/ha.cf #View configuration file
cat /var/lib/heartbeat/crm/cib.xml #Is replicated aft heartbeat start
/etc/init.d/heartbeat start #On the other node(s)
chkconfig --level 35 heartbeat on #On the other node(s)
passwd hacluster #Give user(s) hacluster a password for:
hb_gui #Add a resource (from any node), e.g.:
+ (add new item), native type, OK,
Resource ID: test-ip, IPaddr (OCF RA) as Type, Param. Value: ip:172.17.0.170,
optional: Add Parameter, Name: nic, Value: eth0, OK,
Start Resource (MB2) #Check via ping, ifconfig and unplug cable

```

<http://linux-ha.org/v2>

**iSCSI Setup**

```

Initiator #Used as cheap SAN for Image Store
Target #User of the block level iSCSI device
Add a new partition on target #Host sharing the block device
yast disk #e.g. /dev/hdab, /dev/vg/lv,
           #/dev/evms/lvm2/cont/lv
Do not mix LVM & EVMS #Best practices
No cluster support for SLE10 and LVM vg. Use EVMS container(s) instead,
but iSCSI provides a block device not an LVM volume group
dmsetup ls #Map Volume Name to mapper device
system-vm2(253,3) #Maps to: /dev/dm-3
Configure iSCSI target #File, block device, RAID or LVM device
yast iscsi-server #Provide /dev/sd device for client
When booting, (Open Firewall) #In Service tab
Delete demo target #In Targets tab
Add target #e.g. /dev/vg-usb/lv-xeni
No authentication in demo #Next, OK, Finish, Restart, Yes
Configure iSCSI initiator #From first node
yast iscsi-client #Connect IET-VIRTUAL-DISK /dev/sda
When booting #In Service tab
Discovery, Fill in IP of target #In Discovered Targets tab
Login, no authentication #
Toggle startup, Finish #Survive reboot
yast disk #Add 3 partitions (IET-VIRTUAL-DISK):
Create part. for VM image #Leave e.g. 300MB free. No mount point
Create part. for VM config. files #Use e.g. 200MB. No mount point
Create part. for VM data storage #Use e.g. 100MB. No mount point
Configure iSCSI initiator #From other node(s) (discovery only)

```

**OCFS2 Setup**

```

Can run in pure OCFS2 Cluster #Oracle Cluster File System
                                #Integration with heartbeat2 (user space)
ocfs2console #GUI for setup and propagation
Initialize the native OCFS2 stack #Only on ONE node
Cluster, Configure nodes, Close #Name, IP and port (TTTT)
Add nodes (incl. first), Close #Copy via ssh and close ocfs2console
Close, Cluster, Propagate Config. #Add heartbeat and bootconfig (all nodes)
/etc/init.d/o2cb configure #On first node only (because of failure)
/etc/init.d/o2cb force-reload #Check for 'user'
cat /sys/o2cb/heartbeat_mode #Interface between kernel & user space
find /sys/kernel/config/cluster #Create OCFS2 file systems (sda1 & sda2)
mkfs.ocfs2 /dev/sda1 #Ask for UUID
mounted.ocfs2 -d /dev/sda1 #Integrate
OCFS2 in heartbeat cluster #Clonesets can run concurrent and on all nodes
                                #OCFS2 via clone File System RA on each node
                                #Notify resource stop/start, node join/leave
                                #Demo stonith device is ssh reboot:
                                #Simulate node crash by killing heartbeat and not unplug cable
vi cibbootstrap.xml #Create XML blobs for the CIB
cibadmin -C -o crm_config -x /cibbootstrap.xml
cibadmin -C -o crm_config -x /stonithcloneset.xml
cibadmin -C -o crm_config -x /imagestorecloneset.xml
cibadmin -C -o crm_config -x /configstorecloneset.xml
#Check for /var/lib/xen/images & /etc/xen/vm via:
mount
crm_mon -l #List cluster resources
Test by manually umount ocfs2 #Remount should occur
Setup ssh keys for root #For unattended ssh stonith
Enable atd #For ssh stonith
pkill heartbeat #Test by node crash emulation
VM as Cluster Resource
Change sync mode of loop device #Undo after installing VM
vi /etc/xen/scripts/block #Remove -y at do_or_die
Create VM on node1 #Use default Sparse File
Restore -y #Sync mode
Stop VM on node1 #Inside VM is a not cluster safe fs
yast xen #Check availability on node2
cibadmin -C -o crm_config -x /vmlocation.xml
cibadmin -C -o crm_config -x /vml.xml
crm_mon -l #xm list
cibadmin -C -o crm_config -x /vmlorderconstraints.xml
pkill heartbeat #Test by node1 crash emulation
crm_mon -l #xm list on node2
Extend maximum loop mounts #Default max loops is 8
rmmod loop #Extend without reboot. Remove module.
modprobe loop max_loop=64 #Extend without reboot
vi /etc/modprobe.conf #Was max_loop=64 as SLES9 boot par.
options loop max_loop=64

```