Flatcar in the world of Cluster API

Cloud Native Prague Meetup | 24. Sep. 2020

Hi, I'm Dongsu

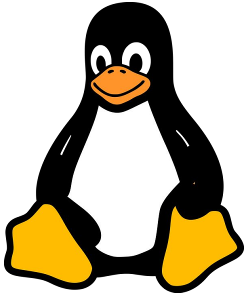
Dongsu Park

Software Engineer, Kinvolk

Github: **dongsupark**
Email: **dongsu@kinvolk.io**

Who is Kinvolk?

Independent,
community-driven company
since 2015



Technical background: Linux,
Security & Containers

Microsoft
CoreOS
Nasdaq
PubNative
ShiftLeft
weaveworks
Lufthansa Systems
CHEF
SAP

Open Source Engineering
and Support Services

Kinvolk and Open Source

LOKOMOTIVE

Modern Kubernetes distro inspired by CoreOS Tectonic

FLATCAR

Minimal Linux distro derived from CoreOS Container Linux

rkt

gobpf

Service Mesh Interface

OpenTelemetry

systemd

weave scope

PROJECT CALICO

Original developers of, and contributors to, numerous other open source projects

100% Open Source Business Model

What is a “Container Linux”?

Just the minimal distribution required for containers

Reduced dependencies

Less base software to manage

Reduced attack surface area

Repeatable deployment without requiring chef/puppet

Immutable file system

Operational simplicity for management at scale

Removes entire category of security threats - e.g. runc vulnerability CVE-2019-5736 kinvolk.io/blog/2019/02/runc-breakout-vulnerability-mitigated-on-flatcar-linux

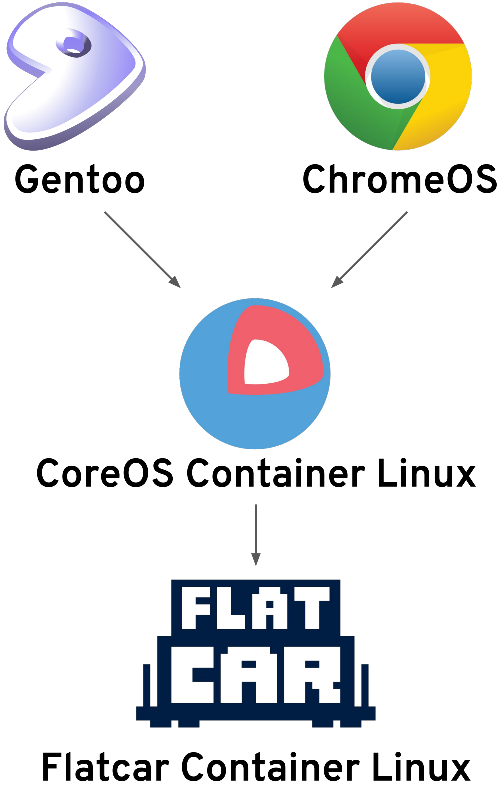
Automated, streamlined updates

Operational simplicity for management at scale

Easily apply all latest security patches

Rollback partition

Flatcar Heritage



Gentoo

ChromeOS

CoreOS Container Linux

Flatcar Container Linux

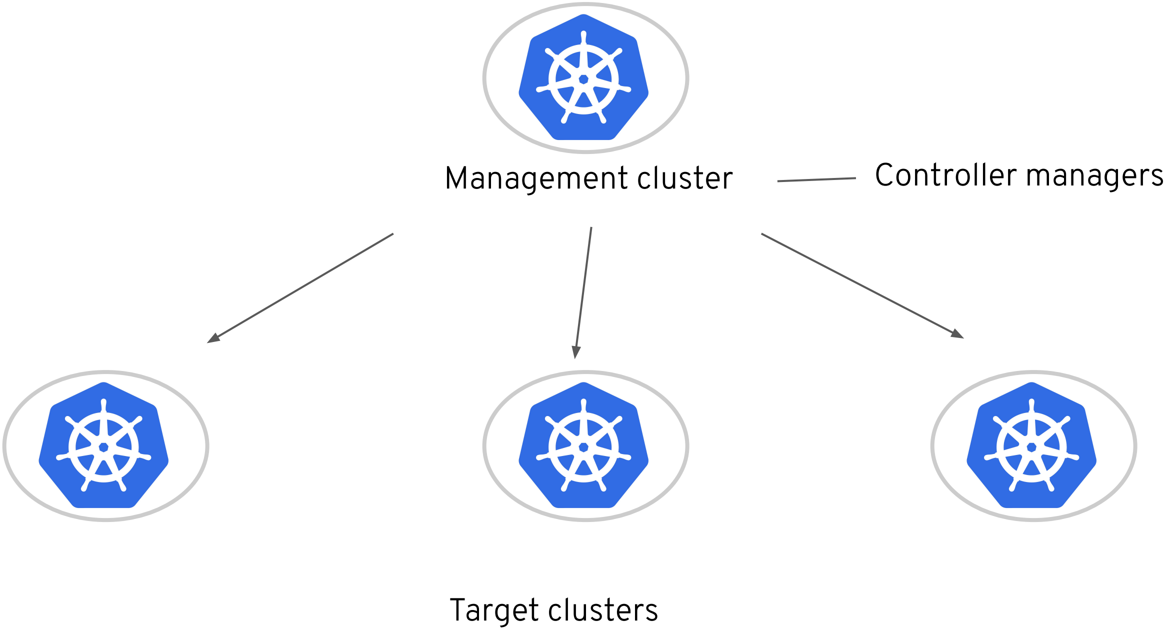
Cluster API - Introduction

- Sub-project of Kubernetes
    - Addresses challenge when bootstrapping Kubernetes clusters
    - Migration across multiple cloud providers or regions
    - Provision of declarative APIs for cluster creation and management
    - SIG-cluster-lifecycle
    - Initial release: Apr. 2019

Cluster API - providers

- Bootstrap provider
  - Kubeadm
  - Talos
- Supports multiple infrastructure providers
  - AWS
  - Azure
  - DigitalOcean
  - Google Cloud
  - Packet
  - VMware
  - etc.

Cluster API



Cluster API - Image Builder

- Tool to generate base images for Cluster API
  - Based on Packer and Ansible
  - Multiple distros
    - CentOS, Photon, Ubuntu, etc.
  - Multiple cloud providers
    - AWS, Azure, DigitalOcean, Google Cloud, VMware OVA, Qemu
  - Includes tools needed for bootstrapping Kubernetes
    - Kubeadm, kubectl, kubelet

Goal for Flatcar

- Make image-builder generate Flatcar images
  - For multiple cloud providers

- Integrate Flatcar into the entire Cluster API
  - Based on images generated by image-builder
  - For multiple infrastructure providers

Image Builder for Flatcar - Challenges

- No package manager in Flatcar
  - Container-optimized OS
  - Number of packages are not available by default
  - Manual installation needed on the image builder side

- Flatcar’s /usr partition is read-only
  - Not possible to simply copy binaries into /usr/local/bin
    - Workaround: /opt/bin
  - Conflict with existing binaries located under read-only partitions
    - Docker, containerd, cri-tools

Image Builder for Flatcar - Challenges

- Limitations in Ansible
    - Ansible cannot detect Flatcar as distro
        - Fixed in Ansible 2.10 (released 22.Sep)
    - Ansible simply requires packages as either rpm or deb
        - Sub-optimal for container-optimized OS

- PRs in progress
    - https://github.com/kubernetes-sigs/image-builder/pull/248
    - https://github.com/kubernetes-sigs/image-builder/pull/371
    - https://github.com/kinvolk/image-builder/pull/7

Demo

Cluster API - challenges

- Bootstrap provider
  - Only supports cloud-init by default
  - No support ignition needed by Flatcar
  - On-going work to support ignition for bootstrap provider
    - https://github.com/kubernetes-sigs/cluster-api/issues/3430
    - https://github.com/kubernetes-sigs/cluster-api/pull/3437

Cluster API - challenges

- Vary across individual infrastructure providers
  - Different requirements for each provider
  - AWS: userdata-related parts heavily rely on cloud-init
    - Multipart mime messages go through the AWS secrets manager
    - Need to reimplement the userdata parts
    - https://github.com/kubernetes-sigs/cluster-api-provider-aws/issues/1875
  - vSphere: network configurations rely on cloud-init

Cluster API - challenges

- Fork ignition for Flatcar (?)
  - Pros: Can resolve on-going issues around ignition
  - Cons: result in diverging from upstream ignition
  - Exploring alternative options

Conclusion

- Flatcar in conventional provisioning world
  - Bumpy ride ahead
  - Progress in adjusting image-builder, Packer, Ansible
  - Make provisioners work without assumptions like package manager

- Cluster API for Flatcar
  - Work in progress, a key focus for the Flatcar team
  - How to efficiently support ignition
  - How to deal with different infrastructure providers

Thank you!

Dongsu Park

Github: **dongsupark**
Email: **dongsu@kinvolk.io**

Kinvolk

Blog: **kinvolk.io/blog**
Github: **kinvolk**
Twitter: **kinvolkio**
Email: **hello@kinvolk.io**