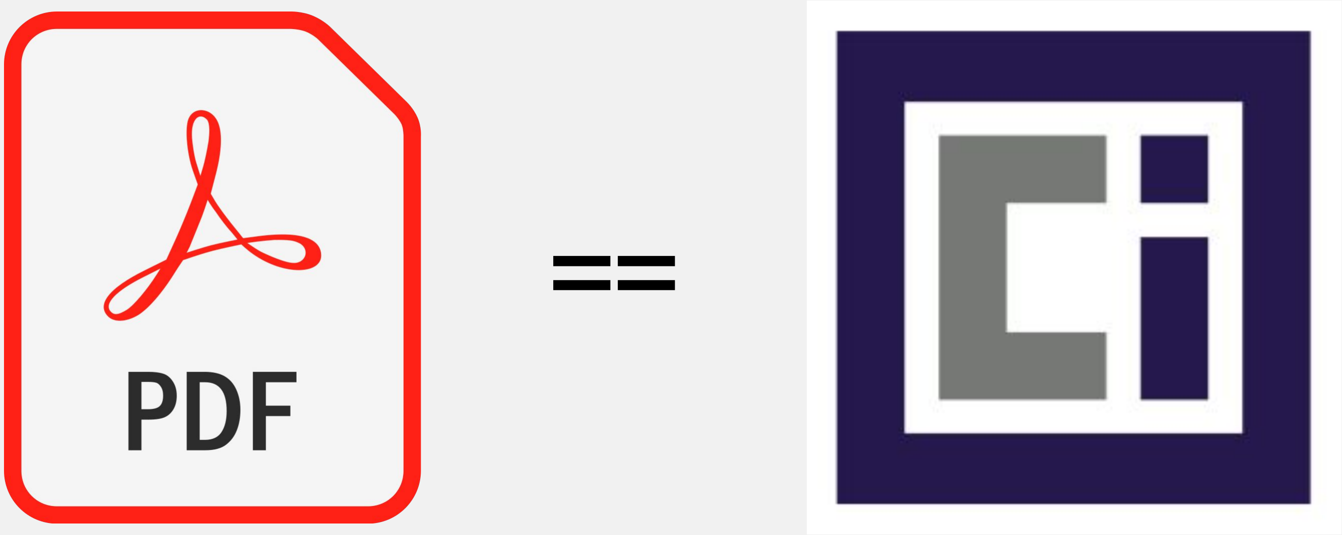
CONTAINER ECOSYSTEM

AN OPEN SOURCE SUPPLY CHAIN

One big tool, or best of breed Unix like tools based on standards

BASIC CONTAINERS ARE SIMILAR TO PDF?

Find, Run, Build, and Share. Collaboration with any reader/writer



MINIMUM TO BUILD OR RUN A CONTAINER?

Standards and open source code

- A standard definition for a container at rest
  - OCI Image Specification - includes image and metadata in a bundle
- A standard mechanism to pull the bundle from a container registry to the host
  - OCI Distribution Specification - specifies protocol for registry servers
  - github.com/containers/image
- Ability to uncompress and map the OCI image bundle to local storage
  - github.com/containers/storage
- A standard mechanism for running a container
  - OCI Runtime Specification - expects only a root file system and config.json
  - The default runc implementation of the Runtime Spec (same tool Docker uses)

WHAT ELSE DOES KUBERNETES NEED?

Standards and open source code

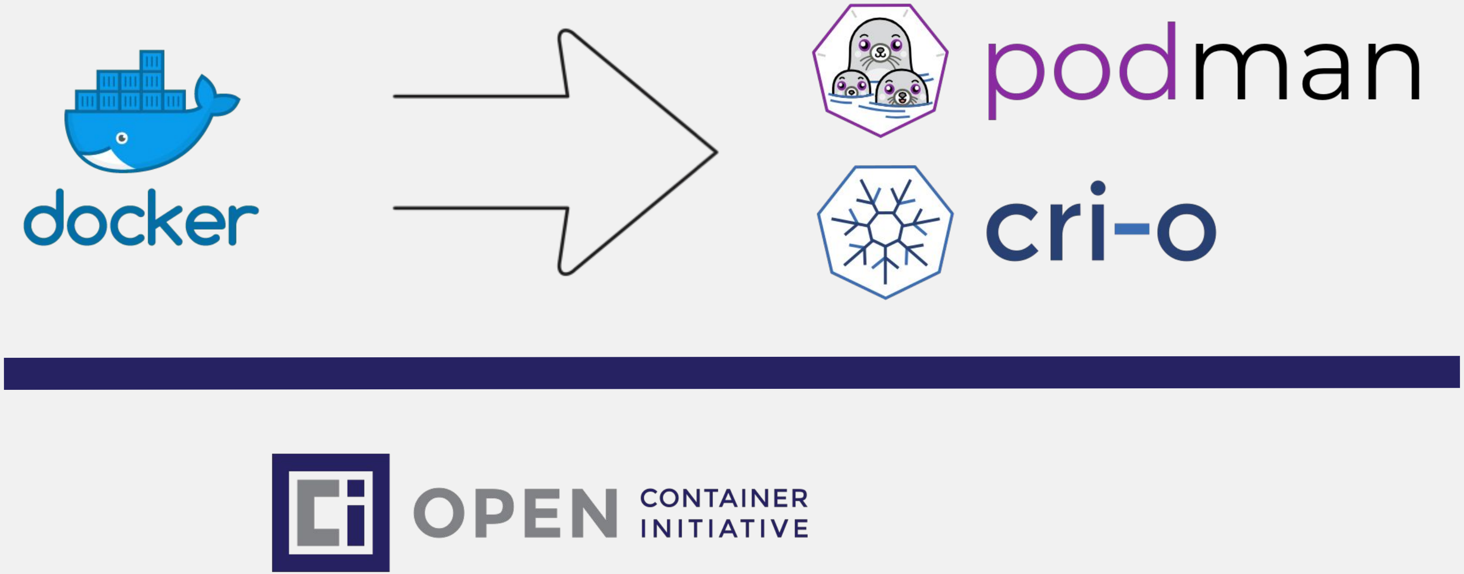
- The minimum to build or run a container

AND

- A standard way for the Kublet to communicate with the Container Engine
  - Container Runtime Interface (CRI) - the protocol between the Kubelet and Engine
- A daemon which communicates with CRI
  - gRPC Server - a daemon or shim which implements this server specification
- A standard way for humans to interface with the gRPC server to troubleshoot and debug
  - cri-ctl - a node based CLI tool that can list images, view running containers, etc

THERE ARE NOW ALTERNATIVES

Moving to Podman in RHEL 8 and CRI-O in OpenShift 4



THE UNDERLYING ECOSYSTEM

Many tools and libraries

containers

podman

buildah

cri-o

skopeo

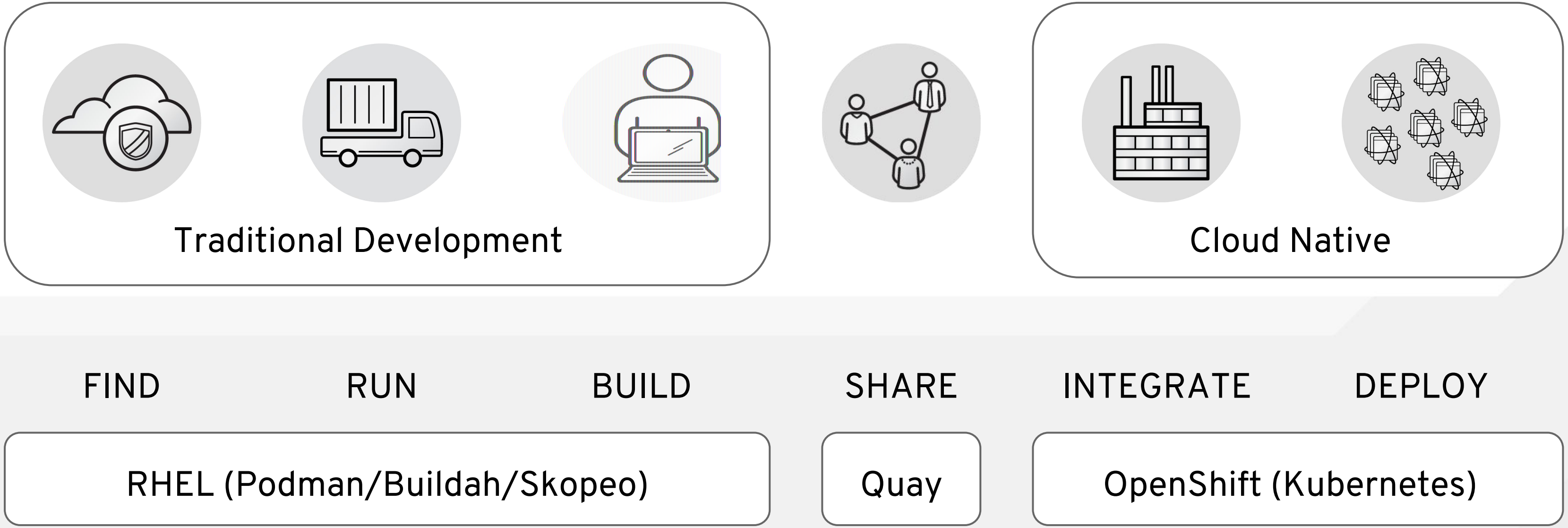
OpenSCAP

CREATING DOWNSTREAM PRODUCTS

Release timing is critical to solving problems

THE JOURNEY

Can start anywhere



Traditional Development

Cloud Native

FIND

RUN

BUILD

SHARE

INTEGRATE

DEPLOY

RHEL (Podman/Buildah/Skopeo)

Quay

OpenShift (Kubernetes)

CUSTOMER NEEDS

Mapping customer needs to solutions

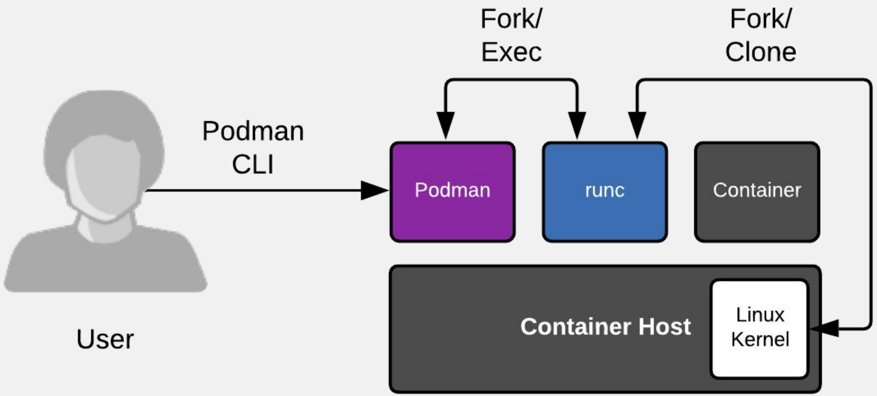
Capability	Platform	Product	Container Engine
Single Node	Linux & Container Tools	Red Hat Enterprise Linux	Podman
Multi Node	Linux & Kubernetes	OpenShift	CRI-O

Red Hat Enterprise Linux 8

The container tools module

PODMAN ARCHITECTURE

Find, Run, Build, and Share. Collaboration with any reader/writer



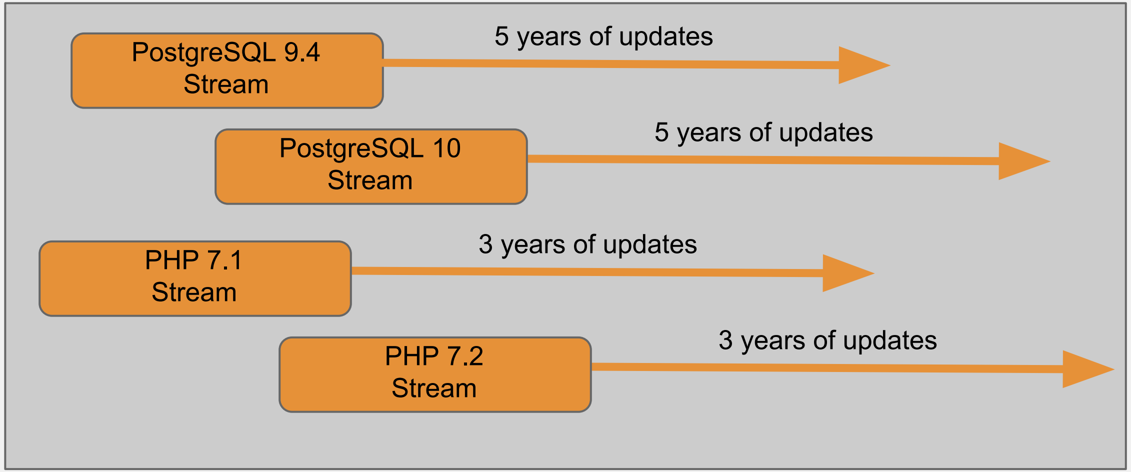
How containers run with a container engine

APPLICATION STREAMS USE MODULES

Modules are the mechanism of delivering multiple streams (versions) of software within a major release. This also works the other way round, a single stream across multiple major releases.

Modules are collections of packages representing a logical unit e.g. an application, a language stack, a database, or a set of tools. These packages are built, tested, and released together.

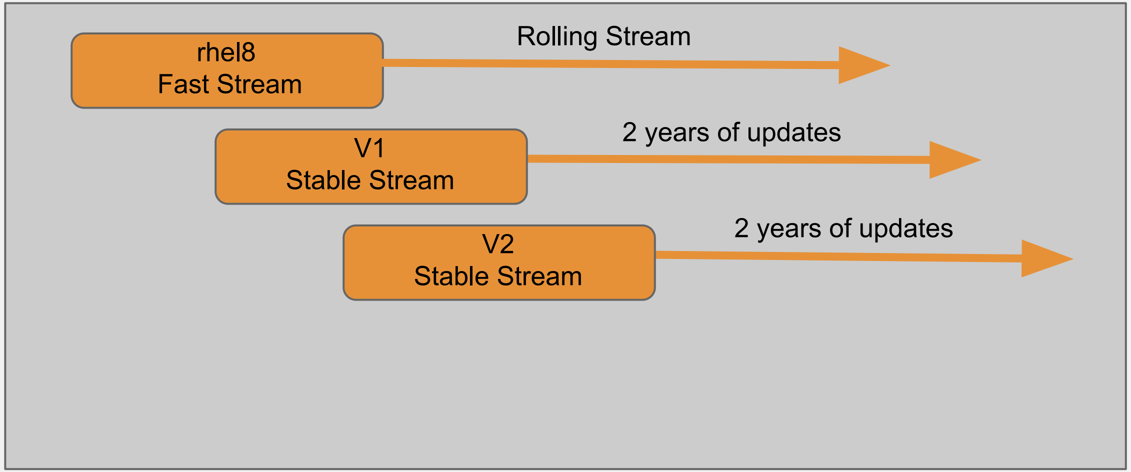
Each module defines its own lifecycle which is closer to the natural life of the app rather than the RHEL lifecycle.



THE CONTAINER TOOLS RELEASES

One **Module** delivered with multiple Application Streams based on different use cases:

- The rhel8 stream delivers new versions for developers
- The versioned, stable streams provide stability for operations
    - Created once a year, supported for two years
    - Only backports of critical fixes

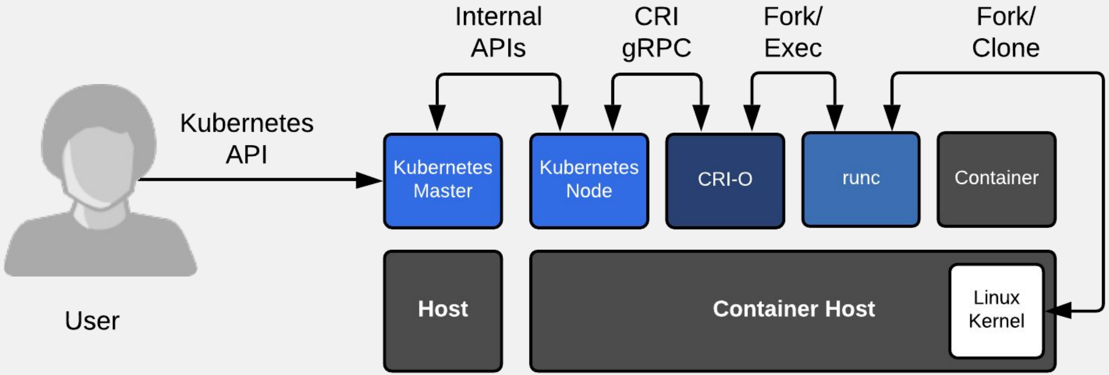


OpenShift 4

CRI-O and Buildah as a library

CRI-O ARCHITECTURE

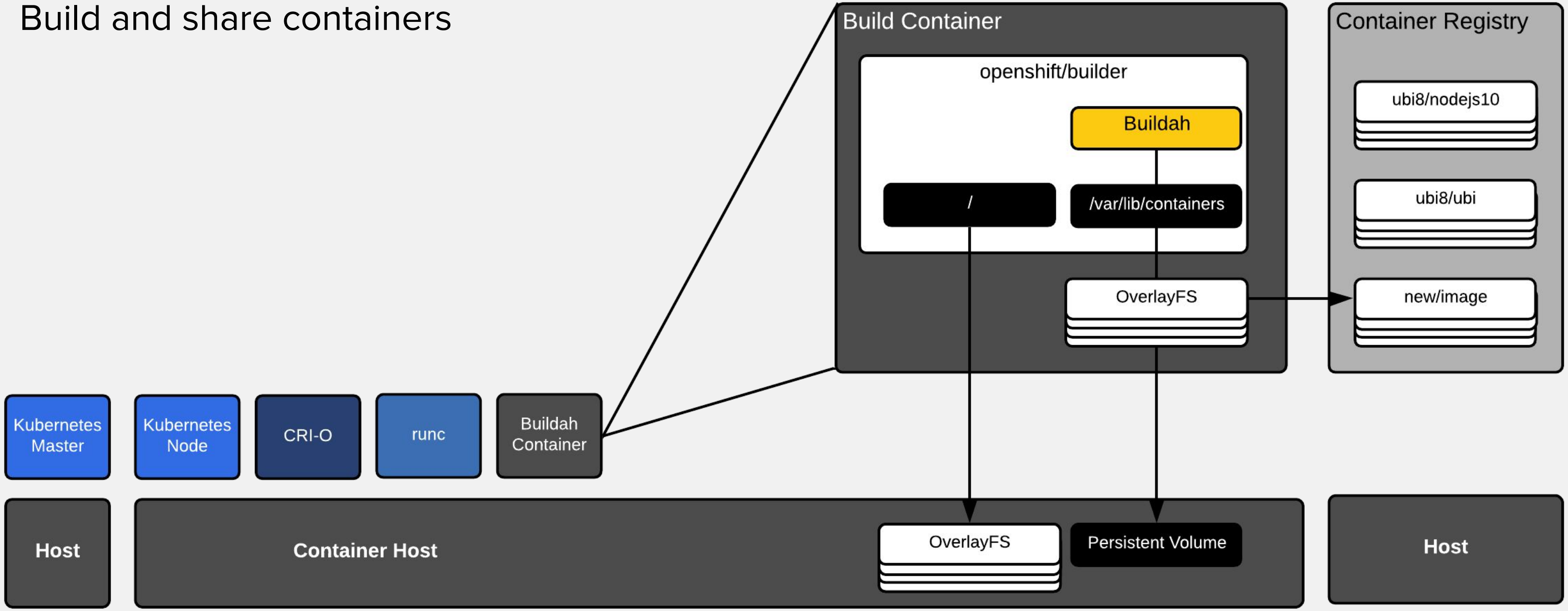
Run containers



How containers run in a Kubernetes cluster

BUILDAH ARCHITECTURE

Build and share containers

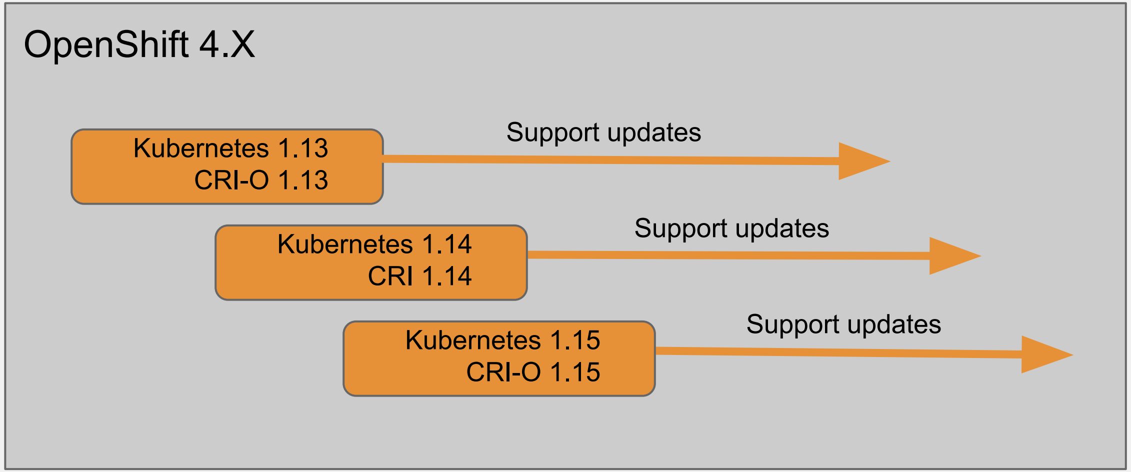


Building when all you can do is run containers

IN LOCKSTEP WITH KUBERNETES

All components for running containers released, tested, and supported together for reliability:

- CRI-O moves in lock-step with the underlying Kubernetes
- The runc container runtime is delivered side by side
- Buildah delivered as a library specifically for OpenShift. No commands for users.


OpenShift 4.X
Kubernetes 1.13
CRI-O 1.13
Support updates
Kubernetes 1.14
CRI 1.14
Support updates
Kubernetes 1.15
CRI-O 1.15
Support updates