Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'systemd-repart.service.8'

$ man systemd-repart.service.8

SYSTEMD-REPART(8)   systemd-repart   SYSTEMD-REPART(8)

NAME

systemd-repart, systemd-repart.service - Automatically grow and add partitions

SYNOPSIS

systemd-repart [OPTIONS...] [[BLOCKDEVICE]...]

systemd-repart.service

DESCRIPTION

systemd-repart grows and adds partitions to a partition table, based on the configuration files described in repart.d(5).

If invoked with no arguments, it operates on the block device backing the root file system partition of the running OS, thus growing and adding partitions of the booted OS image itself. If --image= is used it will operate on the specified image file. When called in the initrd it operates on the block device backing /sysroot/ instead, i.e. on the block device the system will soon transition into. The systemd-repart.service service is generally run at boot in the initrd, in order to augment the partition table of the OS before its partitions are mounted. systemd-repart (mostly) operates in a purely incremental mode: it only grows existing and adds new partitions; it does not shrink, delete or move existing partitions. The service is intended to be run on every boot, but when it detects that the partition table already matches the installed repart.d/*.conf configuration files, it executes no operation.

systemd-repart is intended to be used when deploying OS images, to automatically adjust them to the system they are running on, during first boot. This way the deployed image can be minimal in size and may be augmented automatically at boot when needed, taking possession of disk space available but not yet used. Specifically the following use cases are among those covered:

?   The root partition may be grown to cover the whole available disk space.

?   A /home/, swap or /srv/ partition can be added.

?   A second (or third, ...) root partition may be added, to cover A/B style setups where a second version of the root file system is alternatingly used for implementing update schemes. The deployed image would carry only a single partition ("A") but on first boot a second partition ("B") for this purpose is automatically created.

The algorithm executed by systemd-repart is roughly as follows:

1. The repart.d/*.conf configuration files are loaded and parsed, and ordered by filename (without the directory prefix). For each configuration file, drop-in files are looked for in directories with same name as the configuration file with a suffix ".d" added.
2. The partition table already existing on the block device is loaded and parsed.
3. The existing partitions in the partition table are matched up with the repart.d/*.conf files by GPT partition type UUID. The first existing partition of a specific type is assigned the first configuration file declaring the same type. The second existing partition of a specific type is then assigned the second configuration file declaring the same type, and so on. After this

iterative assigning is complete any left-over existing partitions that have no matching configuration file are considered "foreign" and left as they are. And any configuration files for which no partition currently exists are understood as a request to create such a partition.

4. Taking the size constraints and weights declared in the configuration files into account, all partitions that shall be created are now allocated to the disk, taking up all free space, always respecting the size and padding requests. Similarly, existing partitions that should be grown are grown. New partitions are always appended to the end of the partition table, taking the first partition table slot whose index is greater than the indexes of all existing partitions. Partition table slots are never reordered and thus partition numbers are ensured to remain stable. Note that this allocation happens in memory only, the partition table on disk is not updated yet.
5. All existing partitions for which configuration files exist and which currently have no GPT partition label set will be assigned a label, either explicitly configured in the configuration or ? if that's missing ? derived automatically from the partition type. The same is done for all partitions that are newly created. These assignments are done in memory only, too, the disk is not updated yet.
6. Similarly, all existing partitions for which configuration files exist and which currently have an all-zero identifying UUID will be assigned a new UUID. This UUID is cryptographically hashed from a common seed value together with the partition type UUID (and a counter in case multiple partitions of the same type are defined), see below. The same is done for all partitions that are created anew. These assignments are done in memory only, too, the disk is not updated yet.
7. Similarly, if the disk's volume UUID is all zeroes it is also initialized, also cryptographically hashed from the same common

seed value. This is done in memory only too.

8. The disk space assigned to new partitions (i.e. what was previously free space) is now erased. Specifically, all file system signatures are removed, and if the device supports it, the BLKDISCARD I/O control command is issued to inform the hardware that the space is now empty. In addition any "padding" between partitions and at the end of the device is similarly erased.
9. The new partition table is finally written to disk. The kernel is asked to reread the partition table.

As exception to the normally strictly incremental operation, when called in a special "factory reset" mode, systemd-repart may also be used to erase existing partitions to reset an installation back to vendor defaults. This mode of operation is used when either the --factory-reset=yes switch is passed on the tool's command line, or the systemd.factory_reset=yes option specified on the kernel command line, or the FactoryReset EFI variable (vendor UUID 8cf2644b-4b0b-428f-9387-6d876050dc67) is set to "yes". It alters the algorithm above slightly: between the 3rd and the 4th step above any partition marked explicitly via the FactoryReset= boolean is deleted, and the algorithm restarted, thus immediately re-creating these partitions anew empty.

Note that systemd-repart only changes partition tables, it does not create or resize any file systems within these partitions. A separate mechanism should be used for that, for example systemd-growfs(8) and systemd-makefs.

The UUIDs identifying the new partitions created (or assigned to existing partitions that have no UUID yet), as well as the disk as a whole are hashed cryptographically from a common seed value. This seed value is usually the machine-id(5) of the system, so that the machine ID reproducibly determines the UUIDs assigned to all partitions. If the machine ID cannot be read (or the user passes --seed=random, see below) the seed is generated randomly instead, so that the partition UUIDs are also effectively random. The seed value may also be set explicitly, formatted as UUID via the --seed= option. By hashing these UUIDs from a common seed images prepared with this tool become reproducible and the result of the algorithm above deterministic.

The positional argument should specify the block device to operate on. Instead of a block device node path a regular file may be specified too, in which case the command operates on it like it would if a loopback block device node was specified with the file attached. If --empty=create is specified the specified path is created as regular file, which is useful for generating disk images from scratch.

OPTIONS

The following options are understood:

--dry-run=

Takes a boolean. If this switch is not specified --dry-run=yes is the implied default. Controls whether systemd-repart executes the requested re-partition operations or whether it should only show what it would do. Unless --dry-run=no is specified systemd-repart will not actually touch the device's partition table.

--empty=

Takes one of "refuse", "allow", "require", "force" or "create". Controls how to operate on block devices that are entirely empty, i.e. carry no partition table/disk label yet. If this switch is not specified the implied default is "refuse".

If "refuse" systemd-repart requires that the block device it shall operate on already carries a partition table and refuses operation if none is found. If "allow" the command will extend an existing partition table or create a new one if none exists. If "require" the command will create a new partition table if none exists so far, and refuse operation if one already exists. If "force" it will create a fresh partition table unconditionally, erasing the disk fully in effect. If "force" no existing partitions will be taken into account or survive the operation. Hence: use with care, this is a great way to lose all your data. If "create" a new loopback file is create under the path passed via the device node parameter, of the size indicated with --size=, see below.

--discard=

Takes a boolean. If this switch is not specified --discard=yes is
the implied default. Controls whether to issue the BLKDISCARD I/O
control command on the space taken up by any added partitions or on
the space in between them. Usually, it's a good idea to issue this
request since it tells the underlying hardware that the covered
blocks shall be considered empty, improving performance. If
operating on a regular file instead of a block device node, a
sparse file is generated.

--size=

Takes a size in bytes, using the usual K, M, G, T suffixes, or the
special value "auto". If used the specified device node path must
refer to a regular file, which is then grown to the specified size
if smaller, before any change is made to the partition table. If
specified as "auto" the minimal size for the disk image is
automatically determined (i.e. the minimal sizes of all partitions
are summed up, taking space for additional metadata into account).
This switch is not supported if the specified node is a block
device. This switch has no effect if the file is already as large
as the specified size or larger. The specified size is implicitly
rounded up to multiples of 4096. When used with --empty=create this
specifies the initial size of the loopback file to create.

The --size=auto option takes the sizes of pre-existing partitions
into account. However, it does not accommodate for partition tables
that are not tightly packed: the configured partitions might still
not fit into the backing device if empty space exists between
pre-existing partitions (or before the first partition) that cannot
be fully filled by partitions to grow or create.

Also note that the automatic size determination does not take files
or directories specified with CopyFiles= into account: operation
might fail if the specified files or directories require more disk
space then the configured per-partition minimal size limit.

--factory-reset=

Takes boolean. If this switch is not specified --factory=reset=no is the implied default. Controls whether to operate in "factory reset" mode, see above. If set to true this will remove all existing partitions marked with FactoryReset= set to yes early while executing the re-partitioning algorithm. Use with care, this is a great way to lose all your data. Note that partition files need to explicitly turn FactoryReset= on, as the option defaults to off. If no partitions are marked for factory reset this switch has no effect. Note that there are two other methods to request factory reset operation: via the kernel command line and via an EFI variable, see above.

--can-factory-reset

If this switch is specified the disk is not re-partitioned. Instead it is determined if any existing partitions are marked with FactoryReset=. If there are the tool will exit with exit status zero, otherwise non-zero. This switch may be used to quickly determine whether the running system supports a factory reset mechanism built on systemd-repart.

--root=

Takes a path to a directory to use as root file system when searching for repart.d/*.conf files, for the machine ID file to use as seed and for the CopyFiles= and CopyBlocks= source files and directories. By default when invoked on the regular system this defaults to the host's root file system /. If invoked from the initrd this defaults to /sysroot/, so that the tool operates on the configuration and machine ID stored in the root file system later transitioned into itself.

--image=

Takes a path to a disk image file or device to mount and use in a similar fashion to --root=, see above.

--seed=

Takes a UUID as argument or the special value random. If a UUID is

specified the UUIDs to assign to partitions and the partition table
itself are derived via cryptographic hashing from it. If not
specified it is attempted to read the machine ID from the host (or
more precisely, the root directory configured via --root=) and use
it as seed instead, falling back to a randomized seed otherwise.
Use --seed=random to force a randomized seed. Explicitly specifying
the seed may be used to generated strictly reproducible partition
tables.

--pretty=

Takes a boolean argument. If this switch is not specified, it
defaults to on when called from an interactive terminal and off
otherwise. Controls whether to show a user friendly table and
graphic illustrating the changes applied.

--definitions=

Takes a file system path. If specified the *.conf files are read
from the specified directory instead of searching in
/usr/lib/repart.d/*.conf, /etc/repart.d/*.conf,
/run/repart.d/*.conf.

This parameter can be specified multiple times.

--key-file=

Takes a file system path. Configures the encryption key to use when
setting up LUKS2 volumes configured with the Encrypt=key-file
setting in partition files. Should refer to a regular file
containing the key, or an AF_UNIX stream socket in the file system.
In the latter case a connection is made to it and the key read from
it. If this switch is not specified the empty key (i.e. zero length
key) is used. This behaviour is useful for setting up encrypted
partitions during early first boot that receive their user-supplied
password only in a later setup step.

--private-key=

Takes a file system path. Configures the signing key to use when
creating verity signature partitions with the Verity=signature
setting in partition files.

--certificate=

Takes a file system path. Configures the PEM encoded X.509 certificate to use when creating verity signature partitions with the Verity=signature setting in partition files.

--tpm2-device=, --tpm2-pcrs=

Configures the TPM2 device and list of PCRs to use for LUKS2 volumes configured with the Encrypt=tpm2 option. These options take the same parameters as the identically named options to systemd-cryptenroll(1) and have the same effect on partitions where TPM2 enrollment is requested.

--tpm2-public-key= [PATH], --tpm2-public-key-pcrs= [PCR...]

Configures a TPM2 signed PCR policy to bind encryption to. See systemd-cryptenroll(1) for details on these two options.

--split= [BOOL]

Enables generation of split artifacts from partitions configured with SplitName=. If enabled, for each partition with SplitName= set, a separate output file containing just the contents of that partition is generated. The output filename consists of the loopback filename suffixed with the name configured with SplitName=. If the loopback filename ends with ".raw", the suffix is inserted before the ".raw" extension instead.

Note that --split is independent from --dry-run. Even if --dry-run is enabled, split artifacts will still be generated from an existing image if --split is enabled.

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

--no-pager

Do not pipe output into a pager.

--no-legend

Do not print the legend, i.e. column headers and the footer with hints.

--json=MODE

Shows output formatted as JSON. Expects one of "short" (for the shortest possible output without any redundant whitespace or line breaks), "pretty" (for a pretty version of the same, with indentation and line breaks) or "off" (to turn off JSON output, the default).

EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

SEE ALSO

systemd(1), repart.d(5), machine-id(5), systemd-cryptenroll(1)