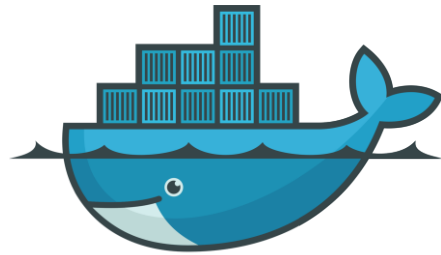


# ■ Docker vs Containerd



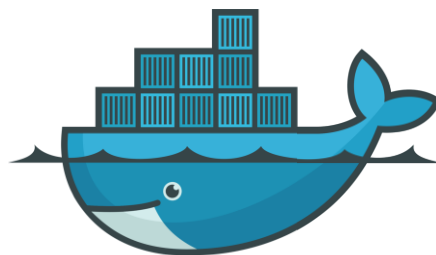
Docker

containerd

ctr

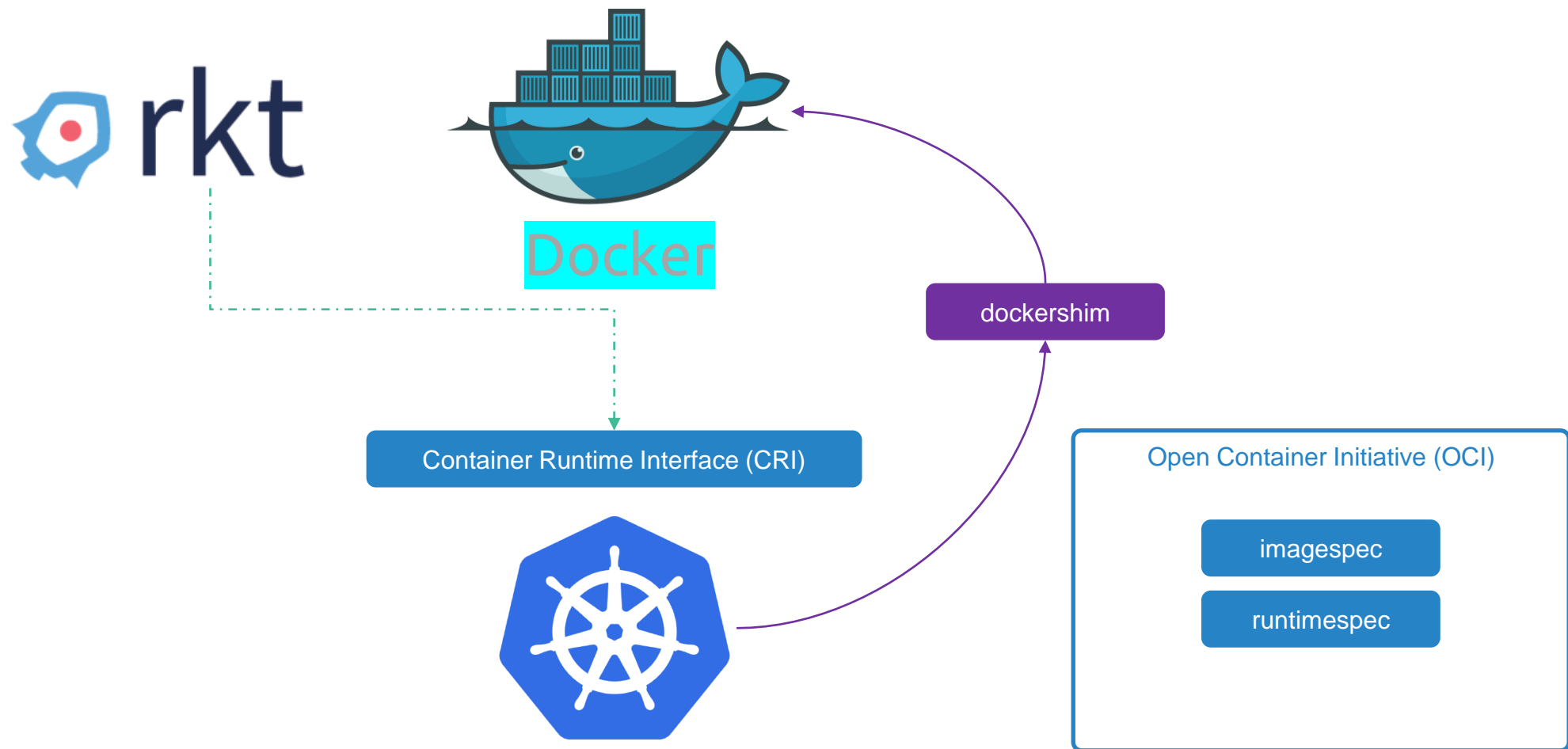
nerdctl

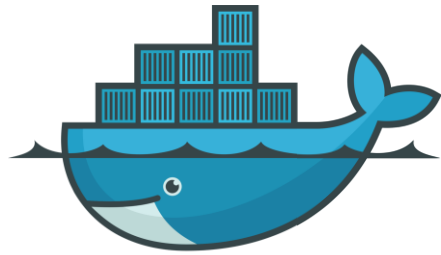
crictl



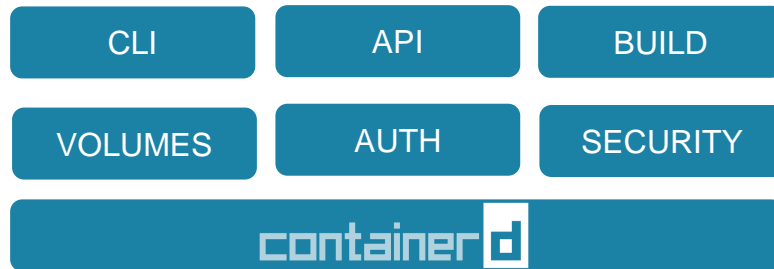
Docker







Docker



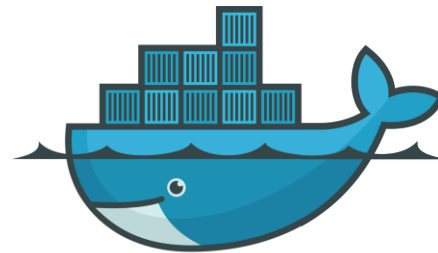
dockershim

Container Runtime Interface (CRI)





containerd



Docker

CLI

API

BUILD

VOLUMES

AUTH

SECURITY

v1.24

dockershim

Container Runtime Interface (CRI)



imagespec

runtimespec

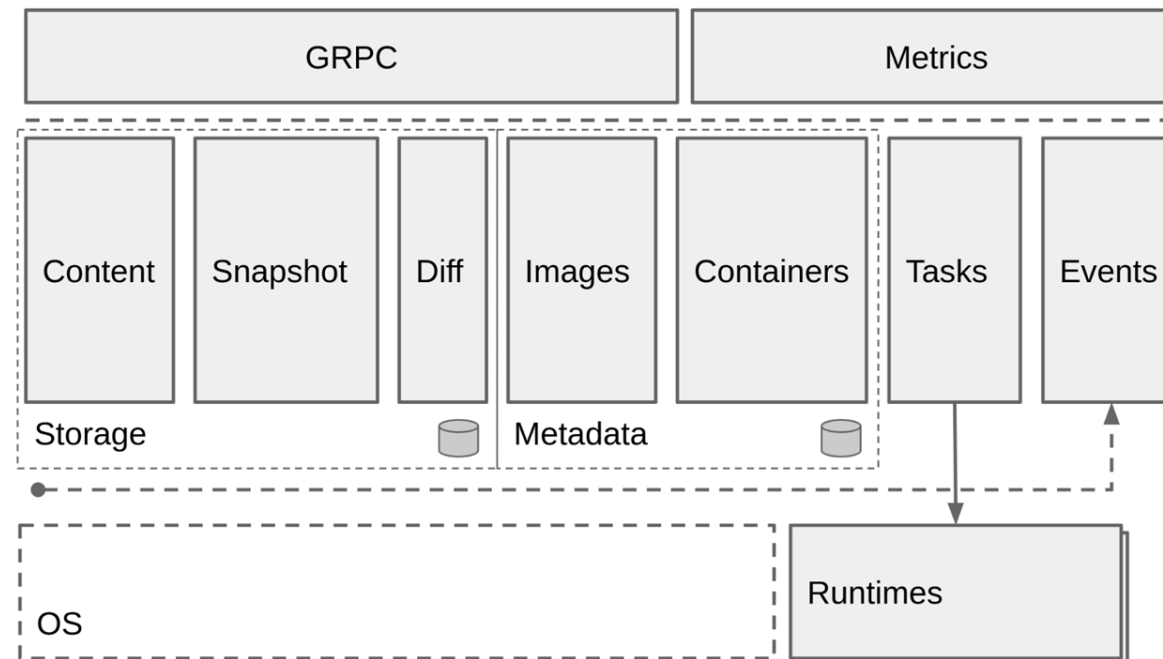
# containerd

[reference](#)
[CI passing](#)
[Nightly passing](#)
[go report A+](#)
[openssf best practices passing](#)

containerd is an industry-standard container runtime with an emphasis on simplicity, robustness, and portability. It is available as a daemon for Linux and Windows, which can manage the complete container lifecycle of its host system: image transfer and storage, container execution and supervision, low-level storage and network attachments, etc.

containerd is a member of CNCF with '[graduated](#)' status.

containerd is designed to be embedded into a larger system, rather than being used directly by developers or end-users.



# Getting started with containerd

## Installing containerd

### Option 1: From the official binaries

The official binary releases of containerd are available for the `amd64` (also known as `x86_64`) and `arm64` (also known as `aarch64`) architectures.

Typically, you will have to install [runc](#) and [CNI plugins](#) from their official sites too.

#### Step 1: Installing containerd

Download the `containerd-<VERSION>-<OS>-<ARCH>.tar.gz` archive from <https://github.com/containerd/containerd/releases>, verify its sha256sum, and extract it under `/usr/local`:

```
$ tar Cxvzf /usr/local containerd-1.6.2-linux-amd64.tar.gz
bin/
bin/containerd-shim-runc-v2
bin/containerd-shim
bin/ctr
bin/containerd-shim-runc-v1
bin/containerd
bin/containerd-stress
```

The `containerd` binary is built dynamically for glibc-based Linux distributions such as Ubuntu and Rocky Linux. This binary may not work on musl-based distributions such as Alpine Linux. Users of such distributions may have to install containerd from the source or a third party package.

**FAQ:** For Kubernetes, do I need to download `cri-containerd-(cni-)<VERSION>-<OS>-<ARCH>.tar.gz` too?

**Answer:** No.

As the Kubernetes CRI feature has been already included in `containerd-<VERSION>-<OS>-<ARCH>.tar.gz`, you do not need to download the `cri-containerd-...` archives to use CRI.

The `cri-containerd-...` archives are [deprecated](#), do not work on old Linux distributions, and will be removed in containerd 2.0.



# CLI - ctr

- ctr comes with containerD
- Not very user friendly
- Only supports limited features

While the `ctr` tool is bundled together with containerd, it should be noted the `ctr` tool is solely made for debugging containerd. The `nerdctl` tool provides stable and human-friendly user experience.

# CLI - ctr

> \_

```
$ ctr
```

```
$ ctr images pull docker.io/library/redis:alpine
```

```
$ ctr run docker.io/library/redis:alpine redis
```



<https://github.com/containerd/containerd/blob/main/docs/getting-started.md>

# CLI - nerdctl

- nerdctl provides a Docker-like CLI for containerD
- nerdctl supports docker compose
- nerdctl supports newest features in containerd
  - Encrypted container images
  - Lazy Pulling
  - P2P image distribution
  - Image signing and verifying
  - Namespaces in Kubernetes

# CLI - nerdctl

> \_

\$ docker

➡ \$ nerdctl

\$ docker run --name redis redis:alpine

➡ \$ nerdctl run --name redis redis:alpine

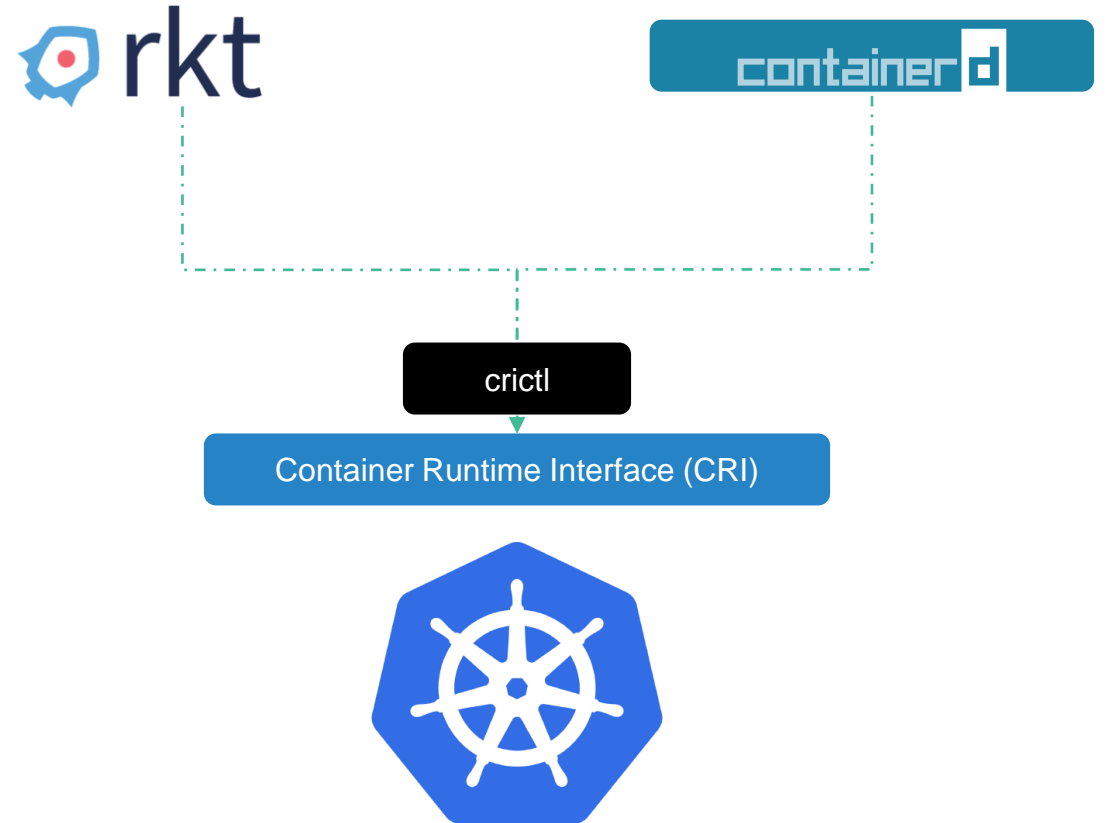
\$ docker run --name webserver -p 80:80 -d nginx

➡ \$ nerdctl run --name webserver -p 80:80 -d nginx

<https://github.com/containerd/containerd/blob/main/docs/getting-started.md>

# CLI - crictl

- crictl provides a CLI for CRI compatible container runtimes
- Installed separately
- Used to inspect and debug container runtimes
  - Not to create containers ideally.
- Works across different runtimes



# CLI - crictl

> \_

```
$ crictl
```

```
$ crictl pull busybox
```

```
$ crictl images
```

```
$ crictl ps -a
```

```
$ crictl exec -i -t 3e025dd50a72d956c4f14881fbb5b1080c9275674e95fb67f965f6478a957d60 ls
```

```
$ crictl logs 3e025dd50a72d956c4f1
```

```
$ crictl pods
```

<https://github.com/containerd/containerd/blob/main/docs/getting-started.md>

# docker vs crictl

Retrieve debugging information [↗](#)

docker cli	crictl	Description	Unsupported Features
attach	attach	Attach to a running container	<code>--detach-keys</code> , <code>--sig-proxy</code>
exec	exec	Run a command in a running container	<code>--privileged</code> , <code>--user</code> , <code>--detach-keys</code>
images	images	List images	
info	info	Display system-wide information	
inspect	inspect , inspecti	Return low-level information on a container, image or task	
logs	logs	Fetch the logs of a container	<code>--details</code>
ps	ps	List containers	
stats	stats	Display a live stream of container(s) resource usage statistics	Column: NET/BLOCK I/O, PIDs
version	version	Show the runtime (Docker, ContainerD, or others) version information	

<https://kubernetes.io/docs/reference/tools/map-crictl-dockercli/>

# docker vs crictl

## Perform Changes

docker cli	crictl	Description	Unsupported Features
create	create	Create a new container	
kill	stop (timeout = 0)	Kill one or more running container	--signal
pull	pull	Pull an image or a repository from a registry	--all-tags , --disable-content-trust
rm	rm	Remove one or more containers	
rmi	rmi	Remove one or more images	
run	run	Run a command in a new container	
start	start	Start one or more stopped containers	--detach-keys
stop	stop	Stop one or more running containers	
update	update	Update configuration of one or more containers	--restart , --blkio-weight and some other resource limit not supported by CRI.

<https://kubernetes.io/docs/reference/tools/map-crictl-dockercli/>



# CLI - crictl

> \_

```
$ crictl --runtime-endpoint
```

```
$ export CONTAINER_RUNTIME_ENDPOINT
```

- `unix:///var/run/dockershim.sock` or
- `unix:///run/containerd/containerd.sock` or
- `unix:///run/crio/crio.sock` or
- `unix:///var/run/cri-dockerd.sock`

<https://github.com/containerd/containerd/blob/main/docs/getting-started.md>



kubernetes

Container Runtime Interface (CRI)

ctr

nerdctl

crictl

Purpose	Debugging	General Purpose	Debugging
Community	ContainerD	ContainerD	Kubernetes
Works With	ContainerD	ContainerD	All CRI Compatible Runtimes