



Yocto Project® : Building and deploying containers with meta-virtualization: now & in the future

Bruce Ashfield, Xilinx

Yocto Project Virtual Summit Europe, October 29/30, 2020

Agenda

- Goals
- (Brief) History of meta-virtualization
- Technology timeline
- Why use the Yocto project for containers ?
- Components of container build (and deployment)
- Past and current container build / deployment
- Future container build / deployment efforts

Presentation Goals

- Introduce the concepts driving container build and deploy
 - Not a how-to, not a survey
- Level set on the history
- Background / understanding of current capabilities
 - What is where, and how (why) it works ..
- Insight into direction and upcoming features



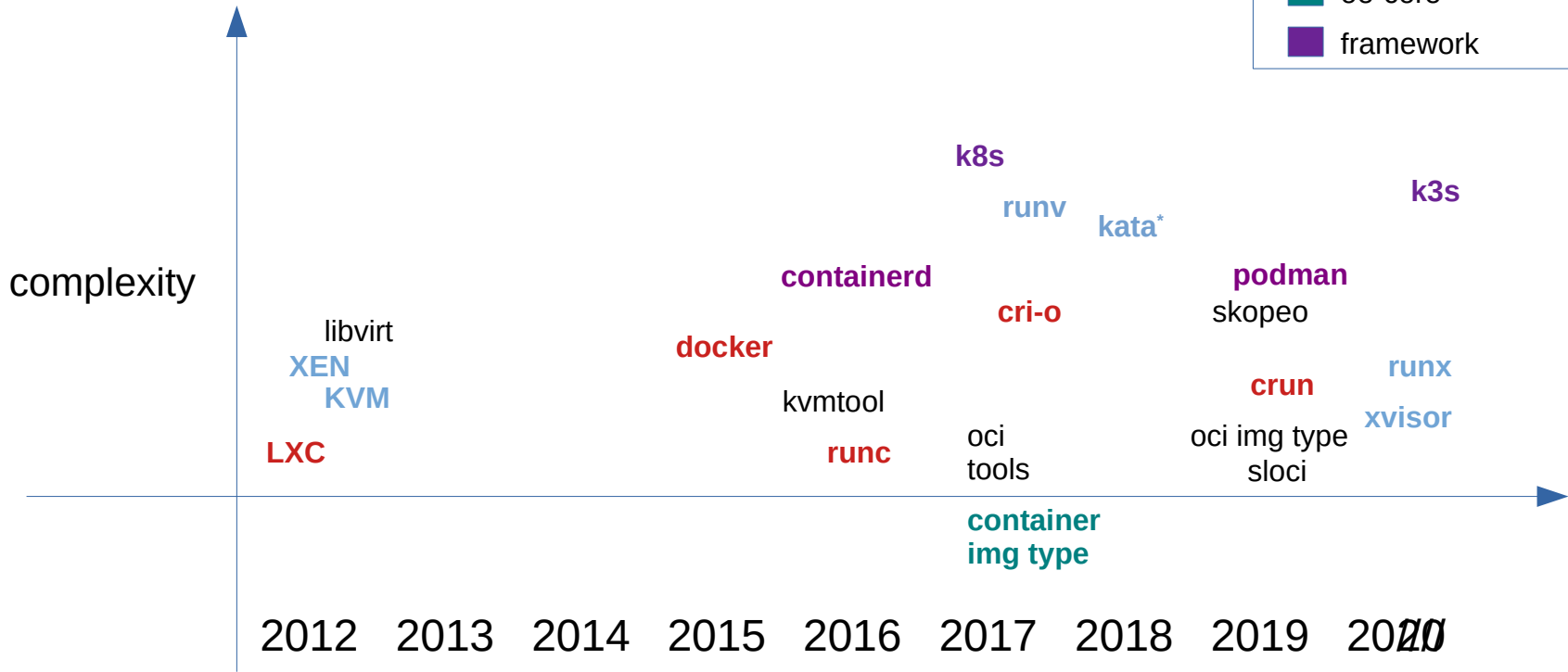
Overview

meta-virtualization: a brief history

- Started June 2012
 - 1306+ commits made by 155 contributors
- Point of integration for ‘virtualization’ technologies
 - VMs and containers
 - Core technology + support software
 - Many audiences: Bleeding edge and established tech
 - Tested (improving) and stable: needs CI
 - Baseline for creating OE derived virtualization solutions
 - Recipes migrate over time

meta-virtualization: technology timeline

- VM based
- container runtime
- tools
- oe-core
- framework



meta-virtualization: timeline summary

- Started with mainly VM solutions
 - now more container focused
- More solutions / choices than ever before
 - Very little has been removed / depreciated
- ‘Complexity’ has increased
 - Hence more complete solutions are possible
- Plumbing, tools and image assists
 - More are needed

Why the Yocto Project and containers

- Isn't it all about the application ?
 - Why care about building from source ?
- Solves problems you don't (yet) know you have
- Standards compliant and compatible
- Building block technologies
 - Choice: don't pick winners and don't lock in
 - Flexibility: Elements of the solution are spread through the ecosystem
- Configurability and tunability



Build & Deployment

What are container “build” and “deployment” ?

- For our purposes:
 - build: compilation / construction from source of a container (or fetching of OE built artifacts)
 - deployment: installing a container on a target (or image)

Container Build

- OE core has base support
- Techniques have evolved over time
 - Iterative / multiple builds and external assembly
 - Multiconfig
 - OCI Image type
- Leverage Yocto Project core values
 - Embrace, not replace or duplicate

Container Build: Challenges

- Clunky / Confusing
 - Regardless of which method you use
 - Not ‘end user’ friendly
- Path to binary container construction / reuse
 - Commonly cited guides don’t apply
- Streamlining work is in progress

Container Deployment

- Not standardized (and shouldn't be)
 - Varies based on container runtime
- Some options:
 - Direct image install
 - Registry / artifact repository push → pull
 - Management framework: k8s, k3s, etc
 - Custom 'hacks'

Container Deployment: Challenges

- Not cross friendly
 - Daemons, root requirements, host requirements, licensing, reproducibility, etc
- Requirement creep
- Large set of runtimes and frameworks



Upcoming ...

Vision

- The Yocto Project as a 1st class platform for building CNCF technologies
 - Leveraging OE advantages and technologies, feeding into external solutions
 - Baked directly into the outputs
- Simple inherit to generate containerized recipe output
- Direct deploy to images, or management framework
- Binary artifacts / re-usable base containers

In progress / Upcoming (1/2)

- Streamlined build via bbclass
 - kernel-module-split style / dynamic packaging ?
- Direct image install
 - Via -native tools
 - Autostart (depending on runtime)
- Multi-layer OCI container image build
 - Pluggable / flexible back end

In progress / Upcoming (2/2)

- Reference container host and app / system container images
- Target container (on host) rapid test
- Framework test / deployment streamlining
 - k8s, k3s, runX ..

A decorative pattern of semi-transparent grey hexagons is located in the upper-left corner of the slide.

Thanks for your time

yocto ·
PROJECT

THE
LINUX
FOUNDATION

What is the Yocto Project® ?

IT'S NOT AN EMBEDDED LINUX DISTRIBUTION,
IT CREATES A CUSTOM ONE FOR YOU.



The Yocto Project (YP) is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.

The project provides a flexible set of tools and a space where embedded developers worldwide can share technologies, software stacks, configurations, and best practices that can be used to create tailored Linux images for embedded and IOT devices, or anywhere a customized Linux OS is needed.



yocto
PROJECT

THE
LINUX
FOUNDATION