NAME

kdump.conf − configuration file for kdump kernel.

DESCRIPTION

kdump.conf is a configuration file for the kdump kernel crash collection service.

kdump.conf provides post-kexec instructions to the kdump kernel. It is stored in the initrd file managed by the kdump service. If you change this file and do not want to restart before it takes effect, restart the kdump service to rebuild to initrd.

For most configurations, you can simply review the examples provided in the stock /etc/kdump.conf.

NOTE: kdump.conf only affects the behavior of the initramfs. Please read the kdump operational flow section of kexec-kdump-howto.txt in the docs to better understand how this configuration file affects the behavior of kdump.

OPTIONS

path <path>

Append path to the filesystem device which you are dumping to. Ignored for raw device dumps. If unset, will default to /var/crash.

core_collector <command> <options>

This allows you to specify the command to copy the vmcore. You could use the dump filtering program makedumpfile, the default one, to retrieve your core, which on some arches can drastically reduce core file size. See /usr/sbin/makedumpfile --help for a list of options. Note that the -i and -g options are not needed here, as the initrd will automatically be populated with a config file appropriate for the running kernel.

If you use other command as a core_collector, probably you also need to provide its full path in extra_bins. Note, for ssh dump, scp should be used instead of cp.

raw <partition>

Will write /proc/vmcore into raw <partition>.

nfs <nfs mount> | nfs4 <nfs mount> | net <nfs mount>

Will mount fs and copy /proc/vmcore to <mnt>/<path>/%HOST-%DATE/, supports DNS. Note that a fqdn should be used as the server name in the mount point. By default, NFSv3 is used, nfs4 will use NFSv4. "net" is deprecated by "nfs"

ssh <user@server> | net <user@server>

Will copy /proc/vmcore to <user@server>:<path>/%HOST-%DATE/ via SSH, supports DNS. If makedumpfile is the core_collector, it is piped to an "ssh" shell, otherwise use the specified core_collector like scp. NOTE: make sure user has necessary write permissions on server and that a fqdn is used as the server name. "net" is deprecated by "ssh"

<fs type> <partition>

Will mount -t <fs type> <partition> /mnt and copy /proc/vmcore to /mnt/<path>/127.0.0.1-%DATE/. Currently, the supported fs types are: ext4, ext3, ext2, minix, btrfs and xfs, other fs types like "auto" are not supported. NOTE: <partition> can be a device node, label or uuid.

disk_timeout <seconds>

Number of seconds to wait for disks to appear prior to continue to save dump. By default kdump waits 180 seconds for the disks to show up it needs. This can be useful in some cases if disk never shows up (Either because disk was removed or because kdump is waiting on wrong disk).

link_delay <seconds>

Some network cards take a long time to initialize, and some spanning tree enabled networks do not transmit user traffic for long periods after a link state changes. This optional parameter defines a wait period after a link is activated in which the initramfs will wait before attempting to transmit user data.

kdump_post <binary | script>

This directive allows you to run a specified executable just after the memory dump process terminates. The exit status from the dump process is fed to the kdump_post executable, which can be used to trigger different actions for success or failure.

Note that scripts written for use with this directive must use the /bin/hush interpreter

kdump_pre <binary | script>

Works just like the kdump_post directive, but instead of running after the dump process, runs immediately before. Exit status of this binary is interpreted as follows:

0 - continue with dump process as usual

non 0 - reboot/halt the system

Note that scripts written for this directive must use the /bin/hush interpreter

extra_bins <binaries | shell scripts>

This directive allows you to specify additional binaries or shell scripts you'd like to include in your kdump initrd. Generally only useful in conjunction with a kdump_post binary or script that relies on other binaries or scripts.

extra_modules <module(s)>

This directive allows you to specify extra kernel modules that you want to be loaded in the kdump initrd, typically used to set up access to non-boot-path dump targets that might otherwise not be accessible in the kdump environment. Multiple modules can be listed, separated by a space, and any dependent modules will automatically be included. Module name should be specified without ".ko" suffix.

options <module> <option list>

This directive allows you to specify options to apply to modules in the initramfs. This directive overrides options specified in /etc/modprobe.conf. Module name should be specified without ".ko" suffix.

blacklist <module>

The blacklist keyword indicates that all of that particular modules are to be ignored in the initramfs. General terminology for blacklist has been that module is present in initramfs but it is not actually loaded in kernel. This directive can be specified multiple times or as a space separated list. Module name should be specified without ".ko" suffix.

sshkey <path>

Specifies the path of the ssh key you want to use when do ssh dump, the default value is /root/.ssh/kdump_id_rsa.

default <reboot | halt | poweroff | shell | mount_root_run_init>

Action to perform in case dumping to intended target fails. If no default action is specified, "reboot" is assumed default.

reboot: If the default action is reboot simply reboot the system (this is what most people will want, as it returns the system to a nominal state).

halt: bring the system to a halt, requiring manual reset.

poweroff: The system will be powered down.

shell: If the default action is shell, then drop to an hush session inside the initramfs from where you can manually perform additional recovery actions. Exiting this shell reboots the system.

mount_root_run_init: Mount root filesystem and run init. Kdump initscript will try to save dump to root filesystem in /var/crash dir. This will likely require a lot more memory to be reserved for kdump kernel.

debug_mem_level <0-3>

Turns on debug/verbose output of kdump scripts regarding free/used memory at various points of execution. Higher level means more debugging output.

0 - no output

1 - partial /proc/meminfo

2 - /proc/meminfo

3 - /proc/meminfo + /proc/slabinfo

force_rebuild <0 | 1>

By default, kdump initrd only will be rebuilt when necessary. Specify 1 here to force rebuilding kdump initrd every time when kdump service starts.

fence_kdump_args <arg(s)>

Command line arguments for fence_kdump_send (it can contain all valid arguments except hosts to send notification to).

fence_kdump_nodes <node(s)>

List of cluster node(s) separated by space to send fence_kdump notification to (this option is mandatory to enable fence_kdump).

EXAMPLES

see */etc/kdump.conf*

SEE ALSO

kexec(8) mkdumprd(8)