



## ***Rocky Enterprise Linux 9.2 Manual Pages on command 'ctr.8'***

**C:\>man ctr.8**

ctr(8)() ctr(8)()

ctr is an unsupported debug and administrative client for interacting with the containerd daemon. Because it is unsupported, the commands, options, and operations are not guaranteed to be backward compatible or stable from release to release of the containerd project.

### NAME

ctr

### SYNOPSIS

ctr

`[--address|-a]=[value]`

`[--connect-timeout]=[value]`

`[--debug]`

`[--namespace|-n]=[value]`

`[--timeout]=[value]`

Usage:

`ctr [GLOBAL OPTIONS] command [COMMAND OPTIONS] [ARGUMENTS...]`

### GLOBAL OPTIONS

`--address, -a=""`: address for containerd's GRPC server (default: `/run/containerd/containerd.sock`)

`--connect-timeout=""`: timeout for connecting to containerd (default: 0s)

`--debug`: enable debug output in logs

`--namespace, -n=""`: namespace to use with commands (default: default)

--timeout="": total timeout for ctr commands (default: 0s)

## COMMANDS

### plugins, plugin

provides information about containerd plugins

#### list, ls

lists containerd plugins

--detailed, -d: print detailed information about each plugin

--quiet, -q: print only the plugin ids

### version

print the client and server versions

### containers, c, container

manage containers

#### create

create container

--allow-new-privs: turn off OCI spec's NoNewPrivileges feature flag

--apparmor-default-profile="": enable AppArmor with the default profile with the specified name, e.g. "cri-containerd.apparmor.d"

--apparmor-profile="": enable AppArmor with an existing custom profile

--config, -c="": path to the runtime-specific spec config file

--cpu-period="": Limit CPU CFS period (default: 0)

--cpu-quota="": Limit CPU CFS quota (default: -1)

--cwd="": specify the working directory of the process

--device="": file path to a device to add to the container; or a path to a directory tree of devices to add to the container

--env="": specify additional container environment variables (e.g. FOO=bar)

--env-file="": specify additional container environment variables in a file (e.g. FOO=bar, one per line)

--gpus="": add gpus to the container (default: 0)

--label="": specify additional labels (e.g. foo=bar)

--memory-limit="": memory limit (in bytes) for the container (default: 0)

--mount="": specify additional container mount (e.g. type=bind,src=/tmp,dst=/host,options=rbind:ro)

--net-host: enable host networking for the container

--no-pivot: disable use of pivot-root (linux only)  
--pid-file="": file path to write the task's pid  
--privileged: run privileged container  
--read-only: set the containers filesystem as readonly  
--rootfs: use custom rootfs that is not managed by containerd snapshotter  
--runtime="": runtime name (default: io.containerd.runc.v2)  
--runtime-config-path="": optional runtime config path  
--seccomp: enable the default seccomp profile  
--seccomp-profile="": file path to custom seccomp profile. seccomp must be set to true, before using seccomp-profile  
--snapshotter="": snapshotter name. Empty value stands for the default value.  
--tty, -t: allocate a TTY for the container  
--with-ns="": specify existing Linux namespaces to join at container runtime (format: ':')

#### delete, del, rm

delete one or more existing containers  
--keep-snapshot: do not clean up snapshot with container

#### info

get info about a container  
--spec: only display the spec

#### list, ls

list containers  
--quiet, -q: print only the container id

#### label

set and clear labels for a container

#### checkpoint

checkpoint a container  
--image: include the image in the checkpoint  
--rw: include the rw layer in the checkpoint  
--task: checkpoint container task

#### restore

restore a container from checkpoint  
--live: restore the runtime and memory data from the checkpoint

--rw: restore the rw layer from the checkpoint

## content

manage content

## active

display active transfers

--root="": path to content store root (default: /tmp/content)

--timeout, -t="": total timeout for fetch (default: 0s)

## delete, del, remove, rm

permanently delete one or more blobs

## edit

edit a blob and return a new digest

--editor="": select editor (vim, emacs, etc.)

--validate="": validate the result against a format (json, mediatype, etc.)

## fetch

fetch all content for an image into containerd

--all-metadata: Pull metadata for all platforms

--all-platforms: pull content from all platforms

--hosts-dir="": Custom hosts configuration directory

--http-dump: dump all HTTP request/responses when interacting with container reg?

## istry

--http-trace: enable HTTP tracing for registry interactions

--label="": labels to attach to the image

--metadata-only: Pull all metadata including manifests and configs

--plain-http: allow connections using plain HTTP

--platform="": Pull content from a specific platform

--refresh="": refresh token for authorization server

--skip-verify, -k: skip SSL certificate validation

--tlscacert="": path to TLS root CA

--tlscert="": path to TLS client certificate

--tlskey="": path to TLS client key

--user, -u="": user[:password] Registry user and password

## fetch-object

retrieve objects from a remote

--hosts-dir="": Custom hosts configuration directory

--http-dump: dump all HTTP request/responses when interacting with container registry

--http-trace: enable HTTP tracing for registry interactions

--plain-http: allow connections using plain HTTP

--refresh="": refresh token for authorization server

--skip-verify, -k: skip SSL certificate validation

--tlscacert="": path to TLS root CA

--tlscert="": path to TLS client certificate

--tlskey="": path to TLS client key

--user, -u="": user[:password] Registry user and password

## get

get the data for an object

## ingest

accept content into the store

--expected-digest="": verify content against expected digest

--expected-size="": validate against provided size (default: 0)

## list, ls

list all blobs in the store

--quiet, -q: print only the blob digest

## push-object

push an object to a remote

--hosts-dir="": Custom hosts configuration directory

--http-dump: dump all HTTP request/responses when interacting with container registry

registry

--http-trace: enable HTTP tracing for registry interactions

--plain-http: allow connections using plain HTTP

--refresh="": refresh token for authorization server

--skip-verify, -k: skip SSL certificate validation

--tlscacert="": path to TLS root CA

--tlscert="": path to TLS client certificate

--tlskey="": path to TLS client key

--user, -u="": user[:password] Registry user and password

## label

add labels to content

## prune

prunes content from the content store

## references

prunes preference labels from the content store (layers only by default)

--async: allow garbage collection to cleanup asynchronously

--dry: just show updates without applying (enables debug logging)

## events, event

display containerd events

## images, image, i

manage images

## check

check existing images to ensure all content is available locally

--quiet, -q: print only the ready image refs (fully downloaded and unpacked)

--snapshotter="": snapshotter name. Empty value stands for the default value.

## export

export images

--all-platforms: exports content from all platforms

--platform="": Pull content from a specific platform (default: [])

--skip-manifest-json: do not add Docker compatible manifest.json to archive

--skip-non-distributable: do not add non-distributable blobs such as Windows layers  
to archive

## import

import images

--all-platforms: imports content for all platforms, false by default

--base-name="": base image name for added images, when provided only images with  
this name prefix are imported

--compress-blobs: compress uncompressed blobs when creating manifest (Docker format  
only)

--digests: whether to create digest images (default: false)

--index-name="": image name to keep index as, by default index is discarded

--no-unpack: skip unpacking the images, false by default

--snapshotter="": snapshotter name. Empty value stands for the default value.

## list, ls

list images known to containerd

--quiet, -q: print only the image refs

## mount

mount an image to a target path

--hosts-dir="": Custom hosts configuration directory

--http-dump: dump all HTTP request/responses when interacting with container reg?

## istry

--http-trace: enable HTTP tracing for registry interactions

--label="": labels to attach to the image

--plain-http: allow connections using plain HTTP

--platform="": Mount the image for the specified platform (default: linux/amd64)

--refresh="": refresh token for authorization server

--rw: Enable write support on the mount

--skip-verify, -k: skip SSL certificate validation

--snapshotter="": snapshotter name. Empty value stands for the default value.

--tlscacert="": path to TLS root CA

--tlscert="": path to TLS client certificate

--tlskey="": path to TLS client key

--user, -u="": user[:password] Registry user and password

## unmount

unmount the image from the target

--hosts-dir="": Custom hosts configuration directory

--http-dump: dump all HTTP request/responses when interacting with container reg?

## istry

--http-trace: enable HTTP tracing for registry interactions

--label="": labels to attach to the image

--plain-http: allow connections using plain HTTP

--refresh="": refresh token for authorization server

--rm: remove the snapshot after a successful unmount

--skip-verify, -k: skip SSL certificate validation

--snapshotter="": snapshotter name. Empty value stands for the default value.

--tlscacert="": path to TLS root CA  
--tlscert="": path to TLS client certificate  
--tlskey="": path to TLS client key  
--user, -u="": user[:password] Registry user and password

## pull

pull an image from a remote

--all-metadata: Pull metadata for all platforms  
--all-platforms: pull content and metadata from all platforms  
--hosts-dir="": Custom hosts configuration directory  
--http-dump: dump all HTTP request/responses when interacting with container reg?

## istry

--http-trace: enable HTTP tracing for registry interactions  
--label="": labels to attach to the image  
--max-concurrent-downloads="": Set the max concurrent downloads for each pull (default: 0)  
--plain-http: allow connections using plain HTTP  
--platform="": Pull content from a specific platform (default: [])  
--print-chainid: Print the resulting image's chain ID  
--refresh="": refresh token for authorization server  
--skip-verify, -k: skip SSL certificate validation  
--snapshotter="": snapshotter name. Empty value stands for the default value.  
--tlscacert="": path to TLS root CA  
--tlscert="": path to TLS client certificate  
--tlskey="": path to TLS client key  
--user, -u="": user[:password] Registry user and password

## push

push an image to a remote

--hosts-dir="": Custom hosts configuration directory  
--http-dump: dump all HTTP request/responses when interacting with container reg?

## istry

--http-trace: enable HTTP tracing for registry interactions  
--manifest="": digest of manifest  
--manifest-type="": media type of manifest digest (default: application/vnd.oci.im?)

age.manifest.v1+json)

--max-concurrent-uploaded-layers="": Set the max concurrent uploaded layers for each push (default: 0)

--plain-http: allow connections using plain HTTP

--platform="": push content from a specific platform (default: [])

--refresh="": refresh token for authorization server

--skip-verify, -k: skip SSL certificate validation

--tlscacert="": path to TLS root CA

--tlscert="": path to TLS client certificate

--tlskey="": path to TLS client key

--user, -u="": user[:password] Registry user and password

remove, rm

remove one or more images by reference

--sync: Synchronously remove image and all associated resources

tag

tag an image

--force: force target\_ref to be created, regardless if it already exists

label

set and clear labels for an image

--replace-all, -r: replace all labels

convert

convert an image

--all-platforms: exports content from all platforms

--oci: convert Docker media types to OCI media types

--platform="": Pull content from a specific platform (default: [])

--uncompress: convert tar.gz layers to uncompressed tar layers

leases

manage leases

list, ls

list all active leases

--quiet, -q: print only the blob digest

create

create lease

--expires, -x="": expiration of lease (0 value will not expire) (default: 24h0m0s)

--id="": set the id for the lease, will be generated by default

delete, rm

delete a lease

--sync: Synchronously remove leases and all unreferenced resources

namespaces, namespace, ns

manage namespaces

create, c

create a new namespace

list, ls

list namespaces

--quiet, -q: print only the namespace name

remove, rm

remove one or more namespaces

--cgroup, -c: delete the namespace's cgroup

label

set and clear labels for a namespace

pprof

provide golang pprof outputs for containerd

--debug-socket, -d="": socket path for containerd's debug server (default:

/run/containerd/debug.sock)

block

goroutine blocking profile

goroutines

dump goroutine stack dump

heap

dump heap profile

profile

CPU profile

--seconds, -s="": duration for collection (seconds) (default: 30s)

threadcreate

goroutine thread creating profile

trace

collect execution trace

--seconds, -s="": trace time (seconds) (default: 5s)

run

run a container

--allow-new-privs: turn off OCI spec's NoNewPrivileges feature flag

--apparmor-default-profile="": enable AppArmor with the default profile with the specified name, e.g. "cri-containerd.apparmor.d"

--apparmor-profile="": enable AppArmor with an existing custom profile

--cgroup="": cgroup path (To disable use of cgroup, set to "" explicitly)

--cni: enable cni networking for the container

--config, -c="": path to the runtime-specific spec config file

--cpu-period="": Limit CPU CFS period (default: 0)

--cpu-quota="": Limit CPU CFS quota (default: -1)

--cpus="": set the CFS cpu quota (default: 0.000000)

--cwd="": specify the working directory of the process

--detach, -d: detach from the task after it has started execution

--device="": file path to a device to add to the container; or a path to a directory tree of devices to add to the container

--env="": specify additional container environment variables (e.g. FOO=bar)

--env-file="": specify additional container environment variables in a file (e.g. FOO=bar, one per line)

--fifo-dir="": directory used for storing IO FIFOs

--gidmap="": run inside a user namespace with the specified GID mapping range; specified with the format container-gid:host-gid:length

--gpus="": add gpus to the container (default: 0)

--label="": specify additional labels (e.g. foo=bar)

--log-uri="": log uri

--memory-limit="": memory limit (in bytes) for the container (default: 0)

--mount="": specify additional container mount (e.g. type=bind,src=/tmp,dst=/host,options=rbind:ro)

--net-host: enable host networking for the container

--no-pivot: disable use of pivot-root (linux only)

--null-io: send all IO to /dev/null

--pid-file="": file path to write the task's pid  
--platform="": run image for specific platform  
--privileged: run privileged container  
--read-only: set the containers filesystem as readonly  
--remap-labels: provide the user namespace ID remapping to the snapshotter via labels options; requires snapshotter support  
--rm: remove the container after running  
--rootfs: use custom rootfs that is not managed by containerd snapshotter  
--runc-binary="": specify runc-compatible binary  
--runc-root="": specify runc-compatible root  
--runc-systemd-cgroup: start runc with systemd cgroup manager  
--runtime="": runtime name (default: io.containerd.runc.v2)  
--runtime-config-path="": optional runtime config path  
--seccomp: enable the default seccomp profile  
--seccomp-profile="": file path to custom seccomp profile. seccomp must be set to true, before using seccomp-profile  
--snapshotter="": snapshotter name. Empty value stands for the default value.  
--tty, -t: allocate a TTY for the container  
--uidmap="": run inside a user namespace with the specified UID mapping range; specified with the format container-uid:host-uid:length  
--with-ns="": specify existing Linux namespaces to join at container runtime (format: namespace[:mountpoint])

snapshots, snapshot

manage snapshots

--snapshotter="": snapshotter name. Empty value stands for the default value.

commit

commit an active snapshot into the provided name

diff

get the diff of two snapshots. the default second snapshot is the first snapshot's parent.

--keep: keep diff content. up to creator to delete it.

--label="": labels to attach to the image

--media-type="": media type to use for creating diff (default: application/vnd.oci.image.layer.v1.tar)

tion/vnd.oci.image.layer.v1.tar+gzip)

--ref="": content upload reference to use

info

get info about a snapshot

list, ls

list snapshots

mounts, m, mount

mount gets mount commands for the snapshots

prepare

prepare a snapshot from a committed snapshot

--mounts: Print out snapshot mounts as JSON

--target, -t="": mount target path, will print mount, if provided

remove, rm

remove snapshots

label

add labels to content

tree

display tree view of snapshot branches

unpack

unpack applies layers from a manifest to a snapshot

--snapshotter="": snapshotter name. Empty value stands for the default value.

usage

usage snapshots

-b: display size in bytes

view

create a read-only snapshot from a committed snapshot

--mounts: Print out snapshot mounts as JSON

--target, -t="": mount target path, will print mount, if provided

tasks, t, task

manage tasks

attach

attach to the IO of a running container

checkpoint

checkpoint a container

--exit: stop the container after the checkpoint

--image-path="": path to criu image files

--work-path="": path to criu work files and logs

delete, rm

delete one or more tasks

--exec-id="": process ID to kill

--force, -f: force delete task process

exec

execute additional processes in an existing container

--cwd="": working directory of the new process

--detach, -d: detach from the task after it has started execution

--exec-id="": exec specific id for the process

--fifo-dir="": directory used for storing IO FIFOs

--log-uri="": log uri for custom shim logging

--tty, -t: allocate a TTY for the container

--user="": user id or name

list, ls

list tasks

--quiet, -q: print only the task id

kill

signal a container (default: SIGTERM)

--all, -a: send signal to all processes inside the container

--exec-id="": process ID to kill

--signal, -s="": signal to send to the container

pause

pause an existing container

ps

list processes for container

resume

resume a paused container

start

start a container that has been created

--detach, -d: detach from the task after it has started execution

--fifo-dir="": directory used for storing IO FIFOs

--log-uri="": log uri

--null-io: send all IO to /dev/null

--pid-file="": file path to write the task's pid

## metrics, metric

get a single data point of metrics for a task with the built-in Linux runtime

--format="": "table" or "json" (default: table)

## install

install a new package

--libs, -l: install libs from the image

--path="": set an optional install path other than the managed opt directory

--replace, -r: replace any binaries or libs in the opt directory

## oci

OCI tools

## spec

see the output of the default OCI spec

## shim

interact with a shim directly

--id="": container id

## delete

delete a container with a task

## exec

exec a new process in the task's container

--attach, -a: stay attached to the container and open the fifos

--cwd="": current working directory

--env, -e="": add environment vars (default: [])

--spec="": runtime spec

--stderr="": specify the path to the stderr fifo

--stdin="": specify the path to the stdin fifo

--stdout="": specify the path to the stdout fifo

--tty, -t: enable tty support

## start

start a container with a task

state

get the state of all the processes of the task

ctr(8)(