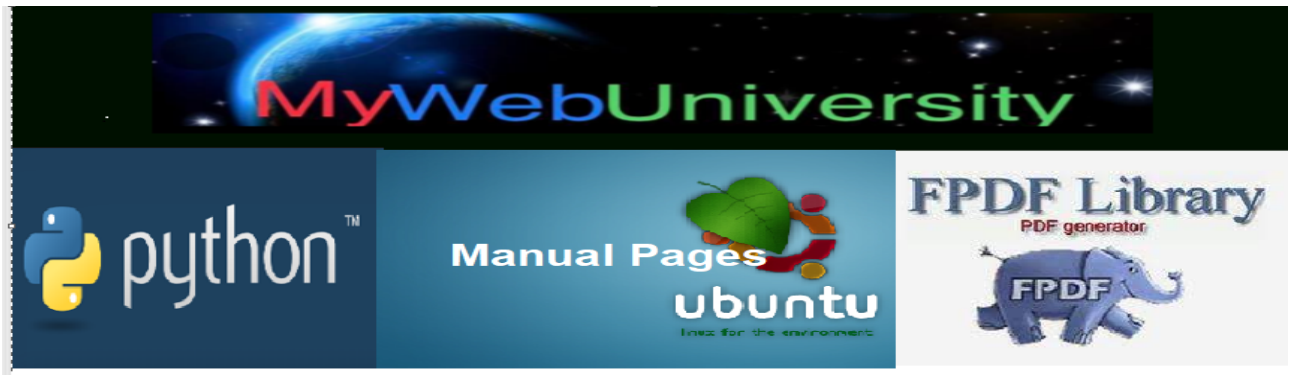

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'docker-container-run.1'

$ man docker-container-run.1

DOCKER(1) Docker User Manuals DOCKER(1)

NAME

docker-container-run - Create and run a new container from an image

SYNOPSIS

docker container run [OPTIONS] IMAGE [COMMAND] [ARG...]

DESCRIPTION

Alias for docker run.

OPTIONS

--add-host= Add a custom host-to-IP mapping (host:ip)

--annotation=map[] Add an annotation to the container (passed through to the OCI run? time)

-a, --attach= Attach to STDIN, STDOUT or STDERR

--blkio-weight=0 Block IO (relative weight), between 10 and 1000, or 0 to disable (de? fault 0)

--blkio-weight-device=[] Block IO weight (relative device weight)

--cap-add= Add Linux capabilities

--cap-drop= Drop Linux capabilities

--cgroup-parent="" Optional parent cgroup for the container

--cgroupns="" Cgroup namespace to use (host|private) 'host': Run the container in the Docker host's cgroup namespace 'private': Run the container in its own private cgroup namespace '': Use the cgroup namespace as configured by the default-cgroupns-mode option on the daemon (default)

--cidfile="" Write the container ID to the file

--cpu-count=0 CPU count (Windows only)

--cpu-percent=0 CPU percent (Windows only)

--cpu-period=0 Limit CPU CFS (Completely Fair Scheduler) period

--cpu-quota=0 Limit CPU CFS (Completely Fair Scheduler) quota

--cpu-rt-period=0 Limit CPU real-time period in microseconds

--cpu-rt-runtime=0 Limit CPU real-time runtime in microseconds

-c, --cpu-shares=0 CPU shares (relative weight)

--cpus= Number of CPUs

--cpuset-cpus="" CPUs in which to allow execution (0-3, 0,1)

--cpuset-mems="" MEMs in which to allow execution (0-3, 0,1)

-d, --detach[=false] Run container in background and print container ID

--detach-keys="" Override the key sequence for detaching a container

--device= Add a host device to the container

--device-cgroup-rule= Add a rule to the cgroup allowed devices list

--device-read-bps=[] Limit read rate (bytes per second) from a device

--device-read-iops=[] Limit read rate (IO per second) from a device

--device-write-bps=[] Limit write rate (bytes per second) to a device

--device-write-iops=[] Limit write rate (IO per second) to a device

--dns= Set custom DNS servers

--dns-option= Set DNS options

--dns-search= Set custom DNS search domains

--domainname="" Container NIS domain name

--entrypoint="" Overwrite the default ENTRYPOINT of the image

-e, --env= Set environment variables

--env-file= Read in a file of environment variables

--expose= Expose a port or a range of ports

--gpus= GPU devices to add to the container ('all' to pass all GPUs)

--group-add= Add additional groups to join

--health-cmd="" Command to run to check health

--health-interval=0s Time between running the check (ms|s|m|h) (default 0s)

--health-retries=0 Consecutive failures needed to report unhealthy

--health-start-interval=0s Time between running the check during the start period (ms|s|m|h) (default 0s)

--health-start-period=0s Start period for the container to initialize before starting health-retries countdown (ms|s|m|h) (default 0s)

--health-timeout=0s Maximum time to allow one check to run (ms|s|m|h) (default 0s)

--help[=false] Print usage

-h, --hostname="" Container host name

--init[=false] Run an init inside the container that forwards signals and reaps processes

-i, --interactive[=false] Keep STDIN open even if not attached

--io-maxbandwidth=0 Maximum IO bandwidth limit for the system drive (Windows only)

--io-maxiops=0 Maximum IOps limit for the system drive (Windows only)

--ip= IPv4 address (e.g., 172.30.100.104)

--ip6= IPv6 address (e.g., 2001:db8::33)

--ipc="" IPC mode to use

--isolation="" Container isolation technology

-l, --label= Set meta data on a container

--label-file= Read in a line delimited file of labels

--link= Add link to another container

--link-local-ip= Container IPv4/IPv6 link-local addresses

--log-driver="" Logging driver for the container

--log-opt= Log driver options

--mac-address="" Container MAC address (e.g., 92:d0:c6:0a:29:33)

-m, --memory=0 Memory limit

--memory-reservation=0 Memory soft limit

--memory-swap=0 Swap limit equal to memory plus swap: '-1' to enable unlimited swap

--memory-swappiness=-1 Tune container memory swappiness (0 to 100)

--mount= Attach a filesystem mount to the container

--name="" Assign a name to the container

--network= Connect a container to a network

--network-alias= Add network-scoped alias for the container

--no-healthcheck[=false] Disable any container-specified HEALTHCHECK

--oom-kill-disable[=false] Disable OOM Killer

--oom-score-adj=0 Tune host's OOM preferences (-1000 to 1000)

--pid="" PID namespace to use

--pids-limit=0 Tune container pids limit (set -1 for unlimited)

--platform="" Set platform if server is multi-platform capable

--privileged[=false] Give extended privileges to this container

-p, --publish= Publish a container's port(s) to the host

-P, --publish-all[=false] Publish all exposed ports to random ports

--pull="missing" Pull image before running ("always", "missing", "never")

-q, --quiet[=false] Suppress the pull output

--read-only[=false] Mount the container's root filesystem as read only

--restart="no" Restart policy to apply when a container exits

--rm[=false] Automatically remove the container and its associated anonymous volumes when it exits

--runtime="" Runtime to use for this container

--security-opt= Security Options

--shm-size=0 Size of /dev/shm

--sig-proxy[=true] Proxy received signals to the process

--stop-signal="" Signal to stop the container

--stop-timeout=0 Timeout (in seconds) to stop a container

--storage-opt= Storage driver options for the container

--sysctl=map[] Sysctl options

--tmpfs= Mount a tmpfs directory

-t, --tty[=false] Allocate a pseudo-TTY

--ulimit=[] Ulimit options

--use-api-socket[=false] Bind mount Docker API socket and required auth

-u, --user="" Username or UID (format: [:])

--userns="" User namespace to use

--uts="" UTS namespace to use

-v, --volume= Bind mount a volume

--volume-driver="" Optional volume driver for the container

--volumes-from= Mount volumes from the specified container(s)

-w, --workdir="" Working directory inside the container

SEE ALSO

docker-container(1)