Rocky Enterprise Linux 9.2 Manual Pages on command 'kdump.conf.5'

$ man kdump.conf.5

KDUMP.CONF(5) File Formats Manual KDUMP.CONF(5)

NAME

kdump.conf - configuration file for kdump kernel.

DESCRIPTION

kdump.conf is a configuration file for the kdump kernel crash collec?

tion 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 reboot in order for the changes to take

effect, restart the kdump service to rebuild the initrd.

For most configurations, you can simply review the examples provided in

the stock /etc/kdump.conf.

NOTE: For filesystem dumps the dump target must be mounted before

building kdump initramfs.

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

auto_reset_crashkernel <yes|no>

determine whether to reset kernel crashkernel to new default value or not when kexec-tools updates the default crashkernel value and existing kernels using the old default kernel crashk? ernel value

raw <partition>

Will dd /proc/vmcore into <partition>. Use persistent device names for partition devices, such as /dev/vg/<dev? name>.

nfs <nfs mount>

Will mount nfs to <mnt>, 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.

ssh <user@server>

Will save /proc/vmcore through ssh pipe to <user@server>:<path>/%HOST-%DATE/, supports DNS. NOTE: make sure user has necessary write permissions on server and that a fqdn is used as the server name.

sshkey <path>

Specify the path of the ssh key to use when dumping via ssh. The default value is /root/.ssh/kdump_id_rsa.

<fs type> <partition>

Will mount -t <fs type> <partition> <mnt>, and copy /proc/vmcore to <mnt>/<path>/%HOST_IP-%DATE/. NOTE: <partition> can be a device node, label or uuid. It's recommended to use persistent device names such as /dev/vg/<devname>. Otherwise it's suggested to use label or uuid.

path <path>

"path" represents the file system path in which vmcore will be saved. If a dump target is specified in kdump.conf, then "path" is relative to the specified dump

target.

Interpretation of "path" changes a bit if the user didn't specify any dump target explicitly in kdump.conf. In this case, "path" represents the absolute path from root. The dump target and adjusted path are arrived at automati? cally depending on what's mounted in the current system.

Ignored for raw device dumps. If unset, will use the de? fault "/var/crash".

core_collector <command> <options>

This allows you to specify the command to copy the vm? core. The default is makedumpfile, which on some archi? tectures can drastically reduce core file size. See /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.

Note 1: About default core collector: The default core_collector for raw/ssh dump is: "makedumpfile -F -l --message-level 7 -d 31". The default core_collector for other targets is: "makedumpfile -l --message-level 7 -d 31". Even if core_collector option is commented out in kdump.conf, makedumpfile is the default core collector and kdump uses it internally. If one does not want make? dumpfile as default core_collector, then they need to specify one using core_collector option to change the be? havior.

Note 2: If "makedumpfile -F" is used then you will get a flattened format vmcore.flat, you will need to use "make? dumpfile -R" to rearrange the dump data from standard in? put to a normal dumpfile (readable with analysis tools). ie. "makedumpfile -R vmcore < vmcore.flat"

Note 3: If specified core_collector simply copy the vm? core file to the dump target (eg: cp, scp), the vmcore

could be significantly large. Please make sure the dump target has enough space, at leaset larger than the sys? tem's RAM.

kdump_post <binary | script>

This directive allows you to run a specified executable just after the vmcore dump process terminates. The exit status of the current dump process is fed to the kdump_post executable as its first argument($1). Exe? cutable can modify it to indicate the new exit status of succeeding dump process,

All files under /etc/kdump/post.d are collectively sorted and executed in lexical order, before binary or script specified kdump_post parameter is executed.

Note that scripts written for use with this directive must use the /bin/bash interpreter. And since these scripts run in kdump enviroment, the reference to the storage or network device in the scripts should adhere to the section

kdump_pre <binary | script>

Works just like the "kdump_post" directive, but instead of running after the dump process, runs immediately be? fore. Exit status of this binary is interpreted as fol? lows:

0 - continue with dump process as usual

non 0 - run the final action (reboot/poweroff/halt)

All files under /etc/kdump/pre.d are collectively sorted and executed in lexical order, after binary or script specified kdump_pre parameter is executed. Even if the binary or script in /etc/kdump/pre.d directory returns non 0 exit status, the processing is continued.

Note that scripts written for use with this directive must use the /bin/bash interpreter. And since these scripts run in kdump enviroment, the reference to the

storage or network device in the scripts should adhere to the section

extra_bins <binaries | shell scripts>

This directive allows you to specify additional binaries or shell scripts you'd like to include in your kdump ini? trd. 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 environ? ment. Multiple modules can be listed, separated by spa? ces, and any dependent modules will automatically be in? cluded.

failure_action <reboot | halt | poweroff | shell | dump_to_rootfs>

Action to perform in case dumping to the intended target fails. The default is "reboot". reboot: Reboot the sys? tem (this is what most people will want, as it returns the system to a normal state). halt: Halt the system and lose the vmcore. poweroff: The system will be powered down. shell: Drop to a shell session inside the initramfs, from which you can manually perform additional recovery actions. Exiting this shell reboots the system by default or performs "final_action". Note: kdump uses bash as the default shell. dump_to_rootfs: If non-root dump target is specified, the failure action can be set as dump_to_rootfs. That means when dumping to target fails, dump vmcore to rootfs from initramfs context and reboot by default or perform "final_action".

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

Same as the "failure_action" directive above, but this directive is obsolete and will be removed in the future.

final_action <reboot | halt | poweroff>

Action to perform in case dumping to the intended target succeeds. Also performed when "shell" or "dump_to_rootfs" failure action finishes. Each action is same as the "failure_action" directive above. The de? fault is "reboot".

force_rebuild <0 | 1>

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

force_no_rebuild <0 | 1>

By default, kdump initrd will be rebuilt when necessary. Specify 1 to bypass rebuilding of kdump initrd. force_no_rebuild and force_rebuild options are mutually exclusive and they should not be set to 1 simultaneously.

override_resettable <0 | 1>

Usually an unresettable block device can't be a dump tar? get. Specifying 1 means that even though the block target is unresettable, the user wants to try dumping anyway. By default, it's set to 0, which will not try something destined to fail.

dracut_args <arg(s)>

Kdump uses dracut to generate initramfs for second ker? nel. This option allows a user to pass arguments to dra? cut directly.

fence_kdump_args <arg(s)>

Command line arguments for fence_kdump_send (it can con? tain all valid arguments except hosts to send notifica? tion to).

fence_kdump_nodes <node(s)>

List of cluster node(s) except localhost, separated by

spaces, to send fence_kdump notification to (this option
is mandatory to enable fence_kdump).

DEPRECATED OPTIONS

net <nfs mount>|<user@server>

net option is replaced by nfs and ssh options. Use nfs or ssh options directly.

options <module> <option list>

Use KDUMP_COMMANDLINE_APPEND in /etc/sysconfig/kdump to add mod?ule options as kernel command line parameters. For example, specify 'loop.max_loop=1' to limit maximum loop devices to 1.

link_delay <seconds>

link_delay was used to wait for a network device to initialize before using it. Now dracut network module takes care of this issue automatically.

disk_timeout <seconds>

Similar to link_delay, dracut ensures disks are ready before kdump uses them.

debug_mem_level <0-3>

Turn on verbose debug output of kdump scripts regarding free/used memory at various points of execution. This feature has been moved to dracut now. Use KDUMP_COMMANDLINE_APPEND in /etc/sysconfig/kdump and append dracut cmdline param rd.memde?bug=[0-3] to enable the debug output.

Higher level means more debugging output.

0 - no output

1 - partial /proc/meminfo

2 - /proc/meminfo

3 - /proc/meminfo + /proc/slabinfo

blacklist <list of kernel modules>

blacklist option was recently being used to prevent loading mod?ules in initramfs. General terminology for blacklist has been that module is present in initramfs but it is not actually loaded in kernel. Hence retaining blacklist option creates more

confusing behavior. It has been deprecated.

Instead, use rd.driver.blacklist option on second kernel to blacklist a certain module. One can edit /etc/syscon? fig/kdump.conf and edit KDUMP_COMMANDLINE_APPEND to pass kernel command line options. Refer to dracut.cmdline man page for more details on module blacklist option.

EXAMPLES

Here are some examples for core_collector option:

Core collector command format depends on dump target type. Typically for filesystem (local/remote), core_collector should accept two argu? ments. First one is source file and second one is target file. For ex.

ex1. core_collector "cp --sparse=always"

Above will effectively be translated to:

cp --sparse=always /proc/vmcore <dest-path>/vmcore

ex2. core_collector "makedumpfile -l --message-level 7 -d 31"

Above will effectively be translated to:

makedumpfile -l --message-level 7 -d 31 /proc/vmcore <dest-path>/vmcore

For dump targets like raw and ssh, in general, core collector should expect one argument (source file) and should output the processed core on standard output (There is one exception of "scp", discussed later). This standard output will be saved to destination using appropriate commands.

raw dumps examples:

ex3. core_collector "cat"

Above will effectively be translated to.

cat /proc/vmcore | dd of=<target-device>

ex4. core_collector "makedumpfile -F -l --message-level 7 -d 31"

Above will effectively be translated to.

makedumpfile -F -l --message-level 7 -d 31 | dd of=<target-de? vice>

ssh dumps examples

ex5. core_collector "cat"

Above will effectively be translated to.

cat /proc/vmcore | ssh <options> <remote-location> "dd of=path/vmcore"

ex6. core_collector "makedumpfile -F -l --message-level 7 -d 31"

Above will effectively be translated to.

makedumpfile -F -l --message-level 7 -d 31 | ssh <options> <re?mote-location> "dd of=path/vmcore"

There is one exception to standard output rule for ssh dumps. And that is scp. As scp can handle ssh destinations for file transfers, one can specify "scp" as core collector for ssh tar?gets (no output on stdout).

ex7. core_collector "scp"

Above will effectively be translated to.

scp /proc/vmcore <user@host>:path/vmcore

examples for other options please see /etc/kdump.conf

SEE ALSO

kexec(8) mkdumprd(8) dracut.cmdline(7)