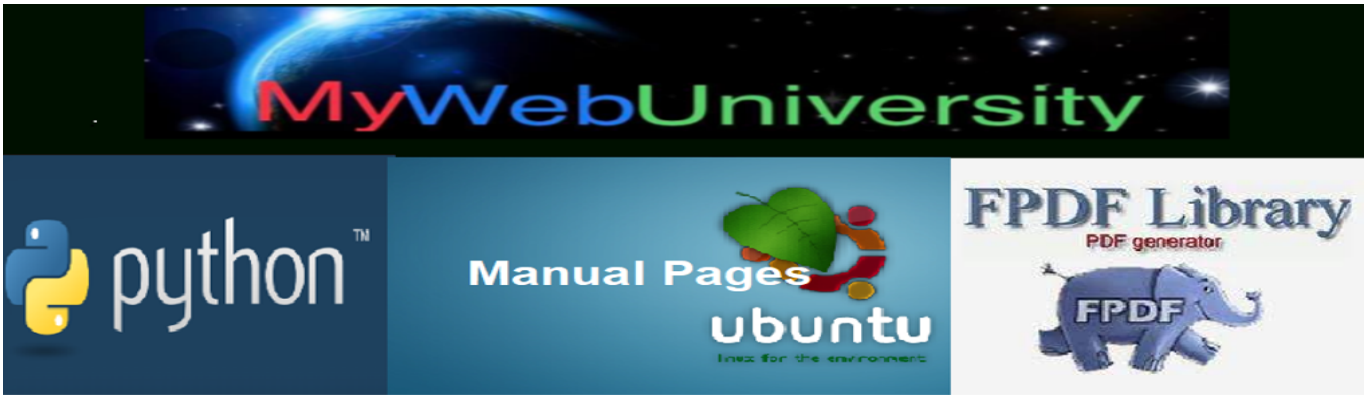

Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'crun.1'

$ man crun.1

crun(1) General Commands Manual crun(1)

NAME

crun - a fast and lightweight OCI runtime

SYNOPSIS

crun [global options] command [command options] [arguments...]

DESCRIPTION

crun is a command line program for running Linux containers that follow the Open Container Initiative (OCI) format.

COMMANDS

create Create a container. The runtime detaches from the container process once the con? tainer environment is created. It is necessary to successively use start for starting the container.

delete Remove definition for a container.

exec Exec a command in a running container.

list List known containers.

kill Send the specified signal to the container init process. If no signal is specified, SIGTERM is used.

ps Show the processes running in a container.

run Create and immediately start a container.

spec Generate a configuration file.

start Start a container that was previously created. A container cannot be started multi? ple times.

state Output the state of a container.

pause Pause all the processes in the container.

resume Resume the processes in the container.

update Update container resource constraints.

checkpoint Checkpoint a running container using CRIU

restore Restore a container from a checkpoint

STATE

By default, when running as root user, crun saves its state under the /run/crun directory.

As unprivileged user, instead the XDG_RUNTIME_DIR environment variable is honored, and the directory $XDG_RUNTIME_DIR/crun is used. The global option --root overrides this setting.

GLOBAL OPTIONS

--debug Produce verbose output.

--log=LOG-DESTINATION Define the destination for the error and warning messages generated by crun. If the error happens late in the container init process, when crun already stopped watching it, then it will be printed to the container stderr.

It is specified in the form BACKEND:SPECIFIER.

These following backends are supported:

? file:PATH

? journald:IDENTIFIER

? syslog:IDENTIFIER

If no backend is specified, then file: is used by default.

--log-format=FORMAT Define the format of the log messages. It can either be text, or json. The default is text.

--no-pivot Use chroot(2) instead of pivot_root(2) when creating the container. This op? tion is not safe, and should be avoided.

--root=DIR Defines where to store the state for crun containers.

--systemd-cgroup Use systemd for configuring cgroups. If not specified, the cgroup is created directly using the cgroupfs backend.

--cgroup-manager=MANAGER Specify what cgroup manager must be used. Permitted values are cgroupfs, systemd and disabled.

-?, --help Print a help list.

--usage Print a short usage message.

-V, --version Print program version

CREATE OPTIONS

crun [global options] create [options] CONTAINER

--bundle=BUNDLE Path to the OCI bundle, by default it is the current directory.

--config=FILE Override the configuration file to use. The default value is config.json.

--console-socket=SOCKET Path to a UNIX socket that will receive the ptmx end of the tty for the container.

--no-new-keyring Keep the same session key

--preserve-fds=N Additional number of FDs to pass into the container.

--pid-file=PATH Path to the file that will contain the container process PID.

RUN OPTIONS

crun [global options] run [options] CONTAINER

--bundle=BUNDLE Path to the OCI bundle, by default it is the current directory.

--config=FILE Override the configuration file to use. The default value is config.json.

--console-socket=SOCKET Path to a UNIX socket that will receive the ptmx end of the tty for the container.

--no-new-keyring Keep the same session key.

--preserve-fds=N Additional number of FDs to pass into the container.

--pid-file=PATH Path to the file that will contain the container process PID.

--detach Detach the container process from the current session.

DELETE OPTIONS

crun [global options] delete [options] CONTAINER

--force Delete the container even if it is still running.

--regex=REGEX Delete all the containers that satisfy the specified regex.

EXEC OPTIONS

crun [global options] exec [options] CONTAINER CMD

--console-socket=SOCKET Path to a UNIX socket that will receive the ptmx end of the tty for the container.

--cwd=PATH Set the working directory for the process to PATH.

--cap=CAP Specify an additional capability to add to the process.

--detach Detach the container process from the current session.

--env=ENV Specify an environment variable.

--preserve-fds=N Additional number of FDs to pass into the container.

--process=FILE Path to a file containing the process JSON configuration.

--pid-file=PATH Path to the file that will contain the new process PID.

-t --tty Allocate a pseudo TTY.

-u USERSPEC --user=USERSPEC Specify the user in the form UID[:GID].

LIST OPTIONS

crun [global options] list [options]

-q --quiet Show only the container ID.

KILL OPTIONS

crun [global options] kill [options] CONTAINER SIGNAL

--all Kill all the processes in the container.

--regex=REGEX Kill all the containers that satisfy the specified regex.

PS OPTIONS

crun [global options] ps [options]

--format=FORMAT Specify the output format. It must be either table or json. By default table is used.

SPEC OPTIONS

crun [global options] spec [options]

--rootless Generate a config.json file that is usable by an unprivileged user.

UPDATE OPTIONS

crun [global options] update [options] CONTAINER

--blkio-weight=VALUE Specifies per cgroup weight.

--cpu-period=VALUE CPU CFS period to be used for hardcapping.

--cpu-quota=VALUE** CPU CFS hardcap limit.

--cpu-rt-period=VALUE CPU realtime period to be used for hardcapping.

--cpu-rt-runtime=VALUE CPU realtime hardcap limit.

--cpu-share=VALUE CPU shares.

--cpuset-cpus=VALUE CPU(s) to use.

--cpuset-mems=VALUE Memory node(s) to use.

--kernel-memory=VALUE Kernel memory limit.

--kernel-memory-tcp=VALUE Kernel memory limit for TCP buffer.

--memory=VALUE Memory limit.

--memory-reservation=VALUE Memory reservation or soft_limit.

--memory-swap=VALUE Total memory usage.

--pids-limit=VALUE Maximum number of pids allowed in the container.

-r, --resources=FILE Path to the file containing the resources to update.

CHECKPOINT OPTIONS

crun [global options] checkpoint [options] CONTAINER

--image-path=DIR Path for saving CRIU image files

--work-path=DIR Path for saving work files and logs

--leave-running Leave the process running after checkpointing

--tcp-established Allow open TCP connections

--ext-unix-sk Allow external UNIX sockets

--shell-job Allow shell jobs

RESTORE OPTIONS

crun [global options] restore [options] CONTAINER

-b DIR --bundle=DIR Container bundle directory (default ".")

--image-path=DIR Path for saving CRIU image files

--work-path=DIR Path for saving work files and logs

--tcp-established Allow open TCP connections

--ext-unix Allow external UNIX sockets

--shell-job Allow shell jobs

--detach Detach from the container's process

--pid-file=FILE Where to write the PID of the container

Extensions to OCI

run.oci.seccomp.receiver=PATH

If the annotation run.oci.seccomp.receiver=PATH is specified, the seccomp listener is sent to the UNIX socket listening on the specified path. It can also set with the RUN_OCI_SEC? COMP_RECEIVER environment variable. It is an experimental feature, and the annotation will be removed once it is supported in the OCI runtime specs. It must be an absolute path.

run.oci.seccomp.plugins=PATH

If the annotation run.oci.seccomp.plugins=PLUGIN1[:PLUGIN2]... is specified, the seccomp listener fd is handled through the specified plugins. The plugin must either be an abso? lute path or a file name that is looked up by ldopen(3). More information on how the lookup is performed are available on the ld.so(8) man page.

run.oci.seccomp_fail_unknown_syscall=1

If the annotation run.oci.seccomp_fail_unknown_syscall is present, then crun will fail when an unknown syscall is encountered in the seccomp configuration.

run.oci.keep_original_groups=1

If the annotation run.oci.keep_original_groups is present, then crun will skip the set? groups syscall that is used to either set the additional groups specified in the OCI con? figuration, or to reset the list of additional groups if none is specified.

run.oci.systemd.force_cgroup_v1=/PATH

If the annotation run.oci.systemd.force_cgroup_v1=/PATH is present, then crun will over? ride the specified mount point /PATH with a cgroup v1 mount made of a single hierarchy none,name=systemd. It is useful to run on a cgroup v2 system containers using older ver? sions of systemd that lack support for cgroup v2.

run.oci.timens_offset=ID SEC NSEC

Specify the offset to be written to /proc/self/timens_offsets when creating a time name? space.

run.oci.systemd.subgroup=SUBGROUP

Override the name for the systemd sub cgroup created under the systemd scope, so the final cgroup will be like:

/sys/fs/cgroup/$PATH/$SUBGROUP

When it is set to the empty string, a sub cgroup is not created.

If not specified, it defaults to container on cgroup v2, and to "" on cgroup v1.

e.g.

/sys/fs/cgroup//system.slice/foo-352700.scope/container

run.oci.hooks.stdout=FILE

If the annotation run.oci.hooks.stdout is present, then crun will open the specified file and use it as the stdout for the hook processes. The file is opened in append mode and it is created if it doesn't already exist.

run.oci.hooks.stderr=FILE

If the annotation run.oci.hooks.stderr is present, then crun will open the specified file and use it as the stderr for the hook processes. The file is opened in append mode and it is created if it doesn't already exist.

run.oci.handler=HANDLER

It is an experimental feature.

If specified, run the specified handler for execing the container. The only supported value is krun. When krun is specified, the libkrun.so shared object is loaded and it is used to launch the container using libkrun.

tmpcopyup mount options

If the tmpcopyup option is specified for a tmpfs, then the path that is shadowed by the tmpfs mount is recursively copied up to the tmpfs itself.

Automatically create user namespace

When running as user different than root, an user namespace is automatically created even if it is not specified in the config file. The current user is mapped to the ID 0 in the container, and any additional id specified in the files /etc/subuid and /etc/subgid is au? tomatically added starting with ID 1.

Intermediate user namespace

If the configuration specifies a new user namespace made of a single mapping to the root user, but either the UID or the GID are set as nonzero then crun automatically creates an? other user namespace to map the root user to the specified UID and GID.

It enables running unprivileged containers with UID and GID different than zero, even when a single UID and GID are available, e.g. rootless users on a system without newuidmap/newgidmap.

CGROUP v2

If the cgroup configuration found is for cgroup v1, crun attempts a conversion when run? ning on a cgroup v2 system.

These are the OCI resources currently supported with cgroup v2 and how they are converted when needed from the cgroup v1 configuration.

Memory controller

??

?OCI (x) ? cgroup 2 value (y) ? conversion ? comment ?

??

?limit ? memory.max ? y = x ? swap ?

??

?reservation ? memory.low ? y = x ? ?

??

PIDs controller

??

?OCI (x) ? cgroup 2 value (y) ? conversion ? comment ?

??

?limit ? pids.max ? y = x ? ?

??

CPU controller

??

?OCI (x) ? cgroup 2 value (y) ? conversion ? comment ?

??

?shares ? cpu.weight ? y = (1 + ((x - 2) * ? ?

? ? ? 9999) / 262142) ? ?

??

? ? convert from ? ? ?

? ? [2-262144] to ? ? ?

? ? [1-10000] ? ? ?

??

?period ? cpu.max ? y = x ? period and quota ?

? ? ? ? are written to? ?

? ? ? ? gether ?

??

?quota ? cpu.max ? y = x ? period and quota ?

? ? ? ? are written to? ?

? ? ? ? gether ?

??

? ? ? ? ?

??

blkio controller

??

?OCI (x) ? cgroup 2 value (y) ? conversion ? comment ?

??

?weight ? io.bfq.weight ? y = (1 + (x - 10) * ? ?

? ? ? 9999 / 990) ? ?

??

? ? convert linearly ? ? ?

? ? from [10-1000] to ? ? ?

? ? [1-10000] ? ? ?

??

?weight_device ? io.bfq.weight ? y = (1 + (x - 10) * ? ?

? ? ? 9999 / 990) ? ?

??

? ? convert linearly ? ? ?

? ? from [10-1000] to ? ? ?

? ? [1-10000] ? ? ? ?

??

?rbps ? io.max ? y=x ? wbps ?

??

cpuset controller

??

?OCI (x) ? cgroup 2 value (y) ? conversion ? comment ?

??

?cpus ? cpuset.cpus ? y = x ? mems ?

??

hugetlb controller

??

?OCI (x) ? cgroup 2 value (y) ? conversion ? comment ?

??

?.limit_in_bytes ? hugetlb..max ? y = x ? ?

??