NAME

ctr

SYNOPSIS

ctr

```
[--address|-a]=[value]
[--connect-timeout]=[value]
[--debug]
[--namespace|-n]=[value]
[--timeout]=[value]
```

DESCRIPTION

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.

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

inspect-runtime
Display runtime info

--runc-binary="": Specify runc-compatible binary

--runc-root="": Specify runc-compatible root

--runc-systemd-cgroup: Start runc with systemd cgroup manager

--runtime="": Runtime name or absolute path to runtime binary (default: "io.containerd.runc.v2")

--runtime-config-path="": Optional runtime config path

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

--annotation="": Specify additional OCI annotations (e.g. foo=bar)

--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

--blockio-class="": Name of the blockio class to associate the container with

--blockio-config-file="": File path to blockio class definitions. By default class definitions are not loaded.

--cap-add="": Add Linux capabilities (Set capabilities with 'CAP_' prefix)

--cap-drop="": Drop Linux capabilities (Set capabilities with 'CAP_' prefix)

--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

--hostname="": Set the container's host name

--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

--rdt-class="": Name of the RDT class to associate the container with. Specifies a Class of Service (CLOS) for cache and memory bandwidth management.

--read-only: Set the containers filesystem as readonly

--rootfs: Use custom rootfs that is not managed by containerd snapshotter

--rootfs-propagation="": Set the propagation of the container rootfs

--runc-binary="": Specify runc-compatible binary

--runc-root="": Specify runc-compatible root

--runc-systemd-cgroup: Start runc with systemd cgroup manager

--runtime="": Runtime name or absolute path to runtime binary (default: "io.containerd.runc.v2")

--runtime-config-path="": Optional runtime config path

--sandbox="": Create the container in the given sandbox

--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.

--snapshotter-label="": Labels added to the new snapshot for this container.

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

--user, -u="": Username or user id, group optional (format: [:])

--with-ns="": Specify existing Linux namespaces to join at container runtime (format ':')

delete, del, remove, 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-platforms: Pull content from all platforms

--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

--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-metadata: Skips metadata for unused platforms (Image may be unable to be pushed without metadata)

--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

fetch-blob
Retrieve blobs 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

--media-type="": Specify target mediatype for request header

--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

--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

--local: Run export locally rather than through transfer API

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

--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)

--discard-unpacked-layers: Allow the garbage collector to clean layers up from the content store after unpacking, cannot be used with --no-unpack, false by default

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

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

--local: Run import locally rather than through transfer API

--no-unpack: Skip unpacking the images, cannot be used with --discard-unpacked-layers, false by default

--platform="": Imports content for specific platform

--skip-digest-for-named: Skip applying --digests option to images named in the importing tar (use it in conjunction with --digests)

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

--sync-fs: Synchronize the underlying filesystem containing files when unpack images, false by default

inspect, i
inspect an image

--content: Show JSON content

list, ls
List images known to containerd

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

mount
Mount an image to a target path

--expiration, -x="": Set the expiration time for the mount and snapshots (default: 1h0m0s)

--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

--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.

--sync-fs: Synchronize the underlying filesystem containing files when unpack images, false by default

--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 registry

--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-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 registry

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

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

--local: Fetch content from local client rather than using transfer service

--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

--print-chainid: Print the resulting image's chain ID

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

--skip-metadata: Skips metadata for unused platforms (Image may be unable to be pushed without metadata)

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

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

--sync-fs: Synchronize the underlying filesystem containing files when unpack images, false by default

--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

--allow-non-distributable-blobs: Allow pushing blobs that are marked as non-distributable

--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

--local: Push content from local client rather than using transfer service

--manifest="": Digest of manifest

--manifest-type="": Media type of manifest digest (default: "application/vnd.oci.image.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

--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

prune
Remove unused images

--all: Remove all unused images, not just dangling ones (if all is not specified no images will be pruned)

delete, del, 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

--local: Run tag locally rather than through transfer API

--skip-reference-check: Skip the strict check for reference names

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

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

usage
Display usage of snapshots for a given image ref

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

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, del, remove, 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

--debug="": Debug pprof args (default: 0)

goroutines
Dump goroutine stack dump

--debug="": Debug pprof args (default: 2)

heap
Dump heap profile

--debug="": Debug pprof args (default: 0)

profile
CPU profile

--debug="": Debug pprof args (default: 0)

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

threadcreate
Goroutine thread creating profile

--debug="": Debug pprof args (default: 0)

trace
Collect execution trace

--debug="": Debug pprof args (default: 0)

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

run
Run a container

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

--annotation="": Specify additional OCI annotations (e.g. foo=bar)

--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

--blockio-class="": Name of the blockio class to associate the container with

--blockio-config-file="": File path to blockio class definitions. By default class definitions are not loaded.

--cap-add="": Add Linux capabilities (Set capabilities with 'CAP_' prefix)

--cap-drop="": Drop Linux capabilities (Set capabilities with 'CAP_' prefix)

--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)

--cpu-shares="": Set the cpu shares (default: 1024)

--cpus="": Set the CFS cpu quota (default: 0)

--cpuset-cpus="": Set the CPUs the container will run in (e.g., 1-2,4)

--cpuset-mems="": Set the memory nodes the container will run in (e.g., 1-2,4)

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

--detach, -d: Detach from the task after it has started execution, cannot be used with --rm

--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 ranges; specified with the format **container-gid:host-gid:length**

--gpus="": Add gpus to the container

--hostname="": Set the container's host name

--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

--privileged-without-host-devices: Don't pass all host devices to privileged container

--rdt-class="": Name of the RDT class to associate the container with. Specifies a Class of Service (CLOS) for cache and memory bandwidth management.

--read-only: Set the containers filesystem as readonly

--remap-labels: Provide the user namespace ID remapping to the snapshotter via label options; requires snapshotter support

--rm: Remove the container after running, cannot be used with --detach

--rootfs: Use custom rootfs that is not managed by containerd snapshotter

--rootfs-propagation="": Set the propagation of the container rootfs

--runc-binary="": Specify runc-compatible binary

--runc-root="": Specify runc-compatible root

--runc-systemd-cgroup: Start runc with systemd cgroup manager

--runtime="": Runtime name or absolute path to runtime binary (default: "io.containerd.runc.v2")

--runtime-config-path="": Optional runtime config path

--sandbox="": Create the container in the given sandbox

--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.

--snapshotter-label="": Labels added to the new snapshot for this container.

--sync-fs: Synchronize the underlying filesystem containing files when unpack images, false by default

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

--uidmap="": Run inside a user namespace with the specified UID mapping ranges; specified with the

format **container-uid:host-uid:length**

--user, -u="": Username or user id, group optional (format: [:])

--with-ns="": Specify existing Linux namespaces to join at container runtime (format ':')

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+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

--temp: Mounts the snapshot mounts as a temp mount and returns a bind mount

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

delete, del, 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.

--sync-fs: Synchronize the underlying filesystem containing files when unpack images, false by default

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, del, remove, 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

metrics, metric
Get a single data point of metrics for a task with the built-in Linux runtime

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

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

--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

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

--platform="": Platform of the spec to print (Examples: 'linux/arm64', 'windows/amd64')

sandboxes, sandbox, sb, s
Manage sandboxes

run, create, c, r
Run a new sandbox

--runtime="": Runtime name (default: "io.containerd.runc.v2")

list, ls
List sandboxes

--filters="": The list of filters to apply when querying sandboxes from the store

remove, rm
Remove sandboxes

--force, -f: Ignore shutdown errors when removing sandbox

info, i
Get info about a sandbox

info
Print the server info

deprecations
list
Print warnings for deprecations

--format="": output format to use (Examples: 'default', 'json')

shim
Interact with a shim directly

--id="": shim ID

--shim-address="": shim address (default: computed from shim 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

--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 main process of the task

--api-version="": shim API version {2,3} (default: 3)

--task-id, -t="": task ID

shutdown
Shutdown shim if there is no active container

--api-version="": shim API version {2,3} (default: 3)

pprof
Provide golang pprof outputs for containerd-shim

block
Goroutine blocking profile

--debug="": Output format, value = 0: binary, value > 0: plaintext (default: 0)

goroutines
Print goroutine stack dump

--debug="": Output format, value = 0: binary, value > 0: plaintext (default: 2)

heap
Dump heap profile

--debug="": Output format, value = 0: binary, value > 0: plaintext (default: 0)

profile
CPU profile

--debug="": Output format, value = 0: binary, value > 0: plaintext (default: 0)

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

threadcreate
Goroutine thread creating profile

--debug="": Output format, value = 0: binary, value > 0: plaintext (default: 0)

trace
Collect execution trace

--debug="": Output format, value = 0: binary, value > 0: plaintext (default: 0)

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