nerdctl Docker-like CLI for containerd

nerdctl CLI → (gRPC) **containerd** images, namespaces → (shim) **runc** OCI runtime → (starts) **Container** CNI network

CONCEPTS

nerdctl is a Docker-compatible CLI that talks straight to containerd. Most docker commands and flags work unchanged; flags that are not implemented yet are rejected.

Containers and images live in a containerd namespace (default: default). Kubernetes uses k8s.io, so pass --namespace k8s.io to see its containers.

nerdctl build needs a running buildkitd. Rootless use needs a separate rootless containerd setup (see the rootless docs).

RUN

`nerdctl run <image>`
Create and start a container

`nerdctl run -d <image>`
Run in the background

`nerdctl run -it <image> sh`
Interactive shell

`nerdctl run --rm <image>`
Delete the container on exit

RUN (cont.)

`nerdctl run --name <name> <image>`
Give the container a name

`nerdctl run -p <host>:<ctr> <image>`
Publish a port

`nerdctl run -v <name>:<path> <image>`
Mount a named volume

`nerdctl run -v <dir>:<path>:ro <image>`
Bind-mount read-only

`nerdctl run -e KEY=VALUE <image>`
Set an environment variable

`nerdctl run --env-file <file> <image>`
Load variables from a file

`nerdctl run --net <network> <image>`
Join a network

`nerdctl run --restart unless-stopped <image>`
Restart after crashes and reboots

CONTAINERS

`nerdctl ps`
List running containers

`nerdctl ps -a`
List all, including stopped

CONTAINERS (cont.)

`nerdctl stop <container>`
Stop gracefully

`nerdctl start <container>`
Start a stopped container

`nerdctl restart <container>`
Stop and start again

`nerdctl rm <container>`
Remove a stopped container

`nerdctl rm -f <container>`
Force-remove a running one

INSIDE A CONTAINER

`nerdctl exec -it <container> sh`
Open a shell in a running container

`nerdctl exec <container> <cmd>`
Run one command

`nerdctl logs <container>`
Show output

`nerdctl logs -f <container>`
Follow output live

`nerdctl logs -n <lines> <container>`
Only the last lines

INSIDE A CONTAINER (cont.)

`nerdctl inspect <object>`
Low-level details as JSON

IMAGES

`nerdctl pull <image>`
Download an image

`nerdctl images`
List local images

`nerdctl build -t <name> .`
Build from a Dockerfile here

`nerdctl tag <image> <repo>:<tag>`
Add a name to an image

`nerdctl push <repo>:<tag>`
Upload to a registry

`nerdctl rmi <image>`
Remove an image

VOLUMES AND NETWORKS

`nerdctl volume create <name>`
Create a named volume

`nerdctl volume ls`
List volumes

`nerdctl network create <name>`
Create a bridge network

COMPOSE

`nerdctl compose up -d`
Create and start the stack

`nerdctl compose down`
Stop and remove containers and networks

`nerdctl compose ps`
List the stack's containers

`nerdctl compose logs -f <service>`
Follow service logs

`nerdctl compose exec <service> sh`
Shell in a service

`nerdctl compose up --build`
Rebuild images, then start

`nerdctl compose build`
Build service images

NAMESPACES AND DAEMON

`nerdctl --namespace <ns> <cmd>`
Run against a containerd namespace

`nerdctl namespace ls`
List namespaces

`nerdctl system prune`
Remove stopped containers and dangling data