

# General System Troubleshooting

**Sascha Wehnert**

Premium Service Engineer

Attachmate Group Germany GmbH

[swehnert@suse.com](mailto:swehnert@suse.com)



# What is this about?

- This session will cover the following topics:
  - How to speed up a service request
  - How to gather system information using supportconfig
  - Configure serial console in grub to trace kernel boot messages
  - Accessing a non booting systems using the rescue system
  - System crash situations and how to prepare (i586/x86\_64 only)

# The challenge of a service request

- Complete service request description:  
“We need to increase our disk space.”

# The challenge of a service request

- Which SUSE® Linux Enterprise Server version?
- Is this a physical or virtual environment?
- If virtual, what virtualization solution is being used?
- If physical, local SCSI RAID array? What hardware?
- If using HBAs, dm-multipathing or iSCSI connected disks or a 3<sup>rd</sup> party solution?
- Disk and system partition layout?
- What has been done so far? What was achieved? What failed?
- What information do I need in order to help?

# What information would be needed?

- SUSE Linux Enterprise Server version →  
`/etc/SuSE-release, uname -a`
- Physical → `dmidecode`  
XEN → `/proc/xen/xsd_port`  
KVM → `/proc/modules`
- Hardware information → `hwinfo`
- Partition information → `parted -l, /etc/fstab`
- Multipathing/iSCSI → `multipath, iscsiadm`
- Console output or `/var/log/YaST2/y2log` in case YaST2 has been used

# supportconfig

- Since SUSE Linux Enterprise Server 10 SP4 included in default installation.
- Maintained package, updates available via patch channels. For best results always have latest version installed from channels installed.
- One single command to get (almost) everything.
- Splits data into files separated by topic.
- Can be modified to exclude certain data, either via `/etc/supportconfig.conf` or command options.
- Creates txt files in `/var/log/nts_$(hostname)_$(date)_$(time)`, when done compresses everything into a tbz.

# supportconfig

```
sles11sp3:~ # supportconfig
=====
Support Utilities – Supportconfig
Script Version: 2.25-435
Script Date: 2014 09 10
[...]
Gathering system information
Data Directory: /var/log/nts_sles11sp3_141008_1038
Basic Server Health Check...      Done
RPM Database...                    Done
Basic Environment...               Done
System Modules...                  Done
Memory Details...                  Done
Disk I/O...                         Done
B-tree File System...              Skipped
YaST Files...                       Done
File System List...                 Skipped
[...]
System Logs...                      Done
Creating Tar Ball
==[ DONE ]===== Log file
tar ball: /var/log/nts_sles11sp3_141008_1038.tbz
Log file size: 1.1M
Log file md5sum: 3385bcc344ed2300620ee4dc7edfc7f8
=====
```

- Please find examples for common supportconfig commands in the Appendix

Serial console settings to trace kernel errors during boot time

# Serial console settings

- Capturing kernel messages at boot time
- Settings will be put into `/boot/grub/menu.lst`
- Traditional setup with a serial cable to a different system
- Modern remote access controllers allow serial console redirection to a log file.
- For remote access controllers, special features and additional licensing might be needed. Check documentation and contact the vendor for details.

# Serial console settings

- In XEN DomUs add `-console` to `xm start` command to print messages to the console.
- In Vmware, add a serial console port to the VM and select a file name on any given data store.
- In KVM, declaration in the VM XML by using `virsh` command.
- Special considerations when using XEN kernel. Settings need to be added to the kernel and module lines.

# Serial console settings

- If the grub menu should be controlled via serial console, put this into the global settings in `/boot/grub/menu.lst`:

```
serial --unit=0 -speed=115200
terminal --timeout=15 serial console
```

- Adding

```
console=tty0 console=ttyS0,115200
```

to the kernel append line is usually sufficient.

- In case nothing is logged, run a tail on the log file and do `echo "test" > /dev/ttySx` on the machine to check which tty is used for redirection.
- To increase output during boot up, pass `debug ignore_loglevel` as additional options.

System does not boot

# Possible reasons for non booting systems

- Hardware / firmware related problems  
The hardware reports issues before the boot loader starts.
- The boot loader was not properly installed into the MBR.  
Its config file contains errors or its experiencing problems with the specified devices.
- Kernel / initrd related issues
- Filesystem problems
- Problems with certain services

# What you need to access a system?

- DVD1 of the SUSE Linux Enterprise Server DVD image set. This is the one containing the binary RPMs and installation system. DVD2 only contains the source RPMs.
- If you have a network installation source available and can boot via PXE/TFTP, you can edit/copy the entry for the given installation source and add the `rescue=1` option to boot the rescue system via the network.

# How to access a system?

- As long the system can boot and only stops somewhere before the specified runlevel, the first thing to try would be to boot into single user mode using boot option “s” to obtain access to a minimal system.
- If this option is not available, try to:  
boot from DVD1 → select Repair Installed System → wait for the repair system to load → Expert Tools → Boot installed system.
- SUSE Linux Enterprise Server 12 does not provide the repair system anymore. To boot an installed system, please start the installation, abort it in YaST2, exit to linuxrc and then select Start Installation → Boot Installed System

# How to access a system?

- If for some reason the repair system cannot load the installed system, the last line of defense is the rescue system. Boot from DVD1 and select rescue system. For your own convenience, select the proper keyboard layout using F2 before you start.
- In SUSE Linux Enterprise Server 12 the rescue system provides the `loadkeys` command. Here it is possible to change the keyboard layout.

# Rescue system

- Login as root, no password needed
- Basic environment lets you access the failing machine.
- Always make sure you use the right ISO for the correct architecture. Booting a x86\_64 with an i586 ISO won't work if you try to setup a chroot environment.
- The biggest challenge is to have access to the ISO or rescue system when needed.
- When using LVM, be advised the SLE11SP3 rescue system has a known bug not creating the `/dev/<vgname>/<lvname>` links. Run `vgmknodes` to avoid or use the links in `/dev/mapper/`. See also: <https://www.suse.com/support/kb/doc.php?id=7014112>

# Steps to setup chroot

- Identify the partitioning layout: `parted -l`
- Mount the root device first to `/mnt`
- If further mount points exist, mount them as necessary. Keep in mind, you are building a structure at `/mnt`. For mounting the var-partition you would use:  
`mount /dev/sdXY /mnt/var`
- In multipathed environments start `multipathd` first before you mount the LUNs.
- For md based devices use the respective `/dev/mdX` devices, for logical volumes the entries in `/dev/mapper` or `/dev/<vgname>/<lvname>`

# Steps to setup chroot

- When ready, bind-mount `/proc`, `/sys` and `/dev` from the rescue system into `/mnt`:

```
mount -o bind /proc /mnt/proc
```

(same for `sys` or `dev`) or all in one go:

```
for fs in proc sys dev; do mount -o \
bind /$fs /mnt/$fs; done
```

- When everything is mounted:

```
chroot /mnt
```

- From here you have access to the server and should be able to start services such as network and `sshd` if someone wants to provide remote assistance.

# Steps to setup chroot

- For boot related problems, such as errors in the initrd or boot loader configuration make sure the boot device is also mounted.

# Steps to setup chroot

- Example:

```
Rescue:~ # mount /dev/system/root /mnt
Rescue:~ # mount /dev/system/var /mnt/var/
Rescue:~ # mount /dev/sda1 /mnt/boot/
Rescue:~ # for fs in proc sys dev; do mount -o bind /$fs /mnt/$fs;done
Rescue:~ # mount
tmpfs on /dev/shm type tmpfs (rw,mode=1777)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/mapper/system-root on /mnt type ext3 (rw)
/dev/mapper/system-var on /mnt/var type ext3 (rw)
/dev/sda1 on /mnt/boot type ext3 (rw)
/proc on /mnt/proc type none (rw,bind)
/sys on /mnt/sys type none (rw,bind)
/dev on /mnt/dev type none (rw,bind)
Rescue:~ # chroot /mnt
Rescue:/ # dhcpcd eth0
Checking for network time protocol daemon (NTPD):
Rescue:/ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
    inet 127.0.0.2/8 brd 127.255.255.255 scope host secondary lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:44:52:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.122/24 brd 192.168.66.255 scope global eth0
    inet6 fe80::20c:29ff:fe44:52bb/64 scope link
        valid_lft forever preferred_lft forever
Rescue:/ #
```

unused

# Leaving chroot

- When the issue has been fixed and the server is about to be rebooted, we should unmount everything first.
- Stop everything you started in the chroot environment otherwise you won't be able to unmount the devices because of device or resource busy errors.
- `exit`
- `cd`
- `umount /mnt/*`
- `umount /mnt`
- `reboot`

# What about XEN/KVM based systems?

- For XEN/KVM based systems, the rescue system is always the Dom0/host:

```
xen:/var/lib/xen/images/opensuse12.1 # fdisk -l disk0.raw
Disk disk0.raw: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders, total 8388608 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Disk identifier: 0x000c550f

Device	Boot	Start	End	Blocks	Id	System
disk0.raw1		2048	1269759	633856	82	Linux swap / Solaris
disk0.raw2	*	1269760	8388607	3559424	83	Linux

# What about XEN/KVM based systems?

- Import the partitions using `kpartx -a`:  
`kpartx -a /var/lib/xen/images/test/disk0`
- This will create the appropriate loop devices in `/dev/mapper/` which can be used to mount the filesystems.
- When complete, unmount the devices and run `kpartx -d` against the disk file to delete the mapped partitions.

# What about XEN/KVM based systems?

- `xen:/dev/mapper # ls -l`

```
total 0
```

```
drwxr-xr-x  2 root root    60 Oct 14 18:10 ./
```

```
drwxr-xr-x 17 root root  5780 Oct 14 18:10 ../
```

```
crw-rw----  1 root root 10, 236 Sep 29 13:51 control
```

```
xen:/dev/mapper # kpartx -a /var/lib/xen/images/test/disk0.raw
```

```
xen:/dev/mapper # ls -l
```

```
total 0
```

```
drwxr-xr-x  2 root root   100 Oct 14 18:11 ./
```

```
drwxr-xr-x 17 root root  5820 Oct 14 18:11 ../
```

```
crw-rw----  1 root root 10, 236 Sep 29 13:51 control
```

```
lrwxrwxrwx  1 root root    7 Oct 14 18:11 loop0p1 -> ../dm-0
```

```
lrwxrwxrwx  1 root root    7 Oct 14 18:11 loop0p2 -> ../dm-1
```

```
xen:/dev/mapper #
```

- From here, the same rules for the chroot setup apply

System crash

# System down

- Systems crash
- Possible reasons: hardware, firmware (Machine Check Exceptions), kernel or (3<sup>rd</sup> party) driver related issues, userspace applications like high availability daemons that fence systems for various reasons or hypervisor problems
- You can your best to prevent this from happening, but in case it does happen you should try to collect as much data as possible.

# System down

- Data such as `vmstat`, `iostat`, `swap` and `memory` usage is gathered by `sar` but only in 10 minutes intervals. They can indicate if the system was under a particular load or showed abnormal behavior at the time the issue happened. But they are not able to show exactly what happened.
- `/var/log/messages` or `warn` might contain hints, but in most cases nothing is printed to the default log files.
- The best source of information for a system crash is a kernel dump.

# System down

- If a server crashes and a kernel dump has not been configured, the chances are high you won't ever know why it crashed.
- Hence configuring kdump for production systems is highly recommended.
- If systems just freeze but don't write a dump, try to force a dump using Magic SysRQ key strokes.
- Further details in:  
TID 3374462 - Configure kernel core dump capture  
<https://www.suse.com/support/kb/doc.php?id=3374462>

Kdump setup

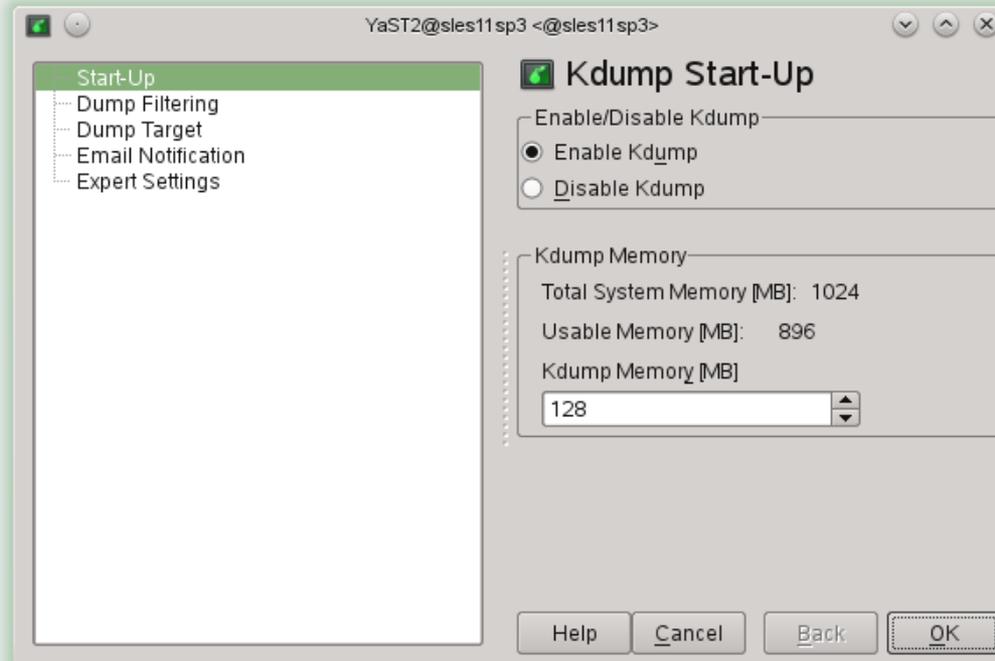
# What is kdump?

- Kdump is a crash dumping mechanism that allows you to capture the state of the kernel at the time it crashes.
- The main kernel reserves a memory region for the dump kernel. While kexec allows booting a new kernel “over” the running one, the crash state is preserved which is essential for taking the dump.
- Once loaded kdump will write the dump to the specified destination and reboot the system (if configured to do so).
- To check if the dump kernel loaded:  

```
sles11sp3:~ # grep Crash /proc/iomem  
27000000-2effffff : Crash kernel
```

# YaST2 kdump setup

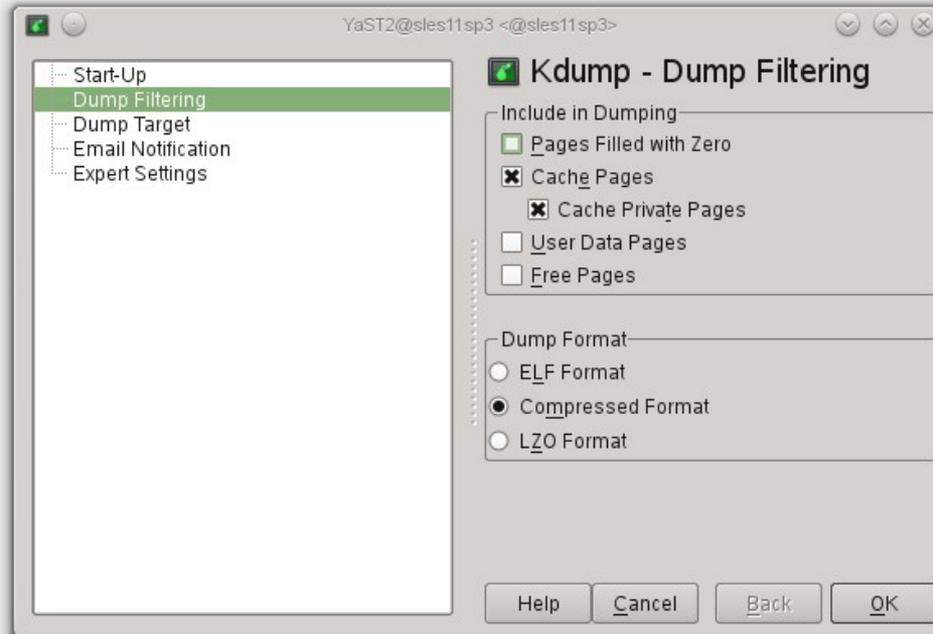
- Use the “yast2 kdump” module to activate kernel dumps:



- Enable the service and specify how much memory should be reserved. 128MB should be considered minimum.

# YaST2 kdump setup

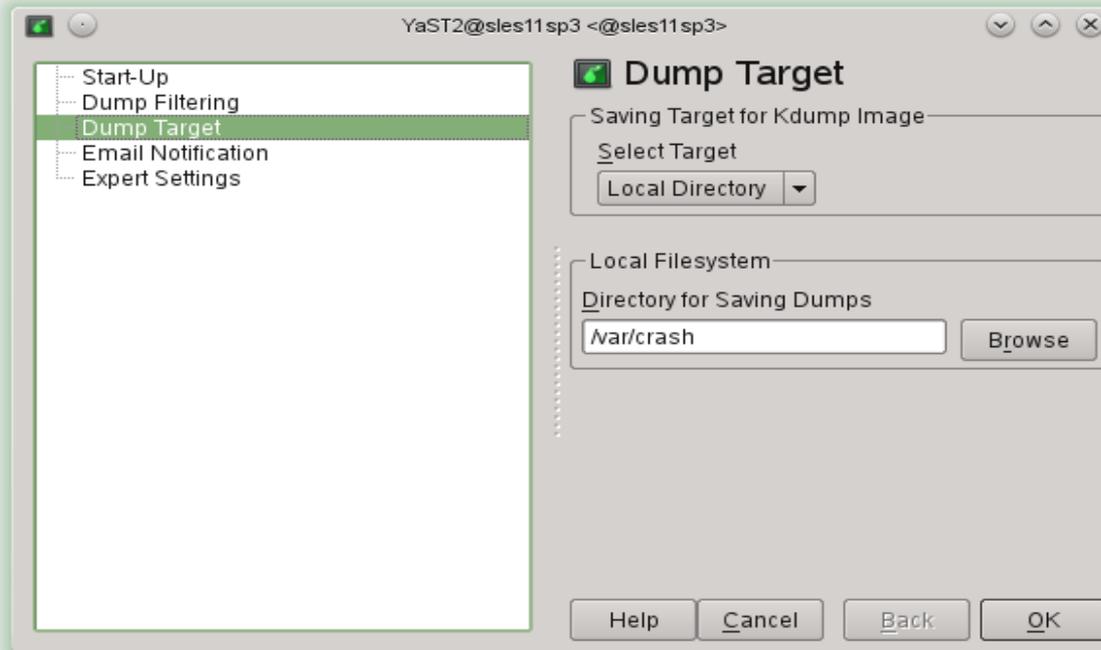
- On “Dump Filtering” you can specify which data should be captured. To minimize the dump as much as possible disable everything.



- XEN hypervisors need to be set to “ELF”, any other kernel may use “Compressed” or “LZO” format.

# YaST2 kdump setup

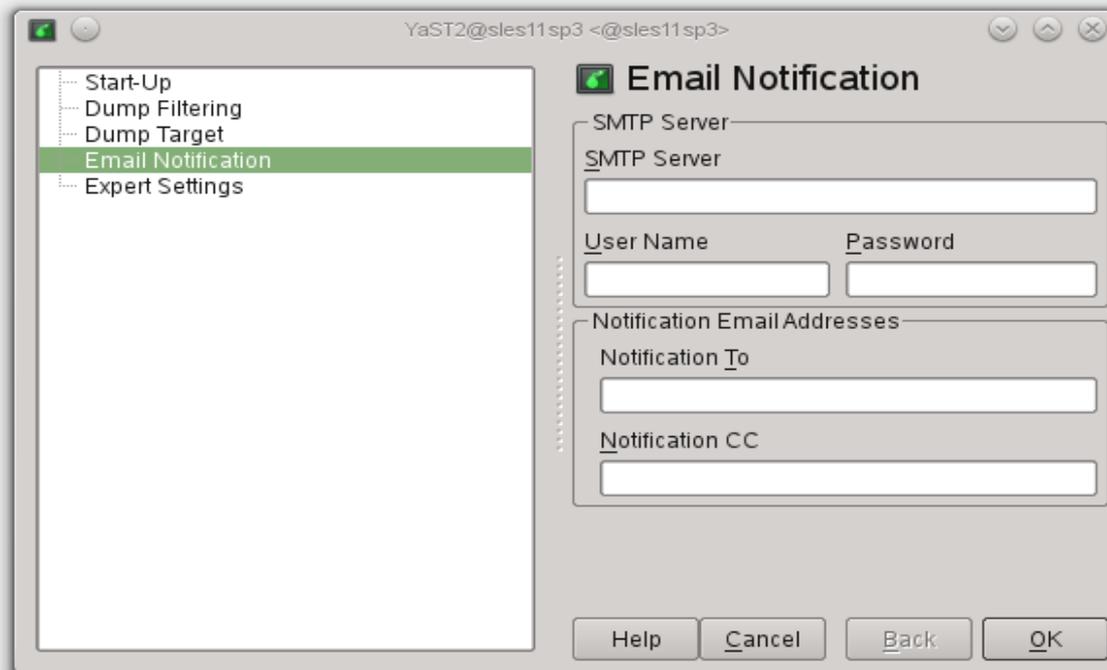
- With “Dump Target” you specify the location where the dump will be saved. Default is `/var/crash`



- The drop down provides further options for different protocol settings (ftp, ssh, nfs, cifs).

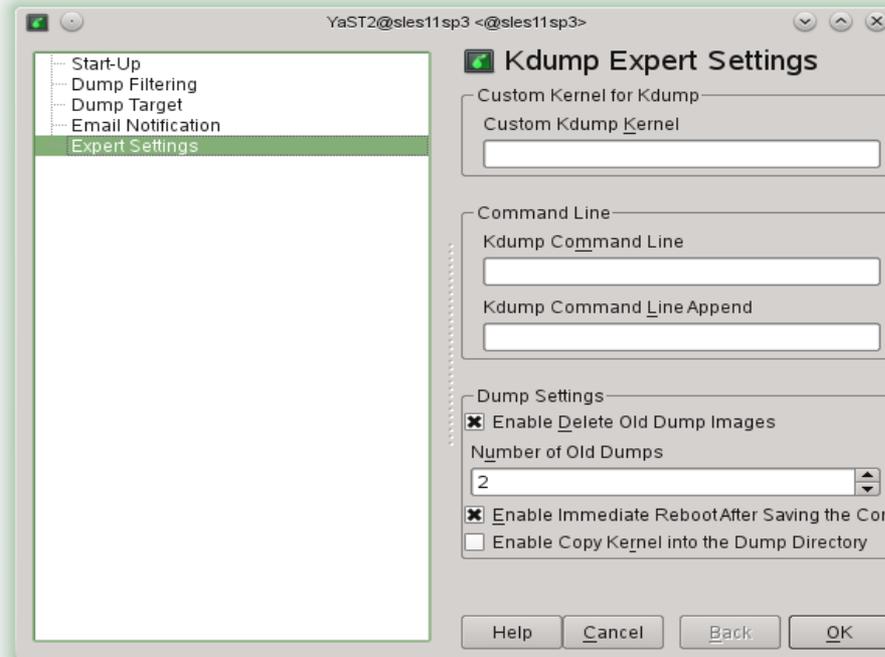
# YaST2 kdump setup

- “Email Notification” lets you specify who to notify in case of an event



# YaST2 kdump setup

- “Expert Settings” lets you configure the number of dumps kept, reboot settings and custom kernel parameters.



- Once you are done with your settings, reboot the server to activate the dump settings.

# YaST2 kdump or `/etc/sysconfig/kdump`?

- All settings can be set manually in `/etc/sysconfig/kdump`. YaST2 just edits this file and enables `/etc/init.d/boot.kdump`.
- Unchecking all dump options sets dump level to 31 which provides the smallest dump file. For more info on compression levels see `man 8 makedumpfile`.
- See for more information `/usr/src/linux/Documentation/kdump/kdump.txt` included in the appropriate kernel-source rpm
- Make sure to have enough disk space in `/var/crash`. Network dumps allow a convenient way to save the dump to cheaper storage.

# Dump settings verification

- After the reboot, check `/var/log/boot.msg` for `boot.kdump` messages.
- Run `grep Crash /proc/iomem` to verify the kdump kernel reserved a memory region.
- To make sure everything works as expected, crash your server manually using `echo c > /proc/sysrq-trigger`
- If everything is correct, the above command will “crash” the server, load the kdump environment, write the dump to `/var/crash` and reboot the server. Depending on your environment, it may be valuable to halt the system in the kdump environment until further investigation is done.

# Special considerations

- Creating a crashdump with VMware Workstation:  
<https://www.suse.com/support/kb/doc.php?id=7009637>
- Configure a XEN Virtual Machine (domu) for a kernel / core dump:  
<https://www.suse.com/support/kb/doc.php?id=7001384>

System down – and now?

# How to provide the files to support?

- Please note: There is no guarantee that the system will trigger kdump in case of a crash. It depends on how the system failed.
- All information required by support is documented in: <https://www.suse.com/support/kb/doc.php?id=7010249>
- If you have a kernel dump, there are two FTP servers available to upload the files:  
EMEA region: <ftp://support-ftp.suse.com/in>  
US region: <ftp://ftp.novell.com/incoming>  
See for best practice advice on uploading core files:  
<https://www.suse.com/support/kb/doc.php?id=7010056>

# How to provide the files to support?

- Once you have everything in place open a service request, provide the information requested in TID 7010249. Upload the supportconfig and the dump files.
- If the compressed dump file is quite big, it is recommended to split it into several smaller files.

Questions?

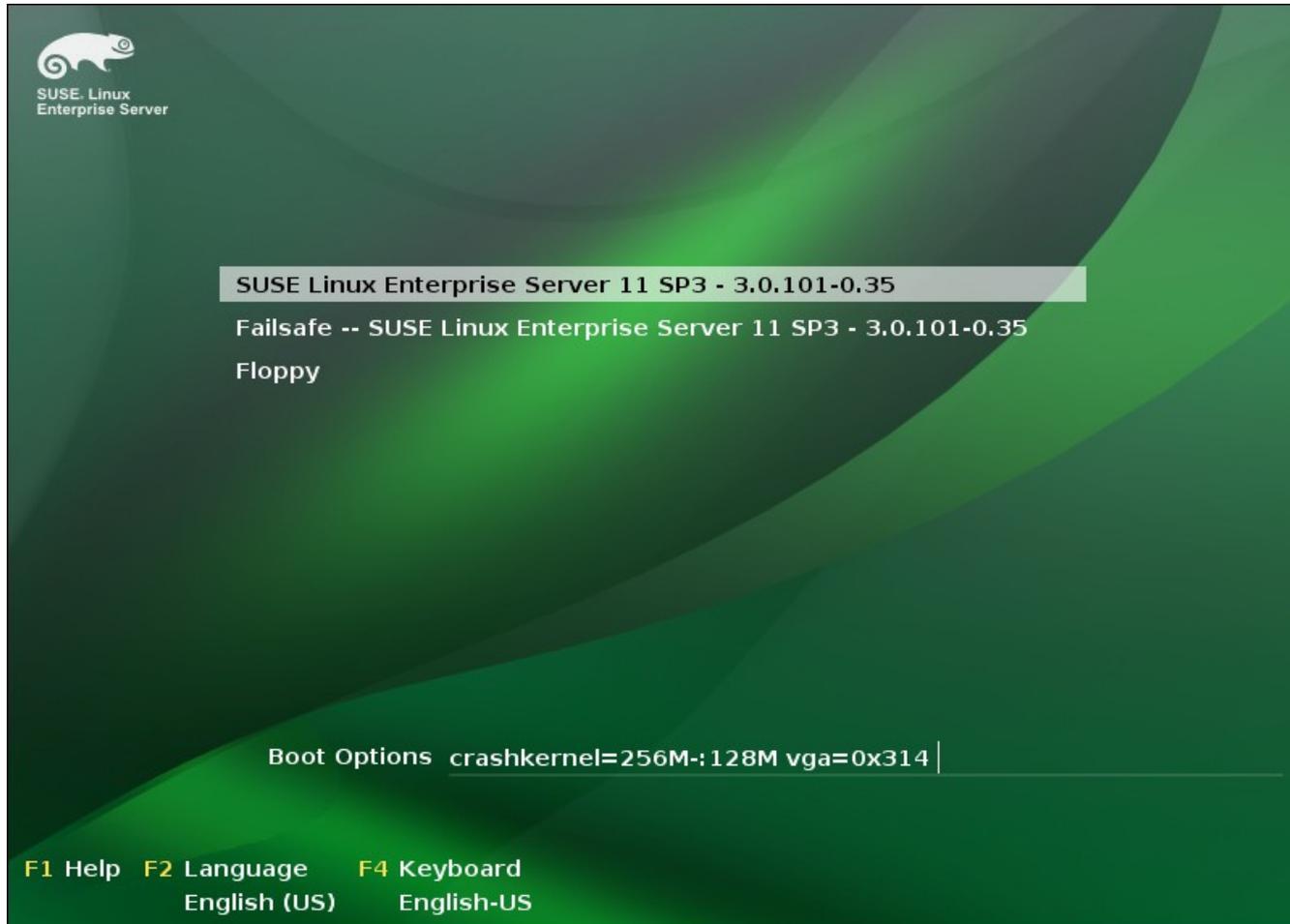
# Appendix

Grub shell

# The grub shell

- Provides limited access to boot loader settings and kernel options
- Options set here are valid for this boot process only, once the system is back online, `/boot/grub/menu.lst` needs to be adjusted manually or via `yast2`.
- Important config files:
  - `/boot/grub/menu.lst`
  - `/boot/grub/device.map`
  - `/etc/grub.conf`
- Reinstall grub in SLE11 rescue system:  
`grub -batch < /etc/grub.conf`

# The grub shell



- To enter the grub interactive shell press ESC

# A scenario

- Specified disk does not exist, due to type error in `/boot/grub/menu.lst`:

```
root (hd0,4)
```

```
Error 22: No such partition
```

```
Press any key to continue..._
```

# How to get the system back online

- Press enter to get back to main menu, select the entry to adjust and press “e” to edit it, like:

```
GNU GRUB  version 0.97  (635K lower / 784256K upper memory)

root (hd0,4)
kernel /boot/vmlinuz-3.0.101-0.35-default root=/dev/sda2 resume=/dev/→
initrd /boot/initrd-3.0.101-0.35-default
```

```
[ Minimal BASH-like line editing is supported.  For the first word, TAB
lists possible command completions.  Anywhere else TAB lists the possible
completions of a device/filename.  ESC at any time exits. ]

grub edit> root (hd0,1)
```

- When done, press enter to return to the previous menu and “b” to boot.

```
GNU GRUB  version 0.97  (635K lower / 784256K upper memory)

root (hd0,1)
kernel /boot/vmlinuz-3.0.101-0.35-default root=/dev/sda2 resume=/dev/→
initrd /boot/initrd-3.0.101-0.35-default
```

# Booting without a menu.lst entry

- Press “c” to get to the grub shell
- You need to specify:
  - the root device
  - the kernel
  - the initrd:

```
grub> root (hd0,1)
  Filesystem type is ext2fs, partition type 0x83

grub> kernel (hd0,1)/boot/vmlinuz
  [Linux-bzImage, setup=0x3c00, size=0x3c10e0]

grub> initrd (hd0,1)/boot/initrd
  [Linux-initrd @ 0x2f750000, 0x77fb70 bytes]

grub> _
```

- In the example above there is no separate boot partition, hence the path is `(hd0,1)/boot/vmlinuz`, assume `hd0,1` would be boot, the path would be `(hd0,1)/vmlinuz` and `(hd0,1)/initrd`

# SUSE Linux Enterprise Server 12

- Uses grub2
- On the boot menu either press “e” to edit or “c” to enter the grub shell
- When done, press `CTRL+x` or `F10` to boot the system

supportconfig options & examples

# Useful supportconfig options

- `-A` = all functions, full logging and RPM verification (this will take some time to complete)
- `-l` = also collects already rotated warn and messages files. By default, supportconfig only captures the last 5000 entries in a given log file. Especially inconvenient when working on system crash situations when the system rebooted and rotated logfiles
- `-ur` = upload the file directly into the service request with a given SR number. Outbound FTP traffic needs to be allowed for this to work in the firewall.
- `-x` = exclude certain checks, use `-F` to get a listing of available options

# Useful supportconfig options

- `-m` = collects only a minimal supportconfig (basic env, basic health, hardware, rpm, messages, y2logs)
- `-i` = include checks, may be used with `-m` to extend the basic data collection
- `-t` = specify alternate target directory for immediate troubleshooting, this will not create a compressed archive
- `--help` = complete list of options

# supportconfig examples

- `supportconfig` = gathers default data, up to 5000 entries for given logfiles
- `supportconfig -A` = like above plus RPM verification
- `supportconfig -lur 12345678901` = will gather additional logs and upload the tbz to the service request #12345678901
- `supportconfig -m -i DNS,MPIO` = minimal supportconfig plus DNS and multipath information
- What happens if supportconfig is stucked on a given task? `supportconfig -x MPIO`

# View a supportconfig

- Your favorite editor, grep and less are your best friend

- ```
sles11sp3:~/ # supportconfig -m -t my_server -i UPD,MPIO,ETC
sles11sp3:~/ # cd my_server/nts_sles11sp3_141008_1222
sles11sp3:~/my_server/nts_sles11sp3_141008_1222 # ls -l
total 5108
drwx----- 2 root root    4096 Oct  8 12:23 ./
drwx----- 3 root root    4096 Oct  8 12:22 ../
-rw----- 1 root root    3013 Oct  8 12:22 basic-environment.txt
-rw----- 1 root root   12877 Oct  8 12:22 basic-health-check.txt
-rw----- 1 root root  244440 Oct  8 12:22 etc.txt
-rw----- 1 root root 1107640 Oct  8 12:23 hardware.txt
-rw----- 1 root root 3192285 Oct  8 12:23 messages.txt
-rw----- 1 root root  103671 Oct  8 12:22 mpio.txt
-rw----- 1 root root  189628 Oct  8 12:22 rpm.txt
-rw----- 1 root root    5061 Oct  8 12:23 summary.xml
-rw----- 1 root root    6481 Oct  8 12:23 supportconfig.txt
-rw----- 1 root root  309846 Oct  8 12:22 y2log.txt
```
- ```
sles11sp3:~/my_server/nts_sles11sp3_141008_1222 # grep -l -E "ntp.conf|multipath
-l" *
basic-health-check.txt
etc.txt
mpio.txt
y2log.txt
```

Boot a rescue system via PXE/TFTP

# PXE/TFTP bootable rescue system

- Requirements:

- a SUSE Linux Enterprise Server
- Packages: atftp, dhcpd, apache2
- approximately 20GB disk space (depending how many rescue systems you would like to make available)
- Setup atftp & dhcpd according to:  
[https://en.opensuse.org/SDB:PXE\\_boot\\_installation](https://en.opensuse.org/SDB:PXE_boot_installation)
- Copy the ISO to the server and loop mount it into  
`/srv/www/htdocs/install/<version>`. This can be done  
via `/etc/fstab` with the following entry:  
`/path/SLES-11-SP3-DVD-x86_64-GM-DVD1.iso \`  
`/srv/www/htdocs/install/sles11sp3 iso9660 \`  
`bind,ro 0 0`

# PXE/TFTP bootable rescue system

- Requirements:

- If you like to use symbolic links and browsing in apache, edit `/etc/apache2/default-server.conf` and change line 22 from  
Options None  
to  
Options Indexes,FollowSymLinks  
and restart apache2
- In `/srv/tftpboot` create a code base subdirectory like “sle11sp3” and copy the linux and initrd files to the subdirectory. This structures the setup in case you like to have different code bases running on the same server.

# PXE/TFTP bootable rescue system

- Edit `/srv/tftboot/pxelinux.cfg/default` and enter the following settings:

```
UI menu.c32  
default harddisk  
prompt 1  
timeout 500
```

```
label rescue  
    kernel sle11sp3/linux  
    append initrd=sle11sp3/initrd \  
    install=http://192.168.67.10/pub/install/sles11sp3-x86_64 rescue=1
```

How to open a kdump

# How do I open the dump myself?

- To open the dump three files are needed:
  1. the vmcore file from `/var/crash/<date-dir>`
  2. the `vmlinuz*gz` file of that system
  3. the `vmlinuz*-debug` file
- The matching kernel-debug file can be retrieved from the respective debuginfo channel. The name of the file would be:  
kernel-\$flavor-debuginfo-version.\$arch.rpm, e.g.:  
kernel-default-debuginfo-3.0.101-0.35.1.x86\_64.rpm
- Please keep in mind the package download location depends on the patch level. For unpatched SP-GA level you will find the package in the Debuginfo-Pool channel, for patched systems look in the Updates channel.

# How do I open the dump myself?

- Hint: Before you attempt to load the dump yourself, check the dmesg file in the crash directory for hints. It contains the same information as the log command inside crash and will save you some time if you need a fast peek at the issue.

# How do I open the dump myself?

- 1. Copy the kernel-`*debuginfo*`rpm to the crash directory
- 2. extract the rpm here with:  
`rpm2cpio <filename> | cpio -vid`
- 3. Install the crash utility (if not already installed):  
`zypper in crash`
- 4. Assuming you are in `/var/crash/<date>` open the dump using:  
`crash /boot/vmlinux-3.0.101-0.35-default.gz usr/lib/debug/boot/vmlinux-3.0.101-0.35-default.debug vmcore`

# How do I open the dump myself?

- sles11sp3:/var/crash/2014-10-09-12:49 # crash /boot/vmlinuz-3.0.101-0.35-default.gz usr/lib/debug/boot/vmlinuz-3.0.101-0.35-default.debug vmcore  
crash 6.0.7  
Copyright (C) 2002-2012 Red Hat, Inc.  
Copyright (C) 2004, 2005, 2006, 2010 IBM Corporation  
Copyright (C) 1999-2006 Hewlett-Packard Co  
Copyright (C) 2005, 2006, 2011, 2012 Fujitsu Limited  
Copyright (C) 2006, 2007 VA Linux Systems Japan K.K.  
Copyright (C) 2005, 2011 NEC Corporation  
Copyright (C) 1999, 2002, 2007 Silicon Graphics, Inc.  
Copyright (C) 1999, 2000, 2001, 2002 Mission Critical Linux, Inc.  
This program is free software, covered by the GNU General Public License,  
and you are welcome to change it and/or distribute copies of it under  
certain conditions. Enter "help copying" to see the conditions.  
This program has absolutely no warranty. Enter "help warranty" for details.

...continue next page

# How do I open the dump myself?

- ...continued output:

```
crash: /boot/vmlinux-3.0.101-0.35-default.gz: original filename unknown
  Use "-f /boot/vmlinux-3.0.101-0.35-default.gz" on command line to prevent this message.
GNU gdb (GDB) 7.3.1
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu"...
  KERNEL: /boot/vmlinux-3.0.101-0.35-default.gz
  DEBUGINFO: usr/lib/debug/boot/vmlinux-3.0.101-0.35-default.debug
  DUMPFILE: vmcore [PARTIAL DUMP]
  CPUS: 1
  DATE: Thu Oct  9 12:49:29 2014
  UPTIME: 05:45:41
  LOAD AVERAGE: 0.08, 0.23, 0.30
  TASKS: 172
  NODENAME: sles11sp3
  RELEASE: 3.0.101-0.35-default
  VERSION: #1 SMP Wed Jul 9 11:43:04 UTC 2014 (c36987d)
  MACHINE: x86_64 (3591 Mhz)
  MEMORY: 1 GB
  PANIC: "[20732.368347] Oops: 0002 [#1] SMP " (check log for details)
  PID: 5618
  COMMAND: "bash"
  TASK: ffff88003b9ba300 [THREAD_INFO: ffff88003c8ee000]
  CPU: 0
  STATE: TASK_RUNNING (PANIC)
```

# How do I open the dump myself?

- Once you loaded the dump, the very first you will see is the dump summary
- From here the most common commands are `sys` (system information), `bt` (backtrace), `log` (display the kernel system message buffer) and `kmem` (display system memory information) to gather more information about the crash.
- An exhaustive explanation of how to analyze kernel dumps is not part of this presentation.

# Useful technical information documents

# Useful technical information documents

- System crash or unexpected reboot - Which info is needed by Novell Technical Services for a root cause analysis?  
<https://www.suse.com/support/kb/doc.php?id=7010249>
- Configure kernel core dump capture  
<https://www.suse.com/support/kb/doc.php?id=3374462>
- Configure a XEN Virtual Machine (domu) for a kernel / core dump  
<https://www.suse.com/support/kb/doc.php?id=7001384>
- Creating a crashdump with VMware Workstation  
<https://www.suse.com/support/kb/doc.php?id=7009637>
- Generating a Kernel Core Dump Analysis File  
<https://www.suse.com/support/kb/doc.php?id=7010484>
- Best practice for providing kernel core dumps to support incidents  
<https://www.suse.com/support/kb/doc.php?id=7010056>

# Useful technical information documents

- Generating a Kernel Core Dump running cluster HAE (High Availability Extension)  
<https://www.suse.com/support/kb/doc.php?id=7015029>
- Configuring a Remote Serial Console for SLES  
<https://www.suse.com/support/kb/doc.php?id=3456486>
- LVM symbolic links /dev/\$VGNAME/\$LVNAME unavailable in SLE11SP3 rescue system  
<https://www.suse.com/support/kb/doc.php?id=7014112>
- XEN: Directly attaching Xen DomU disk images to Dom0 for system rescue  
<https://www.suse.com/support/kb/doc.php?id=3312774>
- Reinstalling GRUB from the Rescue System  
<https://www.suse.com/support/kb/doc.php?id=7008279>
- Troubleshooting Common Boot Issues  
<https://www.suse.com/support/kb/doc.php?id=3864925>

# Useful technical information documents

- Cloned machine does not boot up properly  
<https://www.suse.com/support/kb/doc.php?id=3048119>
- Tainted Kernel  
<https://www.suse.com/support/kb/doc.php?id=3582750>
- For anything else:  
<https://www.suse.com/support/>



## **Unpublished Work of SUSE LLC. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

## **General Disclaimer**

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

