crictl CRI container debugging

crictl CRI client → gRPC socket → **Runtime** containerd / CRI-O → runs → **Pod sandbox** shared network → holds → **Containers** app + sidecars

CONCEPTS

crictl talks to the node's CRI runtime directly, so it sees kubelet-managed pods that docker or nerdctl do not list. Run it on the node, usually as root.

Without an endpoint crictl probes several default sockets and is slow; set one in /etc/crictl.yaml.

CONNECT

crictl --runtime-endpoint *<socket>* **ps**
Use a specific runtime socket

crictl config --set runtime-endpoint=*<socket>*
Store the endpoint in the config

crictl config --list
Show client config values

crictl --timeout *<seconds>* **ps**
Raise the connection timeout

crictl --debug ps
Print debug output

CONNECT (cont.)

CONTAINER_RUNTIME_ENDPOINT
Env var for the runtime socket

sudo -E crictl ps
Keep env vars under sudo

crictl version
Client and runtime versions

PODS

crictl pods
List pod sandboxes

crictl pods --name *<name>*
Filter pods by name

crictl pods --label *<key>*=*<value>*
Filter pods by label

crictl pods --namespace *<ns>*
Filter pods by namespace

crictl pods --state Ready
Only ready pods

crictl pods -q
Print only pod IDs

crictl inspectp *<pod>*
Pod sandbox details

crictl stopp *<pod>*
Stop a pod sandbox

PODS (cont.)

crictl rmp *<pod>*
Remove a stopped pod

crictl runp *<pod.json>*
Create and start a pod sandbox

crictl statsp
Pod resource usage

CONTAINERS

crictl ps
List running containers

crictl ps -a
Include exited containers

crictl ps --name *<name>*
Filter containers by name

crictl ps --pod *<pod>*
Containers of one pod

crictl ps --state exited
Filter by state

crictl ps -q
Print only container IDs

crictl inspect *<container>*
Container details as JSON

crictl stop *<container>*
Stop a container

crictl rm *<container>*
Remove a container

crictl start *<container>*
Start a created container

LOGS AND EXEC

crictl logs *<container>*
Show container logs

crictl logs -f *<container>*
Follow the log

crictl logs --tail=*<n>* *<container>*
Last N lines

crictl logs --since=*<duration>* *<container>*
Logs from the last period

crictl logs --timestamps *<container>*
Prefix lines with time

crictl exec -it *<container>* **sh**
Interactive shell

crictl exec *<container>* *<cmd>*
Run one command

IMAGES

crictl images
List images

crictl images *<repo>*
Filter by repository

crictl images --digests
Show image digests

crictl images -q
Print only image IDs

IMAGES (cont.)

crictl pull *<image>*
Pull an image

crictl rmi *<image>*
Remove an image

crictl rmi --prune
Remove unused images

crictl inspecti *<image>*
Image details

NODE AND METRICS

crictl info
Runtime status and config

crictl stats
Container CPU and memory

crictl stats -a
Include stopped containers

crictl ps -o json
Machine-readable output

crictl help *<command>*
Help for one subcommand

RECIPES

Debug a crashing pod on the node

```
crictl pods --name <pod>
crictl ps -a --pod <pod-id>
crictl logs --tail=50
<container-id>
crictl inspect <container-
id>
```