Container Operating Systems

DevOps Sydney 2021, Bin Chen

Agenda

- why we need it?
- components of a container os
- open source options
- how to build one?

Why

- container is the modern way of delivery software
- NIST SP 800-190 > Use container-specific host OSs instead of general-purpose ones to reduce attack surfaces.

Generic OS

- Ubuntu - Debian based
- Fedora/CentOs/RHEL - RPM based
- Chrome OS - Gentoo based
- Android

Package Management

```
[package format - tool  - OS]
```

- deb - apt/dpkg - Debian/Ubuntu
- spec file/rpm - yum/dnf - RHEL
- ebuild - portage - Gentoo
- oci image - docker

Container Os

- Federa coreos (Redhat/IBM)
- container linux/flatcar Linux (kinvolk)
- Google Container OS (Google)
- EKS node AMI (AWS)
- Bottlerocket OS (AWS)
- Rancher OS (Rancher)

Tasks

- Build: kernel, packages, filesystems and image format
- Run enviroment
- Components
- Customize
- Additional features
- Architecture supported (e.g ARM64)

Build

- A: build very thing from source code and put them into a bootable image (most control and most work, example: google container os)
- B: start from a working machine and install things needed and create a new image (least work and least control, example, build eks node AMI from Amazon linux 2 AMI)
- between A and B
- trend: use docker as build system, new way of implement "cross-compile"
- "bake or fry"
- tools: build system, docker, Packer, aws image builder

Run

- run on local for dev, using qemu/kvm
- run on cloud platform: openstack, aws, gce, etc.

Components Of (Target System)

- bootloader, kernel, initramfs
- system libary * libc, openssl
- system daemon/service * PID1/systemd * ssh * network * time * dns * container runtime and image tools * instance initialization (cloud-init)
- package/artifact management (optional)
- platform enviroment
- security features

Package Management - cont.

- packages are managed as container images
- optionally, a traditional package manager aforementioned

PID1

- systemd
- init

Network, DNS,and Time sync

- systemd-networkd, netplan
- systemd-resolved, coreDns
- systemd-timesyncd, chrony, ntpd

Platform Enviroment

- does the env support dhcp?
- does it has specific requirement of hard configration - encrypt, bond interface
- does it instance metadata service
- how to integrate with indentiy and access management

Container: Pull Image and Run Container

```
(no need for *build image*)
```

- dockerd
- containerd
- podman

Operations Aspects

- update and patch (kernel and components)
- inventory management

Security Features

- SELinux
- verified boot
- encrypt on rest
- Authentication

Bottlerocket

- [Components and Build](https://github.com/bottlerocket-os/bottlerocket/tree/develop/packages)

components are managed as RPM spec file and all components include kernel are built from source
- Run: Currently only support on run as EKS node (since the start up sequence assume you are in AWS environment, e.g assume the metadata service is running)
- Doc

Google Container OS

- Comoponents and Build

Google Container Linux are based on Google Chrome OS which is based on Gentoo, **so** all the components are build from source (as managed by ebuild).
Took long time for first download and build.
- Run
It can be run on on Qemu and Google Cloud.
- Doc: Google Container is a varient of Google Chrome OS, the doc to its build system is same as the chrome os build system; there are some doc to its specific features.

Flatcar Linux

- components and Build

Flatcar is a fork of CoreOS Container Linux; it is based on Geentoo so all the components are build from source.
- Run
It runs on all both local and most of the cloud platform.
- Doc

Rancher OS

- components and build

managed accross a few repos notably os-base, os-kernel and rancher/os, which is the entry to the os build. use container image as the composition build block.
kernel (4.19.x as time of writing) is old. templated build process using dapper which is wrapper on dockerfile for build.
- Run
support on qemu and most cloud enviroment
- Doc

EKS Node

- componnets and build

Build on top of Amazon Linux 2 AMI using Packer to install and create new images
- Run
Target for EKS node so runs EC2 and dependency on EKS. Since it is on top of Amazon Linux 2 it should be able to run it on Qemu but need some hacking.
- Doc
github address

Things to consider how to select

- Go with what cloud platform offers to aim at quick start
- Roll your own, if the Open Source one don't suit

Takeaways

- why container os?
- What are the components?
- what are the open source options?
- How to build one?

Thank you

Refs

- [pm](https://linuxconfig.org/comparison-of-major-linux-package-management-systems)
- [podman and buildah](https://developers.redhat.com/blog/2019/02/21/podman-and-buildah-for-docker-users/)
- [rpm](https://en.wikipedia.org/wiki/RPM_Package_Manager)
- [portage](https://en.wikipedia.org/wiki/Portage_(software))

Thank you