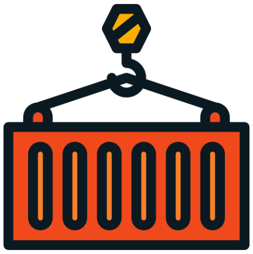
Datasheet: Flatcar Container Linux

Why Container Linux?

Modern applications are increasingly deployed as containers. Hence, enterprises need a reliable, secure foundation for those applications that can be managed at scale.

Traditional enterprise Linux distributions are not well suited to this task. This is where Flatcar Container Linux comes in:



Container Optimized

Flatcar Container Linux is designed for containerized applications.

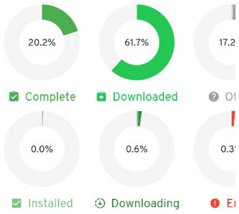
❏ It comes with a minimal set of components required to deploy containers, including a modern Linux kernel and the Docker container engine.

❏ Packages such as language runtimes are already bundled in container images, so are not included at the host level.

❏ Many typical Linux tools are included in a utilities container.

Automated Updates

Flatcar Container Linux discovers when updates are available, and can apply them automatically, ensuring the entire fleet is always up to date. For system administrators wanting more control, the Kinvolk Update Service enables update policy controls.



Secure by Design

Flatcar Container Linux can play a key role in your security strategy.

❏ As a minimal Linux distribution, the potential attack surface area is minimized.

❏ Flatcar has an immutable file system. This eliminates a whole class of vulnerabilities.

❏ Automatic updates ensure security patches are always up to date.

Seamless Migration from CoreOS

In the best traditions of the open source community, Flatcar Container Linux was created as a fork of CoreOS Container Linux, itself derived from Gentoo and ChromeOS.

This ensures a seamless migration path — an in-place upgrade from CoreOS to Flatcar is no more intrusive than a regular CoreOS version update. Even your existing configuration files will continue to work.

Kubernetes Optional

Unlike some container-optimized Linux distributions, which either only work with Kubernetes, or are only available with commercial support as part of a bundle with a Kubernetes distribution, Flatcar Container Linux is a standalone offering.

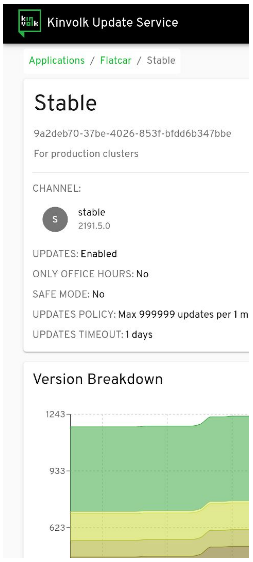
Of course, Flatcar Container Linux works great with Kubernetes (including Kinvolk's own Lokomotive distribution). But it also supports other orchestrators such as DC/OS, or non-orchestrated container deployments.

Kinvolk Update Service

Available as a managed service or on-premise software deployment, the Kinvolk Update Service extends the automatic update capabilities of Flatcar Container Linux with enterprise policy controls and visibility/monitoring.

Through a clean user interface, administrators can

- ❏ define upgrade groups (for example, enabling most servers to be running the latest stable release, with a small subset on beta)
- ❏ restrict upgrade hours
- ❏ limit the rate of updates
- ❏ view version upgrade status across the entire fleet
- ❏ view an audit trail of upgrade-related activity for each host



Kinvolk Update Service

Applications / Flatcar / Stable

Stable

9a2deb70-37be-4026-853f-bfdd6b347bbe

For production clusters

CHANNEL:

S stable 2191.5.0

UPDATES: Enabled

ONLY OFFICE HOURS: No

SAFE MODE: No

UPDATES POLICY: Max 999999 updates per 1 m

UPDATES TIMEOUT: 1 days

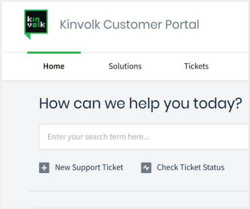
Version Breakdown

100% Open Source

Flatcar Container Linux, as well as the technology behind Kinvolk Update Service, are and always will be 100% open source software, available under Open Source Initiative approved licensed.

No “enterprise edition”. No restrictive license. No vendor lock-in.

Enterprise Grade Support



How can we help you today?

For peace of mind, you want to know that you have the support of the team behind your core infrastructure code. With a Kinvolk Flatcar Container Linux Subscription, you will be assigned a customer success engineer to ensure a smooth migration to Flatcar and responsive handling of any issues raised during production.

The Kinvolk Customer Portal, available to all subscribers, enables easy ticket management and access to knowledge base articles.

An enterprise-grade service level agreement, at either the Standard (business hours) or Premium (24x7) level, backs up Kinvolk's commitment to rapid incident response direct from our renowned engineering team.

SLA Level:	Standard	Premium	
Hours of Coverage	Business hours (9am-5pm)	24x7x365 for P1 & P2 Business hours for P3 & P4	
Number of Cases	Unlimited	Unlimited	
Priority Level	Initial and Ongoing Response Time	Initial Response Time	Ongoing Response Time
P1 - Urgent	1 business hour	1 hour	1 hour
P2 - High	4 business hours	2 hours	4 hours
P3 - Medium	1 business day	4 business hours	1 business day
P4 - Low	2 business days	2 business days	2 business days

Life on the Edge

Flatcar Container Linux is built and delivered in four "channels". The traditional alpha, beta, stable channels indicate the level of maturity of new features, with a gradual progression from alpha (earliest) through to stable (mature and most trusted for production deployment).

With the addition of the Edge channel, users can try out experimental features that may or may not make it into future releases, and hackers have a way to contribute new ideas upstream without having to worry about the stringent standards for new features applied to the main Flatcar development train.

What People Say About Kinvolk and Flatcar

“We believe Flatcar Container Linux will be welcomed by developers and administrators who rely on CoreOS today for its lightweight approach, built-in support of the Docker container engine, and as a proven platform for Docker Enterprise.” – Justin Graham, VP Product Management at Docker

“As we realized CoreOS Container Linux was reaching end of support, we reached out to the team at Kinvolk and were impressed by their commitment to the original CoreOS vision, and their ability to support us through a seamless transition to Flatcar. With a Flatcar Container Linux subscription in place, we now feel comfortable that we have a viable long-term platform strategy.” – Michael Ferraro, VP Platform at UpGuard

“We are inviting our CoreOS users to migrate to Flatcar Container Linux, which has been fully integrated into the Packet platform.” – Zac Smith, CEO at Packet

“Outside of my former CoreOS coworkers, I trust the Kinvolk team most with the CoreOS legacy.” – Joe Thompson, former CoreOS engineer

“Kinvolk's expertise in modern Linux technologies was the reason we wanted to work with them to help us build rkt, the security-first app container runtime.” – Jonathan Boulle, Technical Product Lead, CoreOS & Technical Oversight Committee Member, CNCF

“CoreOS Container Linux has been one of our customers' most popular options, so we are pleased to see Kinvolk taking the initiative to enable them to continue with the same software foundation on a commercially supported basis.” – Jamie Dobson, CEO at Container Solutions

“CoreOS Container Linux has seen wide adoption across the industry, and many users are looking for a path forward that doesn't entail significant operational disruption. Over the past 18 months, Kinvolk has demonstrated their commitment and ability to do this with Flatcar Container Linux, and I'm pleased to see they are now backing that up with commercial support and managed update service.” – Joe Sandoval, SRE Manager, Infrastructure Platform at Adobe, and OpenStack User Committee member

“We are excited to be collaborating with Kinvolk to enable an upgrade path for our customers who are currently deploying Kubernetes with CoreOS, as well as supporting future Kubernetes adopters who need a secure operating system foundation.” – Matt Barker, CEO at Jetstack

“At Giant Swarm, we strongly believe in the minimal and immutable container Linux approach, which is why we built our managed Kubernetes service on CoreOS (in part with the help of Kinvolk's engineering team). Going forward, Flatcar Container Linux offers us an identical operational experience, with the commitment to long-term maintenance, support and security updates that we need to ensure a stable platform for our demanding enterprise customers such as Adidas and Vodafone.” – Timo Derstappen, chief technology officer, Giant Swarm